Handbook of Communications Security

F. GARZIA





Handbook of Communications Security

About the Author



Fabio Garzia is Professor of various subjects regarding security in the Safety & Security and Civil Protection Masters Program at the University of Rome "La Sapienza" and in other Masters programs at the same university and at other universities in Italy. He is also an Adjunct Professor at Wessex Institute of Technology (UK), and a member of the European Academy of Science and Arts (Salzburg, Austria).

He is the author of more than 100 scientific papers published in various international journals and conference proceedings and author or editor of several books regarding security, both in Italian and English. He is co-editor of the International Journal of Safety & Security Engineering (WIT Press). He serves as a reviewer for various international scientific journals, as a member of various committees and working groups regarding security and ICT, a member of the Scientific Committees of various international conferences, a member of the Executive Committee of IEEE International Carnahan Conference on Security Technology, and as co-Chairman of the Safety & Security Engineering conference series.

A consultant, designer, construction manager and tester of security and ICT systems, he has worked or is still working for: Vatican City State, Senate of Italian Republic, Gran Sasso mountain INFN underground laboratories, Italian Space Agency, high velocity railway, high security sites, airports, ports, rail stations, museum, basilicas, different public and private subjects, etc. He is an Expert Member of the Board of Public Works of Italy.

Handbook of Communications Security

F. Garzia

University of Rome "La Sapienza" Italy

WIT PRESS Southampton, Boston



F. Garzia

University of Rome "La Sapienza" Italy

Published by

WIT Press
Ashurst Lodge, Ashurst, Southampton, SO40 7AA, UK
Tel: 44 (0) 238 029 3223; Fax: 44 (0) 238 029 2853
E-Mail: witpress@witpress.com
http://www.witpress.com

For USA, Canada and Mexico

WIT Press
25 Bridge Street, Billerica, MA 01821, USA
Tel: 978 667 5841; Fax: 978 667 7582
E-Mail: infousa@witpress.com
http://www.witpress.com

British Library Cataloguing-in-Publication Data

A Catalogue record for this book is available
from the British Library

ISBN: 978-1-84564-768-1
eISBN: 978-1-84564-769-8

Library of Congress Catalog Card Number: 2012954752

No responsibility is assumed by the Publisher, the Editors and Authors for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein. The Publisher does not necessarily endorse the ideas held, or views expressed by the Editors or Authors of the material contained in its publications.

©WIT Press 2013. All rights reserved.

Printed by Lightning Source, UK.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the Publisher.

To Nadia, Marco and Gabriele

This page intentionally left blank

CONTENTS

Preface ..... **xix**

Introduction **1**

Chapter 1 Fundamentals of Telecommunications **3**

1.1 Introduction 3
1.1.1 Mode of network operation 3
1.1.2 Network hardware 3
1.1.3 Network software 8
1.1.4 Reference models 11
1.1.5 Examples of network 15
1.1.6 International entities of the telecommunications world 22
1.2 The physical layer 24
1.2.1 Signals theory 24
1.2.2 Transmission over guided media 46
1.2.3 Wireless transmission 48
1.2.4 Satellite transmission 50
1.2.5 Fixed telephone network 51
1.2.6 The cellular telephone network 52
1.3 Data link physical layer 55
1.4 Medium Access Control sub-layer 57
1.4.1 Wireless networks 62
1.4.2 Switching in the data link layer 74
1.5 The network layer 79
1.5.1 Routing algorithms 81
1.5.2 Congestion control algorithms 84
1.5.3 Quality of service 86
1.5.4 Connection between networks 87
1.5.5 The layer network on the Internet 88
1.6 The transport layer 96

1.6.1 The UDP transport protocol on the Internet 99
1.6.2 The TCP transport protocol on the Internet 99
1.6.3 Performance on networks 104
1.7 The session layer 108
1.8 The presentation layer 109
1.9 The application layer 109
1.9.1 The domain name system 110
1.9.2 Email 111
1.9.3 The World Wide Web 113
1.9.4 Multimedia 124

Chapter 2 Cryptography 137

2.1 Introduction 137
2.2 General elements of cryptography 141
2.2.1 Replacement ciphers and transposition ciphers 141
2.2.2 XOR operation 142
2.2.3 One-time pad 143
2.2.4 Computer algorithms 144
2.2.5 Introduction to protocols 144
2.2.6 Communication by symmetric cryptography 147
2.2.7 One-way functions 147
2.2.8 One-way hash functions 148
2.2.9 Communication by public-key cryptography 148
2.2.10 Hybrid cryptosystems 149
2.2.11 Digital signature 149
2.2.12 Digital signatures with encryption 153
2.2.13 Generation of random or pseudo-random sequences 153
2.2.14 Exchange of keys 154
2.2.15 Authentication 157
2.2.16 Authentication and key exchange 158
2.2.17 Multiple public-key cryptography 158
2.2.18 Division of a secret 159
2.2.19 Secret sharing 159
2.2.20 Cryptographic protection of archives 160
2.2.21 Stamping services 160
2.2.22 Delegated signature 161
2.2.23 Group signature 161
2.2.24 Key escrow 162
2.2.25 Digitally certified email 162
2.2.26 Length of the symmetric key 162
2.2.27 Public-key length 164
2.2.28 Comparison between the length of the symmetric key and the length of the public key 165
2.2.29 Birthday attacks in relation to one-way functions 165
2.2.30 Optimal key length 165

2.2.31 Key management 166
2.2.32 Key generation 166
2.2.33 Key transfer 168
2.2.34 Key verification 168
2.2.35 Using keys 168
2.2.36 Key update 169
2.2.37 Key storage 169
2.2.38 Compromising of keys 169
2.2.39 Lifespan of keys 170
2.2.40 Destruction of keys 170
2.2.41 Key management in public-key systems 171
2.2.42 Algorithm types and modes 171
2.2.43 Use of algorithms 175
2.3 Elements of basic maths for cryptography 178
2.3.1 Information theory 178
2.3.2 Complexity theory 180
2.3.3 Numbers theory 181
2.3.4 Factorisation 185
2.3.5 The generation of prime numbers 186
2.3.6 Discrete logarithms in finite fields 186
2.4 Data Encryption Standard 187
2.4.1 The DES algorithm 187
2.4.2 Security of DES 191
2.4.3 Differential and linear analysis 193
2.4.4 DES variants 195
2.5 Other block ciphers 196
2.6 Cipher combination 196
2.6.1 Double encryption 197
2.6.2 Triple encryption 197
2.6.3 Whitening 197
2.6.4 Cascading 197
2.7 Pseudo-random sequence generators and flow ciphers 197
2.7.1 Congruent linear generators 197
2.7.2 Linear shift records with feedback 198
2.7.3 Design and analysis of stream ciphers 199
2.7.4 Stream ciphers based on LFSR 199
2.7.5 A5 stream cipher 199
2.7.6 Additive generators 200
2.7.7 PKZIP 200
2.7.8 Design of stream ciphers 200
2.7.9 Generation of multiple streams from a single pseudo-random generator 200
2.8 Real random sequence generators 201
2.8.1 Random noise 201

2.8.2 Computer clock 202
2.8.3 Keyboard latency typing 202
2.8.4 Polarisation and correlation 202
2.8.5 Distillation of randomness 203
2.9 One-way hash functions 203
2.9.1 Use of the symmetric block algorithms for generation of one-way hash functions 204
2.9.2 Use of public-key algorithms for the generation of one-way hash functions 204
2.9.3 Message authentication code 205
2.10 Advanced Encryption Standard 205
2.10.1 Introduction to AES 205
2.10.2 Preliminary concepts 206
2.10.3 Description of the algorithm 210
2.10.4 Rational schema 211
2.10.5 Encryption 212
2.10.6 Key expansion function 212
2.10.7 Decryption 213
2.10.8 Security 213
2.11 Public-key algorithms 214
2.11.1 The RSA algorithm 215
2.11.2 Elliptic curve cryptosystems 217
2.11.3 Other public-key cryptosystems 217
2.12 Public-key algorithms for digital signature 217
2.12.1 Digital signature algorithm 217
2.12.2 Digital signature via discrete logarithms 219
2.12.3 Other algorithms for digital signature 219
2.13 Algorithms for the exchange of keys 220
2.13.1 Diffie–Hellman 220
2.13.2 Station–station protocol 221
2.13.3 Exchange of encrypted keys 221
2.14 Quantum cryptography 222
2.15 Practical applications 223
2.15.1 Management protocol of secret IBM keys 223
2.15.2 STU-III 224
2.15.3 Kerberos 224
2.15.4 Kryptonight 225
2.15.5 SESAME 225
2.15.6 IBM common cryptographic architecture 225
2.15.7 ISO Authentication 226
2.15.8 Privacy Enhanced Mail 228
2.15.9 TIS/PEM 228
2.15.10 Message Security Protocol 228
2.15.11 Pretty Good Privacy 229
2.15.12 Smart card 229

2.15.13 Public-key cryptographic standards 230
2.15.14 CLIPPER 230
2.15.15 CAPSTONE 230
2.15.16 Other systems 231

Chapter 3 Steganography 233

3.1 Introduction 233
3.2 History of steganography 233
3.2.1 The Egyptians 233
3.2.2 The Greeks 233
3.2.3 The Chinese 234
3.2.4 Gaspar Schott 234
3.2.5 Johannes Trithemius 234
3.2.6 Giovanni Porta 234
3.2.7 GirolamoCardano 234
3.2.8 Blaise de Vigenere 235
3.2.9 Auguste Kerckhoffs 235
3.2.10 Bishop John Wilkins 235
3.2.11 Mary Queen of Scots 235
3.2.12 George Washington 235
3.2.13 Air mail by pigeons in Paris in 1870 236
3.2.14 The First World War 236
3.2.15 The Second World War 236
3.2.16 The Vietnam War 237
3.2.17 Margaret Thatcher 237
3.3 Principles of steganography 237
3.3.1 The background to secret communication 237
3.3.2 Steganographic security systems 241
3.3.3 The concealment of information in data noise 242
3.3.4 Adaptive and non-adaptive algorithms 243
3.3.5 Active and malicious hackers 243
3.3.6 Concealment of information within written text 245
3.3.7 Examples of invisible communication 246
3.4 The principal steganographic techniques 246
3.4.1 Preliminary definitions 247
3.4.2 Substitution methods 247
3.4.3 Methods for domain transformation 251
3.4.4 Spread spectrum methods 254
3.4.5 Statistical methods 256
3.4.6 Distortion methods 256
3.5 Steganalysis 257
3.6 Practical examples 259
3.6.1 Cryptapix 260
3.6.2 Data stash 261

3.6.3 Hermeticstego 262
3.6.4 Hide in picture – Blowfish 263
3.6.5 Hide in picture – Rijndael 264
3.6.6 OpenPuff 265
3.6.7 S tools – Data Encryption Standard (DES) 266
3.6.8 S tools – International Data Encryption Algorithm (IDEA) 267
3.6.9 S tools – MDC 268
3.6.10 S tools – Triple DES 269
3.6.11 SilentEye 270

Chapter 4 Digital Watermarking 271

4.1 Introduction 271
4.2 History and terminology 271
4.3 Basic principles 272
4.4 Applications 273
4.5 Algorithm requirements 274
4.6 Evaluation of systems 275
4.7 Watermark removal algorithms 278
4.8 Future evolution and standardization 278
4.9 Watermarking technologies 279
4.9.1 Selection of pixels or blocks 279
4.9.2 Work selection space 280
4.9.3 Formatting of the watermarking signal 283
4.9.4 Fusion of the message in the document to be watermarked 284
4.9.5 Optimisation of the watermark detector 284
4.9.6 Watermarking of video images 285
4.10 Strength requirements 285
4.10.1 Signal decrease 286
4.10.2 Malfunction of the watermarking detector 287
4.10.3 Watermark counterfeiting 288
4.10.4 Watermark detection 290
4.10.5 System architectures 290
4.11 Digital fingerprint 291

Chapter 5 Security in Wired Networks 293

5.1 Introduction 293
5.2 Introduction to security policies and risk analysis 294
5.3 Firewall 297
5.3.1 Design of a firewall 299
5.3.2 Limits of firewalls 300
5.3.3 Risk regions 300
5.3.4 Introduction to firewalls 301
5.3.5 Types of firewalls 302
5.3.6 Firewall architectures 306

5.3.7 Further types of firewalls 307
5.3.8 Firewall selection 317
5.3.9 Further firewall considerations 320
5.3.10 Location of firewalls 323
5.3.11 Network security assessments 324
5.4 The S-HTTP protocol 327
5.4.1 Introduction to S-HTTP 328
5.4.2 Digital signatures in S-HTTP 331
5.5 Secure Socket Layer 333
5.5.1 Features of browsers and SSL servers 336
5.5.2 Tunnels in firewalls and SSL 337
5.5.3 S/MIME: secure extensions 338
5.6 Intrusion detection 339
5.6.1 Installation of an IDS on a host 342
5.6.2 IDS fusion 343
5.6.3 Configuration of an IDS 344
5.7 Network attacks 346
5.7.1 Denial-of-service attack 346
5.7.2 Number sequence anticipation attack 346
5.7.3 TCP protocol hijack 348
5.7.4 Sniffer attack 348
5.7.5 Active desynchronisation attack 349
5.7.6 Spoofing attack 353
5.7.7 Hyperlink spoofing 355
5.7.8 Web spoofing 355
5.8 Authentication 358
5.9 Virtual Private Networks 360
5.9.1 The choice of a VPN 363
5.9.2 Various VPN solutions 364
5.9.3 Setting up a VPN 365
5.10 The exchange of Kerberos keys on distributed systems 365
5.10.1 Ticket flags 372
5.10.2 Kerberos archive 374
5.10.3 Vulnerability of Kerberos 375
5.11 Security of commercial transactions on the Internet 376
5.11.1 Use of credit cards on the Internet 380
5.11.2 The Secure Electronic Transmission protocol 381
5.12 Audit trails 382
5.13 Java language and related security aspects 384
5.14 Web browser security 387
5.14.1 Simple attacks on Web browsers 389
5.14.2 ActiveX components and associated security issues 389
5.14.3 Web cookies 391

5.15 Scripts and security issues 392
5.15.1 CGI scripts 392
5.15.2 The languages used for creating scripts 395
5.15.3 Perl language 396
5.15.4 CGI scripts and security issues 397
5.16 Computer viruses and security policies 399
5.16.1 Replication 400
5.16.2 Concealment 402
5.16.3 Bomb 404
5.16.4 Worm virus 405
5.16.5 Trojan horses 406
5.16.6 Virus prevention 406
5.16.7 Virus protection 409
5.17 Analysis of attacks 411
5.17.1 Execution of the attack 416
5.18 Prevention of attacks 420
5.19 Disaster prevention and recovery 421
5.19.1 Division of disasters 421
5.19.2 Network disasters 421
5.19.3 Server disasters 427
5.19.4 Disaster simulation 432
5.20 Network security policy 432

Chapter 6 Security of Wireless Networks 445

6.1 Introduction 445
6.2 Introduction to wireless networks 445
6.2.1 The propagation of electromagnetic waves 446
6.2.2 The signal-to-noise ratio 448
6.2.3 The main players that operate on wireless 449
6.3 Risks and threats in the wireless industry 449
6.3.1 Objectives of the information theory 449
6.3.2 Analysis 450
6.3.3 Spoofing 450
6.3.4 Denial-of-service 451
6.3.5 Malicious codes 451
6.3.6 Social engineering 451
6.3.7 Rogue access points 452
6.3.8 Security of cellular telephony 452
6.3.9 Hacking and hackers in the wireless industry 453
6.3.10 Radio frequency identification 456
6.4 Wireless technologies in the physical layer 456
6.4.1 The industrial, scientific and medical band 457
6.4.2 Modulation techniques used 457
6.5 Frame management in the wireless industry 458

6.5.1 Beacon 459
6.5.2 Probe request 459
6.5.3 Probe response 459
6.5.4 Authentication 459
6.5.5 Association request 460
6.5.6 Association response 460
6.5.7 Disassociation and de-authentication 460
6.5.8 Carrier sense multiple access/collision avoidance 460
6.5.9 Fragmentation 462
6.5.10 Distributed coordination function 462
6.5.11 Point coordination function 463
6.5.12 Interframe spacing 463
6.5.13 Service set identifier 463
6.6 Local wireless networks and personal wireless networks 464
6.6.1 Ad hoc mode 464
6.6.2 Infrastructure mode 464
6.6.3 Bridging 465
6.6.4 Repeater 465
6.6.5 Mesh networks 466
6.6.6 Wireless LAN standards 466
6.6.7 Personal area networks 467
6.7 Wireless WAN technology 475
6.7.1 Cellular phone technology 475
6.7.2 GPS technology 491
6.7.3 TETRA technology 492
6.7.4 Wireless Application Protocol 495
6.8 Wireless antennae 500
6.8.1 Introduction to antennae for wireless devices 500
6.8.2 Fresnel zone 502
6.8.3 Types of antennae 503
6.9 The implementation of wireless networks 504
6.9.1 Requirement acquisition 504
6.9.2 Cost estimate 505
6.9.3 Evaluation of investment 505
6.9.4 Site analysis 506
6.9.5 Network design 509
6.9.6 Device verification 509
6.9.7 Development and installation 509
6.9.8 Certification 510
6.9.9 Audit 510
6.10 Wireless devices 510
6.10.1 Access points 510
6.10.2 Mobile user devices 511

6.11 The security of wireless LANS 513
6.11.1 History of wireless security 514
6.11.2 Authentication 514
6.11.3 SSID 516
6.11.4 Foundations of wireless security 516
6.11.5 WEP 516
6.11.6 802.1x 518
6.11.7 RADIUS 520
6.11.8 EAP 521
6.11.9 WPA 528
6.11.10 802.11i 529
6.11.11 WPA2 534
6.11.12 WAPI 534
6.11.13 Detection of false access points 535
6.12 Violation of wireless security 535
6.12.1 The process of attack 536
6.12.2 Breach technologies 538
6.12.3 Access point breach techniques 543
6.13 Wireless security policies 545
6.13.1 Introduction to security policies 545
6.13.2 Drafting of security policies 546
6.13.3 Risk assessment 547
6.13.4 Impact analysis 548
6.13.5 The areas of wireless security policies 548
6.14 Wireless security architectures 551
6.14.1 Static WEP 551
6.14.2 VPN 553
6.14.3 Wireless gateway 556
6.14.4 802.1x 558
6.14.5 Comparison between the different wireless architectures 559
6.15 Wireless tools 561
6.15.1 Scanning tools 562
6.15.2 Sniffing tools 562
6.15.3 Hybrid tools 562
6.15.4 DoS tools 563
6.15.5 Cracking tools 563
6.15.6 Access points attack tools 563
6.15.7 Security tools 563

Chapter 7 Voice Security 565

7.1 Introduction 565
7.2 Characteristics of the spoken language 565
7.2.1 The structure of language 567
7.2.2 Phonemes and phones 567

7.3 Voice configuration 567
7.3.1 The classic source–filter model 567
7.3.2 The general source–filter model 568
7.3.3 Linear prediction modeling 569
7.4 The transmission of voice signals 570
7.5 Voice signal encryption 572
7.5.1 Voice signal analogue encryption 573
7.5.2 Digital encryption of voice signals 580
7.6 Voice source encoding 581
7.6.1 The formant vocoder 581
7.6.2 The channel vocoder 581
7.6.3 The vocoder based on linear prediction 582
7.6.4 The sinusoidal model 585
7.6.5 Standards 585
7.7 Voice cryptanalysis 585
7.7.1 Tools and parameters for voice cryptanalysis 586
7.7.2 Using the spectrograph for cryptanalysis 586
7.7.3 Analogue methods 587
7.7.4 Cryptanalysis of digital ciphers 588
7.7.5 Linear prediction vocoder cryptanalysis 588
7.8 VoIP systems security 588

Chapter 8 Protection from Bugging 593

8.1 Introduction 593
8.2 Devices for environmental bugging 594
8.2.1 Bugging devices and miniature cameras 594
8.2.2 Directional microphones 599
8.2.3 Environmental bugging using laser devices 600
8.2.4 Trackers using GPS technology 600
8.2.5 Mobile phone bugging devices 601
8.2.6 Other devices 602
8.2.7 Stethoscopic microphones 602
8.2.8 Miniature audio and video recorders 602
8.2.9 Keystroke recorders on a computer keyboard (key catcher) 602
8.2.10 Bugging software for computers 603
8.2.11 Portable document scanners 603
8.3 Devices and techniques for protection against environmental bugging 604
8.3.1 Scanners 604
8.3.2 Broadband bugging device detectors 605
8.3.3 Bugging device detectors based on cellular technology 606
8.3.4 Spectrum analysers 607
8.3.5 Multifunction spectrum analysers 609
8.3.6 Multifunction devices 609
8.3.7 Non-linear junction detectors 610

8.3.8 Hidden miniature camera detectors 614
8.3.9 Wireless remote camera detectors 614
8.3.10 Electromagnetic jammers 615
8.3.11 Jammers for audio devices 615
8.3.12 Jammers for laser beam bugging devices 616
8.3.13 Encrypted phones 616
8.3.14 Software utilities 617
8.3.15 TEMPEST 617
8.4 Procedures and guidelines for suspected environmental bugging 619

Bibliography 623
Index 635

PREFACE

Communications security is a strategic sector for the protection of privacy as well as personal, corporate, national and international security.

The everyday life of every person and every type of organisation involves communication. Landlines, fax, mobile phones, SMS, MMS, the Internet and email are some of the many means by which communication normally takes place.

The interception, damage to (or loss of) information during its communication, can produce material, non-material and economic damage from a personal, business and societal perspective.

The aim of this book is to provide the concepts relating to all aspects of communications security, even for the reader who is not particularly well versed, commencing with the basic concepts and progressing to those that are more advanced and topical, attempting to best simplify the subject matter, using explanatory figures, graphs, diagrams and tables as much as possible and using mathematical formulas only where absolutely necessary.

In this sense, the book aims to provide a basic comprehensive overview in order to allow the reader to explore specific issues by consulting specialised texts relating to the various topics covered in this book. To facilitate the possible work of exploration of the subjects addressed, there is a bibliography at the end of each chapter for the reader.

This book is aimed at designers and administrators of telecommunications, computer and integrated security systems, system engineers, system analysts, security managers, critical infrastructure managers, the police force, the armed forces, researchers and sector technicians; it is also useful for security personnel, private investigators, university students and for all those who in some way need to communicate securely for personal or business reasons.

My warmest thanks go to Professor Roberto Cusani for his valuable advice and invaluable task of critical review of this book and Engineer Lucia Fontana for the important assistance provided in the preparation of figures used in this book.

This page intentionally left blank

INTRODUCTION

If the eighteenth century was dominated by great mechanical systems that paved the way for the industrial revolution and the nineteenth century with the invention of steam engine that laid the foundation of modern industry, the twentieth century was dominated by the collection, processing and transmission of information via communication systems. A global cable telephone network was in fact created, radio and television were invented and developed, great computer networks including the Internet were also developed, communication satellites were launched and used and so forth.

This rapid process of development is converging at very high speeds towards integrated solutions, nullifying the differences between the collection, storage and processing of information.

If, at the beginning, the mere fact of communication itself took precedence, then over time the issue of communicating in a secure and reliable manner, ensuring the availability, integrity and confidentiality of information being transmitted, emerged. These concepts are discussed in detail in this book.

Currently, most daily activities are based on the communication at any level, of any type of information and by any means: voice, telephone, fax, mobile phone, short message service (SMS), multimedia messaging service (MMS), e-mail, Internet, etc.

To ensure that communications reach the recipient with the required level of security and confidentiality, adequate knowledge and skills must be used that will ensure the availability, integrity and confidentiality of the same communications.

The aim of this book is to explain all the aspects of communications security in order to inform the reader of the risks involved when communicating with any of the means currently available and to be able to implement countermeasures aimed at reducing these same risks.

Following this introduction, the fundamentals of telecommunications will be addressed in a structured manner, which are essential for those without sufficient knowledge of the sector; cryptography; steganography; digital watermarking; wired network security, wireless network security, voice communication security; and security against eavesdropping.

The metric prefixes given in Table I.1 are used in this book:

It is important to remember that in information technology, conventions slightly different from the norm are followed because the magnitudes are always multiples of 2 and, as such, the prefix "kilo" (k) is taken to mean 2^{10}, that is 1,024 and not 1,000. Similarly, the prefix "mega" (M) is taken to mean 2^{20}, that is 1,048,576; the prefix "giga" (G) is taken to mean 2^{30}, that is 1,073,741,824; and so on.

Table 1 1. Metric prefixes used.

Prefix	Exponent
Exa	10^{18}
Peta	10^{15}
Tera	10^{12}
Giga	10^{9}
Mega	10^{6}
Kilo	10^{3}
Milli	10^{-3}
Micro	10^{-6}
Nano	10^{-9}
Pico	10^{-12}
Femto	10^{-15}
Atto	10^{-18}

However, in telecommunications, base 10 notation is followed and thus a 10 mega bit per second (10 Mbps) LAN network transmits 10,000,000 bits per second. So, it is important to highlight this difference between the computer and telecommunications sectors to avoid confusion.

CHAPTER 1

FUNDAMENTALS OF TELECOMMUNICATIONS

1.1 Introduction

Telecommunications currently take place, for the most part, by means of integrated networks that can be fixed, mobile or mixed. In order to understand the concepts of communication security, it is therefore necessary to briefly explain the concepts of network and telecommunication system.

The general organisation of networks and the associated reference models illustrated below, after which the layers of interest of the same for security purposes will be reviewed.

1.1.1 Mode of network operation

Before exploring illustration of the network hardware, the network operation modes must be explained that are client – server and peer-to-peer. Client – server mode of operation is assumed to be in the presence of a computer system where data are stored on a high-performance computer, called server, and in the presence of a series of reduced performance computers, called clients, that need to draw these data from the server. Client and server machines are connected via a network, as shown in Figure 1.1, without providing additional details on the network itself that can be fixed, mobile or mixed.

This type of configuration is widely used by exploiting telecommunications networks. Specifically, this involves a process on the client machine and a process on the server machine. In particular, the client sends a message through the network to the server. The server, having received the request, performs the requested task and sends the client the corresponding response. This mode of operation is shown in Figure 1.2.

In the peer-to-peer communication mode, each computer can communicate with one or several computers simultaneously, departing from the rigid client – server schema, as shown in Figure 1.3, but always using a fixed, mobile or mixed network.

1.1.2 Network hardware

Unfortunately, there is no division shared by all that can be applied to all the telecommunication networks except for two parameters that are represented by transmission technology and scale. These transmission technologies can be divided into broadcast networks and point-to-point networks.

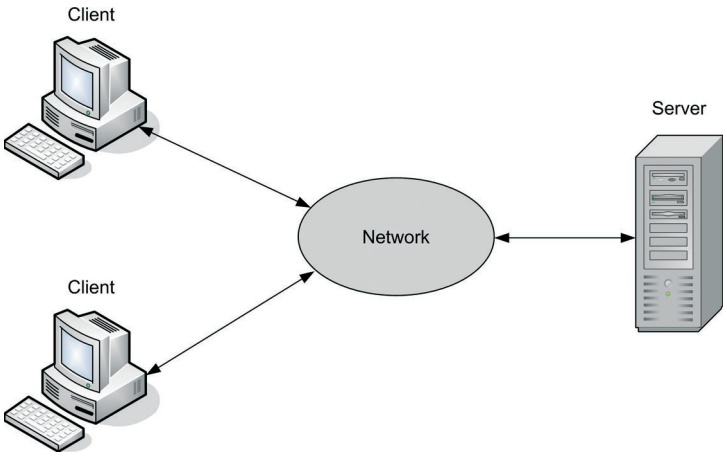


Figure 1.1 Example of client–server connection.

Broadcast networks are characterised by the presence of a single communication channel that is shared by all users. Messages, in some cases called packets, are sent by each user and received by all the others. The recipient is identified by means of a suitable packet address field. When a user receives a packet, he controls the relative address field: if the packet is addressed to the same user, it is processed, otherwise it is ignored. Broadcast networks also allow addressing of packets to all users by using a special code in the destination field: in this case, the message is processed by all users of the network. In some cases, multicast mode is provided that allows transmission to a limited group of users.

Point-to-point networks consist of different connections of individual users. In this case, the message, in order to reach the destination from source, must pass through a certain number of users. This mode of transmission is also called unicast. As there may be multiple paths, the best ones in terms of length and hence the baud rate from time to time must be found.

Usually, the networks characterised by extended dimensions use unicast mode, while the networks that are geographically limited use broadcast mode.

With regard to the scale of the networks, these can be classified starting from the smallest size. Thus:

1. Personal area networks (PANs), which meant for use by one person, for example, that constituted by a computer and related peripherals (keyboard, mouse, printer, etc.).
2. Local area networks (LANs), which is described in section 1.1.2.1.
3. Metropolitan area networks (MANs), which is described section 1.1.2.2.
4. Wide area networks (WANs), which is described in section 1.1.2.3.

To this classification must be added the network constituted by the connection of two or more networks called Internetwork, of which the most significant example is represented by the Internet. The classification in question is summarised in Table 1.1.

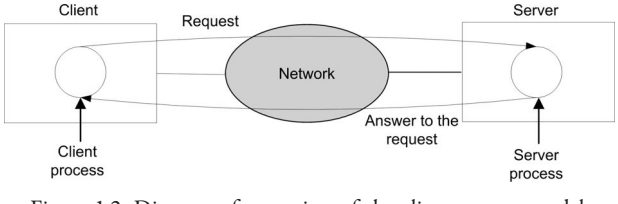


Figure 1.2 Diagram of operation of the client–server model.

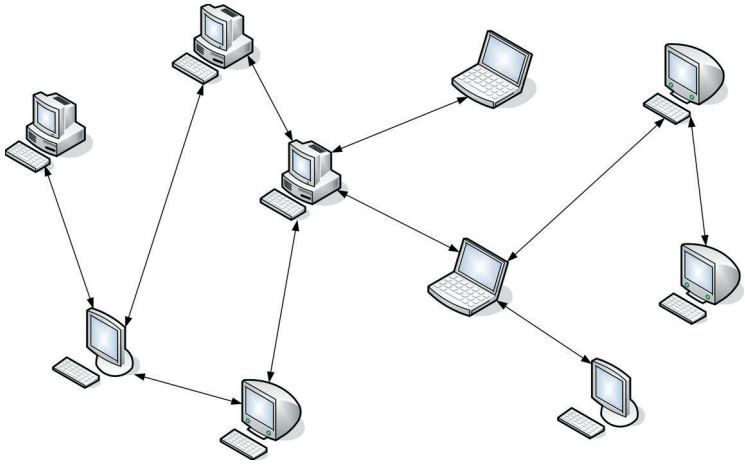


Figure 1.3 Example of operation of the peer-to-peer model.

1.1.2.1 Local network

LANs are represented by networks installed in a room, a building or a campus, and can reach extensions of several kilometres. They are used to connect together various processing units such as computers, workstations and various peripherals in order to exchange information.

LANs are defined by three main characteristic parameters:

1. size;
2. transmission technology;
3. topology.

LANs are obviously characterised by relatively small dimensions for that reason the maximum transmission time is well known at the design stage and allowing their maximum exploitation. It can use cable-based transmission technology with speeds ranging from 10 Mbps (10 million bits per second) to 10 Gbps (10 billion bits per second). LANs are usually of the broadcast type and can use different technologies, represented mainly by bus and ring technology, as shown schematically in Figure 1.4.

On a bus network, only one machine can transmit at a time, which at that particular moment is a master: at the same time, none of the other machines must transmit. To resolve any conflicts of simultaneous transmission, an arbitration mechanism, which can be centralised or distributed, must be

Table 1.1 Network classification on the basis of size.

Distance between computing units	Computing units of the same	Kind of network
1 m	Square metre	Personal area network
10 m	Room	Local area network
100 m	Building	Local area network
1 km	Campus	Local area network
10 km	City	Metropolitan area network
100 km	State	Wide area network
1,000 km	Continent	Wide area network
10,000 km	Planet	Internet

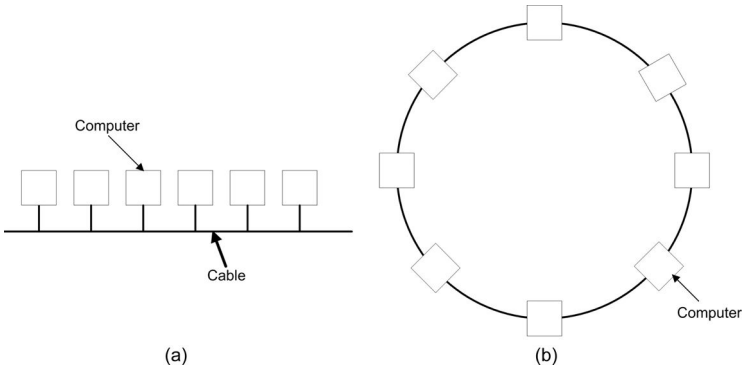


Figure 1.4 Two examples of LAN broadcast networks: (a) bus and (b) ring.

used. For example, the standard known as Ethernet (which will be illustrated in more detail below) is a type of bus broadcast network with decentralised control and able to operate with speeds ranging from 10 Mbps to 10 Gbps. With this type of network, if two or more computers are simultaneously transmitting (collision), the same await a random time to reattempt transmission. Instead, with ring topology, every bit propagates in a completely autonomous fashion, without having to wait for the remaining part of the packet of which it is a part, frequently travelling the entire ring, in most cases, before the whole packet is transmitted. It is clear that, like all broadcast systems, an arbitration system must be used. Broadcast networks can also be divided into static and dynamic, depending on the mode of channel allocation.

1.1.2.2 Metropolitan area network

MANs are able to cover a whole city. They are mainly used to distribute television services via cable as well as Internet services.

1.1.2.3 Wide area network

WANs are able to cover a whole country or an entire continent, connecting between different computers, also called hosts. The various hosts are connected to an appropriate subnet. Hosts belong to various customers while the subnet is the property of telephone companies or Internet Service Providers (ISPs).

In general, the subnet is composed of two elements: transmission lines and switching elements. Transmission lines are used to transmit information between various hosts and can be made up of copper, optical fibre or wireless connections. Switching elements are devices, typically called routers, which, by connecting several transmission lines, when data to be transmitted arrives, decide on which line to send it. Figure 1.5 shows an example of a WAN, in which, as can be seen, the individual hosts are connected to the various LANs to which is connected a router: the subnet is composed of a set of lines and routers (with the exclusion of hosts).

WANs are usually composed of several transmission lines, each of which connects a router pair. If two routers that are not connected to the same line need to communicate, they use intermediate routers which, if the line is not free, store the packets and then transmit them when the line is free. A subnet that uses this principle of operation is also called *store-and-forward* or *packet-switched*. When the network needs to send a message, the sender divides it into packets and labelling them with progressive sequence numbers that are sent in rapid sequence over the network. The packets travel over the network according to the fastest available paths, being transported individually: upon arrival, the

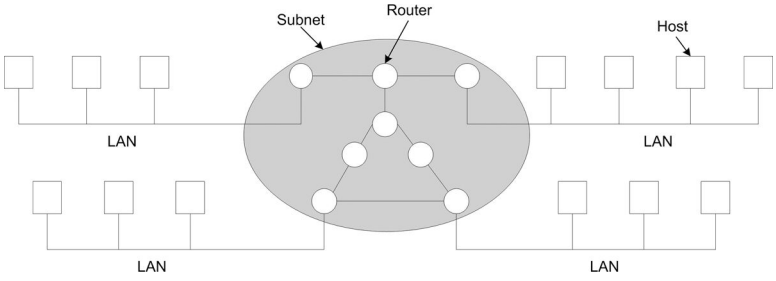


Figure 1.5 Example of LAN connected to a subnet.

receiving host reassembles them according to the sequence number and reconstructs the original message transmitted. Such a mechanism is shown schematically in Figure 1.6.

The routing of packets across the various lines of the subnet is optimised by appropriate algorithms referred to as routing.

1.1.2.4 Wireless networks

Wireless networks allow communications by eliminating the wires and mostly resorting to electromagnetic waves. They can be categorised into three main groups:

1. connections within a system;
2. wireless LAN (WLAN);
3. wireless WAN.

With regard to the first group, it is used to connect a computer with its various peripherals (keyboard, mouse, printer, etc.). The most popular wireless network used in this sense is Bluetooth. Usually, such networks are of the master–slave type where there is a central unit, called master, which connects with the devices, called slaves, determining which addresses to use, when and how to use them.

WLANs are composed of suitable radio devices with antenna connected to various computers. Using these devices, the computer can converse directly in peer-to-peer mode or connect to the fixed network by using a special device called access point (AP). These networks are widely used when cable connections are difficult to provide or in environments requiring the provision of variable user network services. The most widespread standard for WLANs is IEEE 802.11. WLANs operate at speeds of the order of 50 Mbps and distances of the order of tens of metres (Figure 1.7).

Wireless WANs are used over large areas. An example is cellular telephony. From a certain point of view, cellular networks are very similar to WLANs with the difference that they use lower baud rates and greater distances. Mobile phones work at a speed of 1 Mbps and distances in the order of several kilometres. Wireless WANs are, however, at the development stage, characterised by broadband and

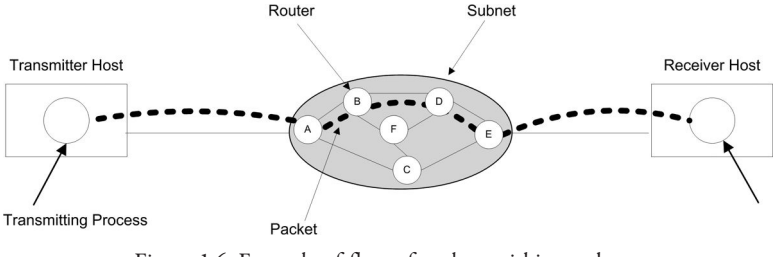


Figure 1.6 Example of flow of packets within a subnet.

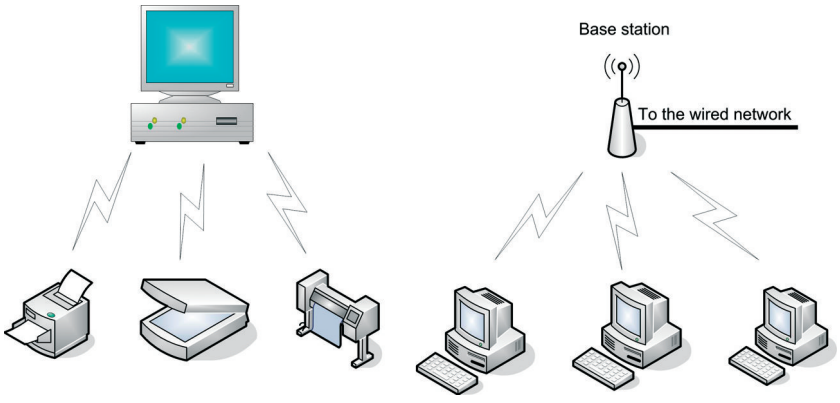


Figure 1.7 Example of wireless connections within a (a) system and (b) wireless LAN.

extended coverage. Just think of high-speed Internet connection, bypassing the traditional telephone system. The standard currently used is the IEEE 802.16.

1.1.2.5 Network between systems

A group of interconnected networks is called Internetwork or Internet. This is composed of a set of networks that mostly use devices called gateways that enable connection and provide related conversion services in terms of hardware and software. A typical example is constituted by a series of LANs connected to a WAN.

1.1.3 Network software

When the first computer networks were created, more attention was paid to the hardware aspect and less to the software. Over time, the importance of developing the software aspect correctly was, however, realised and the basic points are discussed below.

1.1.3.1 Protocols

In order to reduce complexity, networks are organised as a stack of layers or levels, positioned one on top of the other. The various functions and content, as well as their name and the number, vary from network to network. Each layer is designated to provide appropriate services to the higher level layers, without showing them the details of the implemented services, operating as a sort of virtual machine. In a computer, a determined layer is in communication with the counterpart layer located on another machine with which communication is open. A protocol is therefore a kind of agreement between two parties in communication concerning the mode of communication itself. An example of a 5-layer network is shown in Figure 1.8.

In practice, data are not transferred directly between counterpart layers but transferred from the highest to the lowest level until the physical level is reached where it is actually transferred to reach the other computer, where it follows a path from the bottom upwards and vice versa when the flow of communication is reversed.

There is an interface between each pair of layers. If each layer performs a given set of defined functions, these interfaces are clean, minimising the amount of information exchanged between layers and making it possible to replace the implementation of one layer with one that is completely different. The set of layers and protocols is called network architecture.

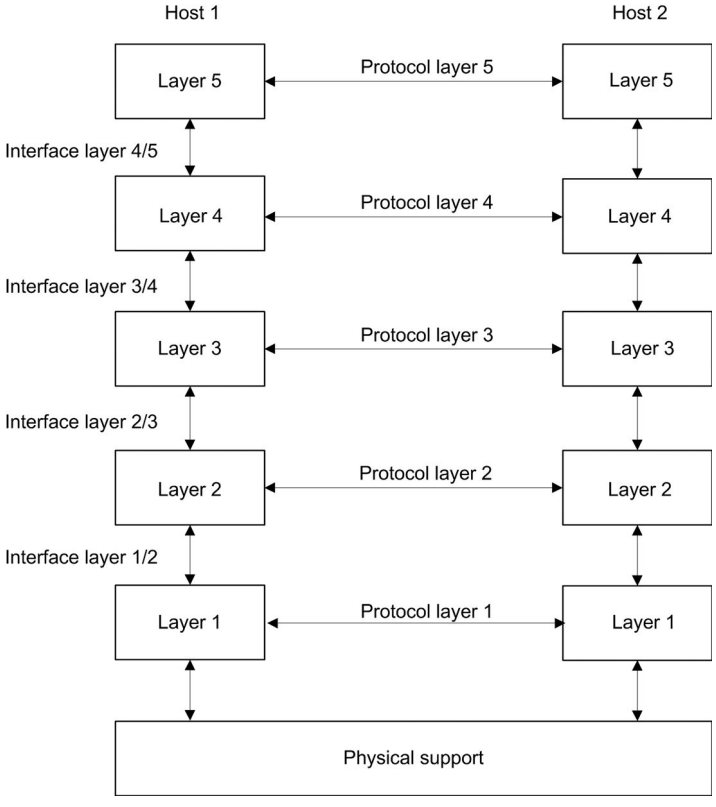


Figure 1.8 Example of layers, protocols and interfaces.

Let us look at a practical example for the 5-layer network shown in Figure 1.8. Suppose that a layer 5 application wants to send a message M to the same application layer of the other machine. Layer 5 passes it to layer 4 that adds an appropriate header to the message M to identify it. Layer 4 passes it to layer 3. Very often layers have limits to the maximum message size that can be handled, as in the case of layer 3. The latter breaks the message into two parts, M_1 and M_2, for example, and adds a suitable header in each of the two parts or packets. Layer 3 decides the output line to be used and sends the packets to layer 2 that not only adds a layer header but also a trailer to each packet and passes it to layer 1 for the physical transmission. Once they have reached the other machine, the packets start to climb the layers, where the trailers and headers are gradually deleted and the original message M is reassembled (Figure 1.9).

1.1.3.2 Layers

Since within a machine, several sources and destinations can operate simultaneously, each layer must be able to identify them uniquely. An addressing system is therefore necessary. A system of rules with regard to the transfer of data must also be established, since the latter can travel in one direction or in both directions, with both normal and high priority. Furthermore, since the physical circuitry is not perfect, errors during communication may be introduced. For this reason, systems of error control, and possibly of error correction, must be introduced. Since not all channels of communication retain the order of the messages sent, protocols must allow them to be reassembled resorting to appropriate sequence numbers. If the sources are too fast in relation to the receivers, the use of an automatic mechanism for reduction of speed, called flow control, is also somehow necessary.

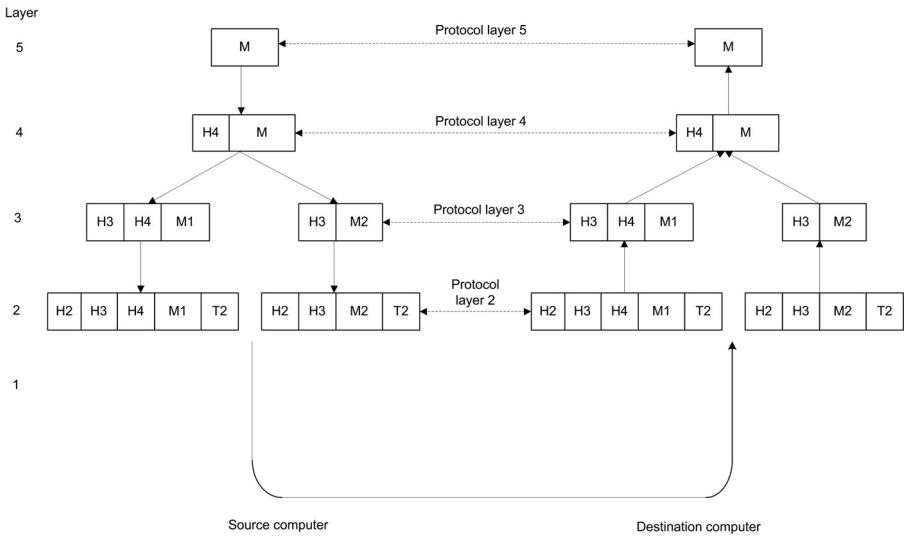


Figure 1.9 Example of communication between layers and different machines.

Another problem is related to the fact that not all processes can accept messages of any length. For this reason, mechanisms of division, separate transmissions and reforming of messages of any length to adapt them to the needs of the process used must be employed. Multiplexing and demultiplexing are often used when an underlying layer must be used by multiple processes above. In most cases, there are multiple paths that connect a source with a destination. In this case, it is necessary to resort to suitable routing algorithms that can be divided between several layers.

1.1.3.3 Services oriented to connection and without connection

The layers are able to provide two types of services: those oriented to connection and those without connection. The service oriented to connection is very similar to a telephone system in which a connection is established, used and released, and in most cases the transmitted data are received, according to the order of transmission, by resorting to a preventive negotiation between transmitter and receiver on the parameters to be used, the maximum size of messages, the quality of the service, etc. Instead, the service without connection is similar to the postal system, in which each message contains the address of the recipient and is sent independently with respect to the others. In this case, the order of sending does not always correspond to the order of arrival.

The services can be categorised on the basis of the quality of the service. They may be reliable, never losing data and using, possibly, confirmation of receipt of the same. This confirmation may in some cases lead to unacceptable delays, as in the case of digital voice traffic. A service without connection devoid of confirmation, that is unreliable, is also called datagram service, similar to the telegraphic system that does not provide any confirmation of receipt. In other cases, reliability is necessary but not the connection: in this situation, reference is made to datagram service with confirmation, similar to the postal service of sending a registered letter with return receipt. When the received message is the response to a datagram that contains a request, this is known as request – reply service (Table 1.2).

1.1.3.4 Service primitives

A service consists of a set of basic operations, called primitives, available to users who need to use the same service. These primitives tell the service how to perform the desired actions.

Table 1.2 Example of various types of service.

Service	Typology	Example
Reliable flow of messages	Connection-oriented	Sequence of pages
Reliable flow of byte	Connection-oriented	Remote login
Non-reliable connection	Connection-oriented	Digitalised voice
Non-reliable datagram	Without connection	Spam email
Datagram with confirmation	Without connection	Registered mail
Request – reply	Without connection	Database query

If the operating system incorporates the protocol stack, the service primitives consist of system calls that cause switching in kernel mode, which permits the taking control of the machine of the operating system to perform the required communication operations. Primitives for a service oriented to connection are different from those of a service without connection (Figure 1.10).

1.1.3.5 Relationship between services and protocols

Services and protocols are different concepts but are often confused with one another. A service is a set of primitives that a layer is able to guarantee to a higher layer. The service says nothing about the implementation of the primitives but only defines the operations that may be offered to the upper layer. The service only refers to the interface between a lower and higher level, where the first is the service provider and the second is the user. A protocol is a set of rules that control the meaning and format of the messages exchanged within the same layer: protocols may be exchanged provided that the service available to users is not changed, making the protocol totally decoupled from the service (Figure 1.11).

1.1.4 Reference models

Two models of reference architectures are illustrated below: the OSI (Open System Interconnection) model and the (Transfer Control Protocol or Transmission Control Protocol/Internet Protocol) TCP/IP model. The first is a general model that is still valid, whose protocols are now in disuse while the second is characterised by a model that is not particularly usable but with protocols that are widely used.

1.1.4.1 The ISO OSI model

The OSI model is based on a proposal by the International Organization for Standardization (ISO) to standardise internationally the protocols used in the various layers. Since it covers the open systems towards communication with others, it is also called ISO OSI model. It is composed of seven layers, as shown in Figure 1.12.

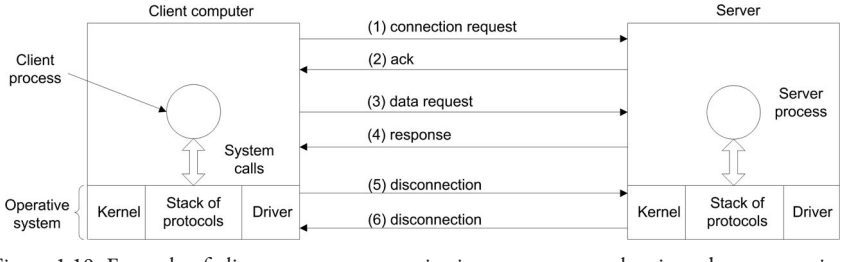


Figure 1.10 Example of client – server communication over a network oriented to connection.

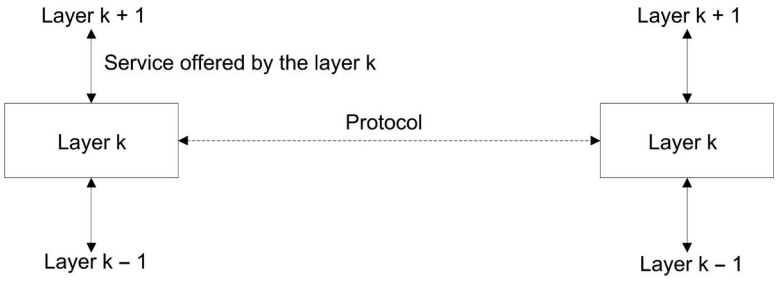


Figure 1.11 Relationship between service and protocol.

The ISO OSI model does not represent a network architecture because it does not specify the services and protocols that are used in each layer: it specifies only what each layer must perform. The features of the various layers are shown below.

The physical layer

The physical layer is concerned with the transmission of bits on the communication channel, ensuring that a bit with a value of 1 is received with a value of 1 and a bit with a value of 0 is received with a value of 0. It can define with which voltage value bits 1 and 0 are defined, how long the impulses should last,

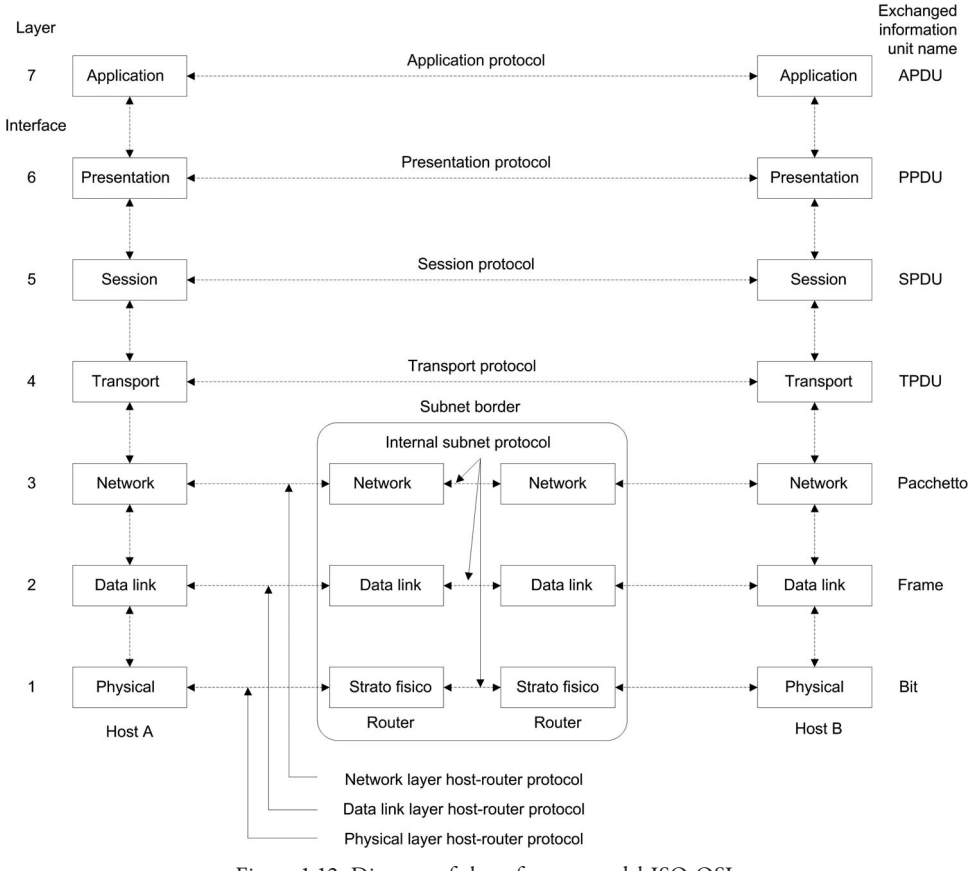


Figure 1.12 Diagram of the reference model ISO OSI.

the possible simultaneity of transmission in both directions, how the connection is established and terminated, the number of contacts that the network connector must have, etc.

It also defines the mechanical and electrical interfaces as well as the physical transmission medium that is used.

The data link layer

The data link layer is responsible for detecting transmission errors and avoiding transmission of the same to the upper level. It forces the transmitter to divide the input data into data frames of variable length between a hundred and a thousand bytes before being transmitted. It also prevents the transmitter becoming saturated when transmission takes place towards a slow receiver, performing what is called a flow control. In broadcast networks, management of access to the transmission channel is also required: this need is managed by a special sub-layer called medium access control (MAC).

The layer network

The network layer deals with management of the subnet by controlling the mode by which the packets are forwarded from the transmitter to the receiver, using suitable tables within the network. It also deals with the handling of possible congestion and quality of the service being offered (delay, jitter, transit time, etc.). Since the routing in broadcast networks is relatively simple, the layer in question is very simple or even absent on such networks.

The transport layer

The transport layer accepts data from the upper layer and divides it into smaller units in order to pass them to the lower layer of the network, ensuring that all the units reach their destination regardless of the hardware used in the lower layers. It also defines the type of service that is provided to the overlying session layer. The layer in question covers the entire path from source to destination, allowing a source computer program to communicate directly with a similar program on the destination computer, unlike the lower layers that deal with the communication of a computer with the more immediate nearby devices that may not be the destination computers.

The session layer

The session layer is concerned with the possibility of establishing an appropriate session on connected computers. It offers various services including control of the dialogue, timing, etc.

The presentation layer

The presentation layer is responsible for the syntax and semantics of the information that is transmitted, as opposed to the lower layers that deal with moving the different bits that make up the same information.

The application layer

The application layer is composed of a series of protocols employed by users including HyperText Transfer Protocol (HTTP) that represents the foundation of the World Wide Web (WWW) or other protocols for email, file transfer, etc.

1.1.4.2 The TCP/IP model

The TCP/IP model represents the parent of all the geographic networks, starting from Advanced Research Projects Agency NETwork (ARPANET) to the current Internet. ARPANET was a geographic network sponsored by the US Defence Department that connected its government installations and universities via leased phone lines. The purpose of this network was to be able to survive the partial loss of the network due to possible external attacks, without interrupting ongoing conversations and ensuring high reliability. The TCP/IP model differs from the ISO OSI model in the absence of certain layers and the amalgamation of some of them, as shown in Figure 1.13.

The Internet layer

The Internet layer of the TCP/IP model is a layer without connection that handles the packet switching network provided by the same model. It allows the various connected computers to send data packets over the network and reaching the destination independently from one another. This layer defines an official format for packets and a protocol called IP. This layer is characterised by functionality similar to the network layer of the ISO OSI model.

The transport layer

The transport layer is responsible for communication between entities of the same level in the source and destination computer. It is composed of two protocols: TCP and User Datagram Protocol (UDP).

The TCP protocol is reliable and connection-oriented. It divides the flow of bytes to be sent in messages of predefined length that are sent to the Internet layer in order to reach the final destination without errors. The TCP process of the receiver machine reassembles the original flow. The TCP protocol also manages flow control to avoid overloads of a slow receiver from a fast source.

The UDP protocol is unreliable without connection and is used by applications that do not require this service. It is used for typical question–answer applications as in the client – server mode of operation and in applications where speed with respect to reliability is preferable, as in audio or video transmission (Figure 1.14).

The application layer

The TCP/IP model is characterised by the absence of the session and presentation layers, for which reason above the transport layer is the application layer containing all the higher level protocols such as Terminal Emulation Link NETwork (TELNET) for the management of a virtual terminal, File

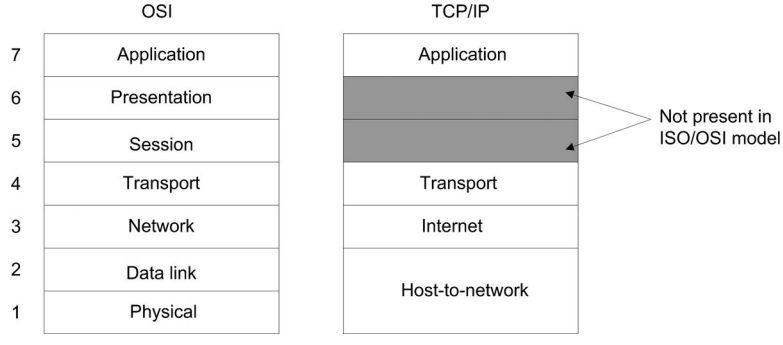


Figure 1.13 Comparison between the reference model ISO OSI and TCP/IP model.

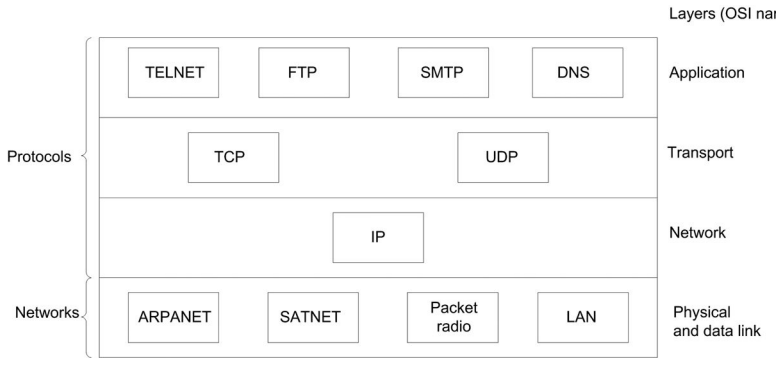


Figure 1.14 Protocols in TCP/IP model.

Transfer Protocol (FTP) for file transfer, Simple Mail Transfer Protocol (SMTP) for email, Domain Name Server or Domain Name System (DNS) that performs network name-address matching, and HTTP to surf the WWW.

The host-to-network layer

The TCP/IP model, in the Internet layer does not have well-defined specifications, merely stating that the computer must use a network protocol that allows transmission in IP packets.

This undefined protocol varies from machine to machine.

1.1.4.3 Comparison between the ISO OSI and TCP/IP models

The ISO OSI and TCP/IP models of course have many points in common. In fact, they are characterised by very similar layer models. But they are also characterised by certain differences. For example, the ISO OSI model draws a distinction between the concepts of service, layer and interface while the original TCP/IP model does not. This difference is due to the fact that in the ISO OSI model, the model itself was initially developed followed by the protocols of implementation while in the TCP/IP model the case is exactly opposite. Moreover, a fundamental difference lies in the number of layers: seven for the ISO OSI model and four in the TCP/IP model. There are, however, further differences that will not be discussed for the sake of brevity, referring the reader to specific texts on the subject provided in the bibliography.

1.1.5 Examples of network

There are currently many types of networks, of variable sizes, with different aims, purposes and technologies. The most famous network used by the general public is of course the Internet but there are also other types of network, such as those that are connection oriented, that are of great importance in the field of telecommunications. A brief illustration of these networks is as follows.

1.1.5.1 Internet

The Internet, more than being just a network, is composed of a set of different networks that use specific protocols and provide common services. It does not have a designer and is not controlled by any one subject. To fully understand its nature, the genesis of this network must be briefly explained.

ARPANET

The history of ARPANET began around the 1950s in the midst of the cold war when the US Defence Department decided to implement a network of command and control that would be able to survive a possible nuclear war, as all the communication of the time took place using the normal telephone network, which was considered too vulnerable: in fact, all that was needed was an attack on a few upper level switching centres to completely isolate the network, as shown in Figure 1.15(a).

The use of a packet switching digital technology was proposed, as shown in Figure 1.15(b), characterised by greater reliability and redundancy, but the US monopolist operator at the time completely rejected this proposal. It was only in the 1960s that it was demonstrated that the packet switching network could operate as it already connected a number of computers at the National Physical Laboratory in Great Britain and a tender was submitted for construction of the first subnet. The experimental network began in 1969, connecting only four junctions but soon all of the United States was connected.

In 1974, the TCP/IP protocol to manage communications on Internetwork was developed since different networks began to connect to ARPANET. Around the 1980s, the network became increasingly complex, connecting to each other a large number of computers and the DNS was created to associate host names with network addresses in order to reach them.

NSFNET

In the 1970s, the US National Science Foundation (NSF) noted the enormous impact on university research that ARPANET had and also became aware of the impossibility, for anyone who did not have a research contract with the ministry of defence, of accessing the same. For this reason, it designed and built a national network (NSFNET) composed of six supercomputers that were flanked by machines of reduced performance that used the TCP/IP protocol. NSF then financed about 20 regional networks to enable connection to the main network of various universities, research centres, libraries and museums. NFSNET was so successful that it was overloaded from the first day of operation by using 56 kbps leased twisted pairs.

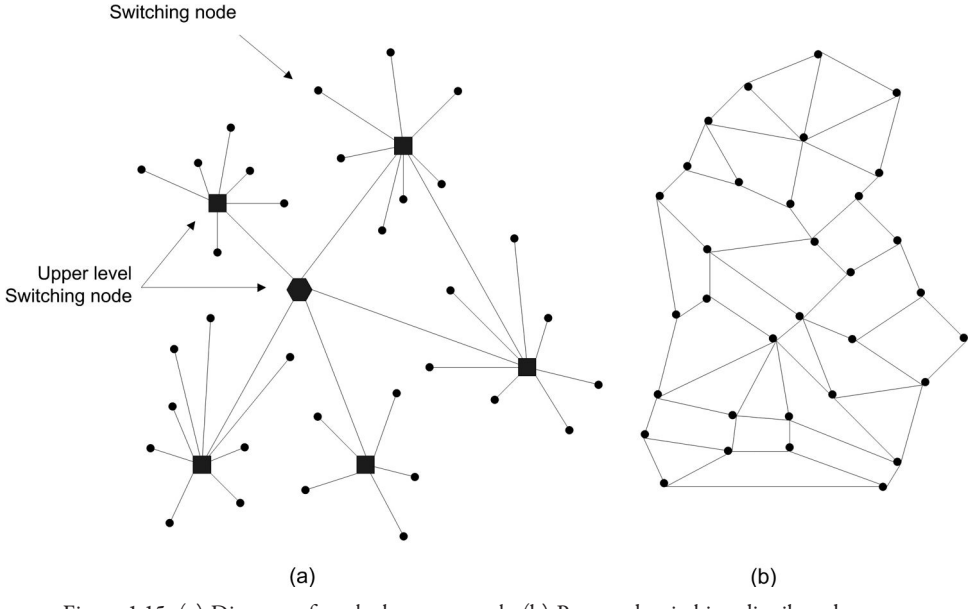


Figure 1.15 (a) Diagram of a telephone network. (b) Proposed switching distributed system.

As the government was unable to further fund the network, a non-profit company called Advanced Networks and Services (ANS) was formed that was be the first step towards commercial applications of the network. In the 1990s, ANS took over NFSNET and increased the speed of the then current connections from 1.5 Mbps to 45 Mbps to create the new ANSNET. Subsequently, various countries created their networks on the model of ARPANET and NSFNET such as the European network EuropaNET and EBONE, which were subsequently sold to industry.

Internet use

In 1983, the TCP/IP protocol became the official protocol of ARPANET and with the merger of the same with NSFNET, the number of users grew at a dizzying speed. Midway through the 1980s, the set of networks began to resemble the current Internet without, however, there being an official commencement date. Currently, using the Internet means using a computer that employs the TCP/IP protocol and has an IP address and can send or receive IP packets over the network. Many users employ temporary connections, connecting to an ISP via a modem, which assigns them a temporary IP address to use the Internet.

From the 1970s to the 1990s, the Internet was mainly used for email, news, remote login and file transfer, being mostly used by researchers, authorities and industrialists.

The introduction of the WWW allowed access to the Internet for a very large number of non-academic users, as it allowed the use of pages containing text, images, sound and video, and links to similar pages, demonstrating that it was a great support for the distribution of information of any kind, including commercial information. A strong impetus to the growth was provided around the 1990s by companies called ISP that enabled individual users, through a normal telephone call by modem, to connect to the Internet.

Internet architecture

Currently, the Internet has a very varied structure as a result of mergers between telephone companies and ISPs. A general layout is shown in Figure 1.16.

When a normal user wants to connect to the Internet, he uses an analogue modem or an asymmetric digital subscriber line (ADSL) connection (which will be discussed further on in more detail) to reach the point of presence (POP) and subsequently the ISP where the data are entered into the regional network in digital form and via the mechanism of packet switching. The regional network is formed of various interconnected routers and present in various localities of ISP competence. If packets are to be exchanged between users that are connected to the same ISP, they remain within the regional network otherwise they are inserted onto the main network (backbone), property of large global operators, composed of a large number of routers connected to each other through broadband fibre – optic networks. Companies that provide Web services have groups of servers (server farm) directly connected to the backbones to reduce delivery times of the requested services. Business LANs can be directly connected to networks. Since data packets may need to pass from the backbone of one owner to that of another, the latter are interconnected by systems called NAPs, which are none other than routers connected to each other and to a backbone.

1.1.5.2 Connection-oriented networks

Since the beginning of the development of networks there has been some contention between networks not oriented to connection that use datagrams, such as ARPANET, and those that are connection-oriented.

On the other hand, the priority of ARPANET was to withstand attacks that would have resulted in the partial loss of the same so that its operation specification was targeted at the resistance of failures.

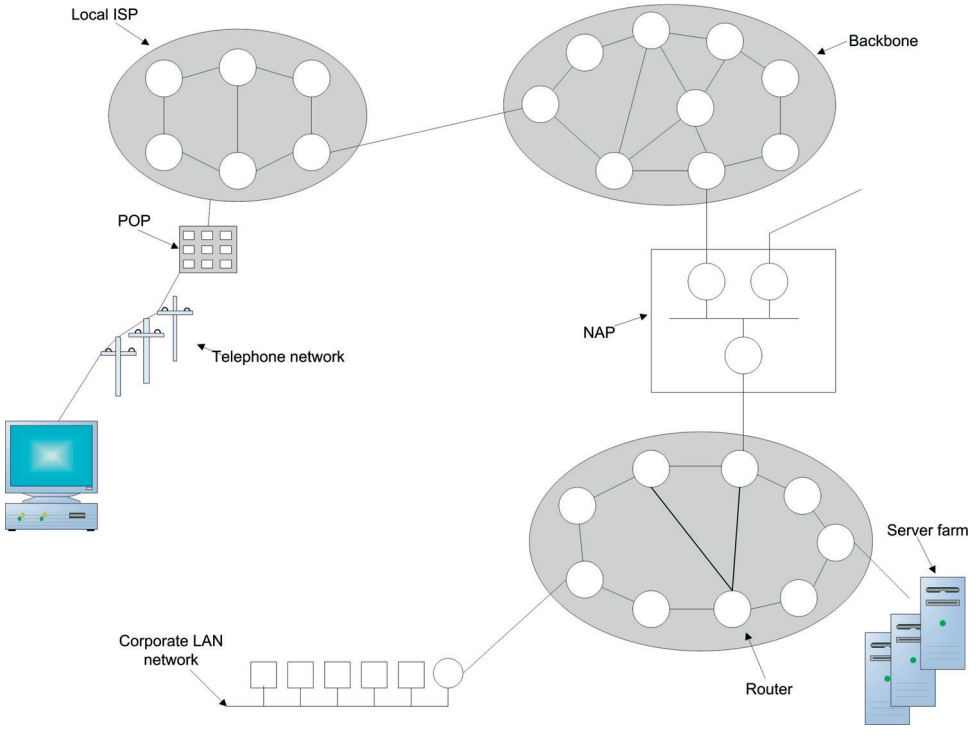


Figure 1.16 General layout of the Internet.

The world of telephone companies instead prefers connection-oriented networks, as is obvious, where the connection is opened, maintained until the sending of all the data on the same line and finally released. The loss of a component that affects connection during the call causes the loss of the call itself, contrary to what happens with ARPANET, where the network reconfigured to send in data through alternative routes. Telephone companies prefer connection-oriented networks for two reasons: better quality of service and the option of invoicing.

X. 25 and frame relay

The first connection-oriented public data network was created in the 1970s and was called X.25. The transmitting computer, in order to use X.25, first had to open a data connection to which was assigned a connection number used for the transmission of data packets. X.25 networks remained in service for about 10 years.

X.25 networks were gradually replaced in the 1980s by frame-relay networks that were connection-oriented networks without flow or error control. Because they were connection-oriented networks, packets reached the recipient in the same sequence in which they were transmitted. They are particularly suited to managing WAN networks and are still used today to connect the LANs of variously located offices.

ATM

Asynchronous Transfer Mode (ATM) technology was developed in the 1990s that, unlike synchronous telephone networks, is in fact asynchronous. ATM networks are able to meet all communication needs as they are capable of unifying voice, data, TV, etc. within a single integrated

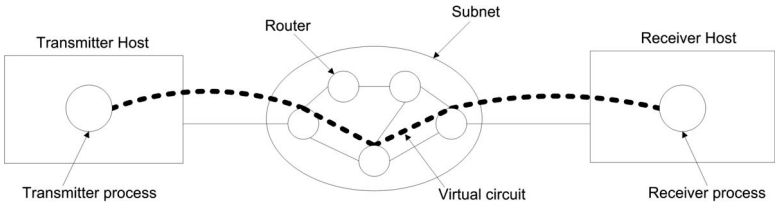


Figure 1.17 Example of ATM virtual circuit.

system. These are therefore mostly used by telephone companies for the internal use of IP packet transport.

Since this network is connection-oriented, before commencement of communication, a suitable configuration packet must be sent over the same network that, travelling across the network from source to destination, reserves resources for the connection to then open what is defined a virtual circuit, through which pass small packets that are easy to handle and to route in virtual circuits. ATM networks are, in practice, WANs with typical speeds of 155 and 622 Mbps but higher speeds are also available (Figure 1.17).

Ethernet

If, on the one hand, the Internet and ATM are networks designed to cover large areas, there also exists a need to establish communication between computers over lesser distances for the creation of LAN. Within this context, Ethernet technology was created and has been immensely successful and gained widespread popularity. This technology was developed in the 1970s by Xerox's Palo Alto Research Centre (PARC).

It uses a coaxial cable and links several computers connected to the cable in parallel, and between each other, using transceivers, in a mode called multi-drop, as shown in Figure 1.18.

Each computer, before transmitting, listens to see if there are other transmissions in progress on the cable: if there are, it waits for the end to then start transmitting. Each computer listens to its own signal being transmitted to check for possible interference with another computer that is transmitting at the same time. If interference is detected, transmission is interrupted and a period of time elapses before retransmitting. If at the second attempt a collision again takes place, transmission is interrupted again and a period of double time elapses, to allow other transmitters to first send their data, and so on in any subsequent attempts. Ethernet is in continuous evolution, with respect to its original version, and currently reaches baud rates greater than 1,000 Mbps.

There are, however, other standards such as token bus and token ring, where packets (token) are exchanged on bus (token bus) or on a ring (token ring) between computers in a sequential manner

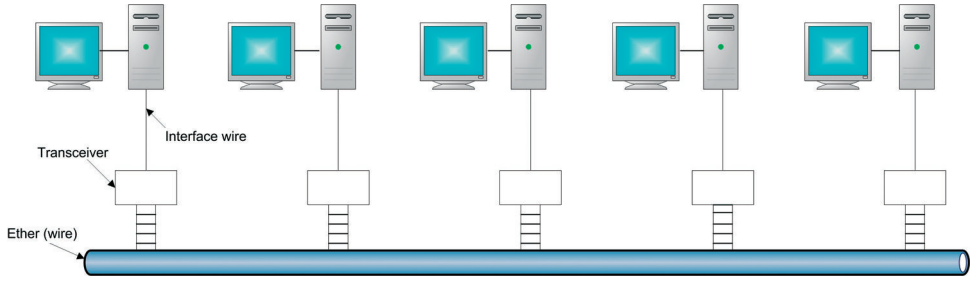


Figure 1.18 Original architecture of an Ethernet system.

which, if empty, are filled with the data to be sent and with the address of the recipient, thus avoiding collisions. However, Ethernet has established itself as the standard reference on the market.

1.1.5.3 Wireless LAN

WLANs are networks that do not require the presence of cables (hence the name wireless) and use low-power radio frequency transceivers. They provide communication between computer and computer or between computer and related peripheral devices without the need to resort to physical connections between the same. Initially, there were different standards that did not particularly facilitate connection between the various devices and, thus, the Institute of Electrical and Electronics Engineers (IEEE) committee was given mandate (which will be discussed below) to standardise WLANs. The standard introduced was called 802.11, better known to the general public as WiFi. This can operate in the presence or absence of a radio base station. In the presence of a radio base station, called AP, all communications pass through it. In the absence of a radio base station, all the machines converse between each other and this is referred to as ad hoc mode (or ad hoc networking) (Figure 1.19).

For historical reasons, since Ethernet was already a consolidated reality at the time of development of the standard in question, it was decided to make it compatible with Ethernet over the data link layer, given that there are the typical problems of mobility in the physical layers and data link itself. In fact, the Ethernet standard requires the computer to listen before beginning to transmit but in the case of wireless it does not always mean a successful transmission. In fact, let us consider the situation as shown in Figure 1.20.

In this situation, computer A reaches computer B but not computer C that instead only reaches computer B. If A wants to transmit to B, it is not able to listen if computer C is transmitting towards B, interfering with any transmission in progress. In addition, the transmitted signal can reach the receiver through multiple reflections on objects in the environment, subject of the communication (multi-path fading) and the protocol must also take this possibility into account. Another problem is due to the fact that the majority of software is not designed for mobility. This means that a number of devices available in a certain environment may not be available if the computer is moved to another environment. The same problem occurs when a computer is moved from an environment served by an AP to another environment served by an AP; this problem is solved, for example, with cellular telephone systems. In this case, use is made up of multiple APs connected to an Ethernet network, which is connected via a specific portal to the external network (Figures 1.21 to 1.24).

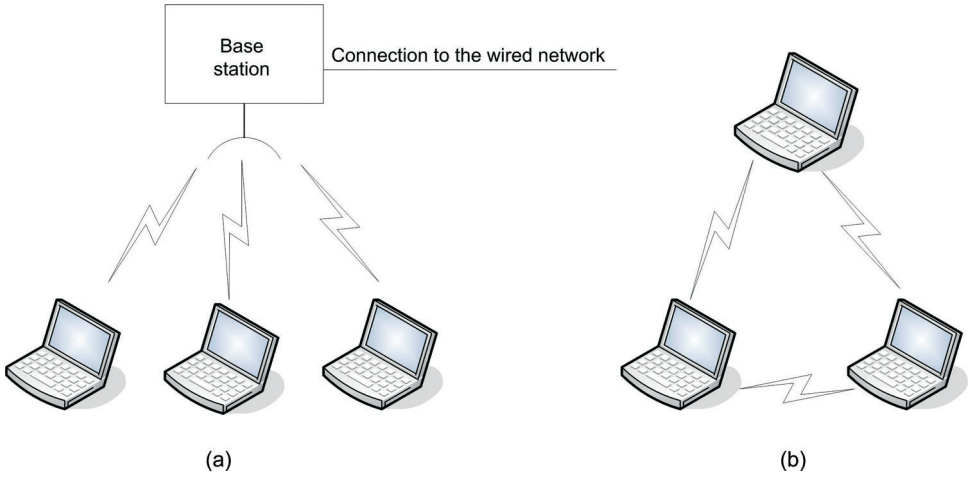


Figure 1.19 Example of wireless (a) with radio base station and (b) without radio base station.

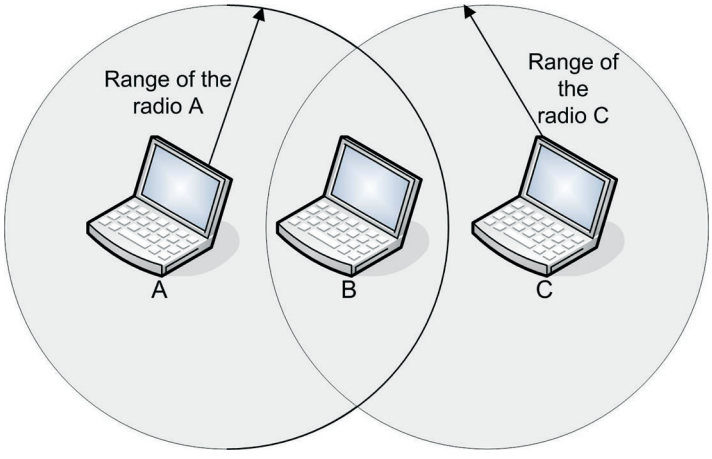


Figure 1.20 Example of how the scope of individual units is unable to cover the entire network.

These problems were solved in 1977 with the production of a standard capable of operating at 1 or 2 Mbps. Since this speed proved immediately insufficient, the committee divided in two in 1999, proposing two different standards: 802.11A that uses a wider frequency band and is able to reach 54 Mbps and 802.11b that uses the same frequency band of the first but a different type of modulation that allows it to reach 11 Mbps. The 802.11g standard that uses the modulation techniques of 802.11a

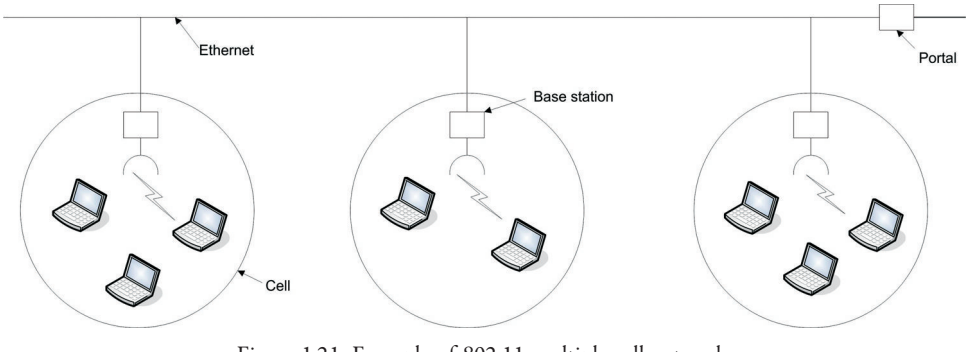


Figure 1.21 Example of 802.11 multiple cell network.

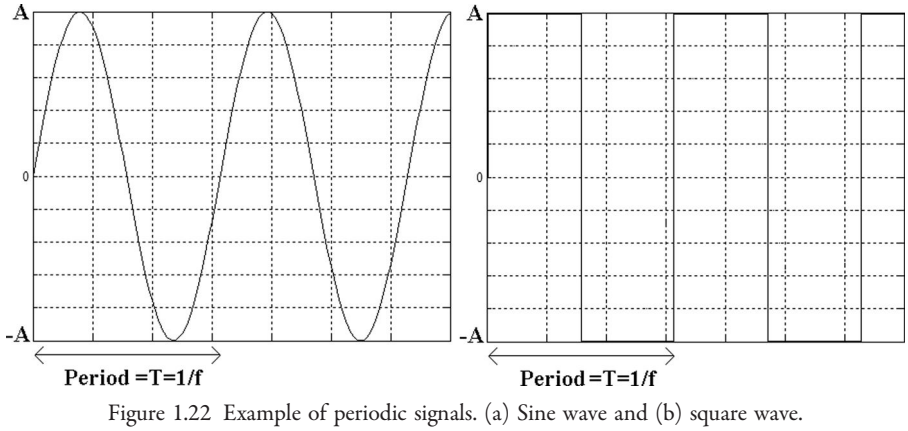


Figure 1.22 Example of periodic signals. (a) Sine wave and (b) square wave.

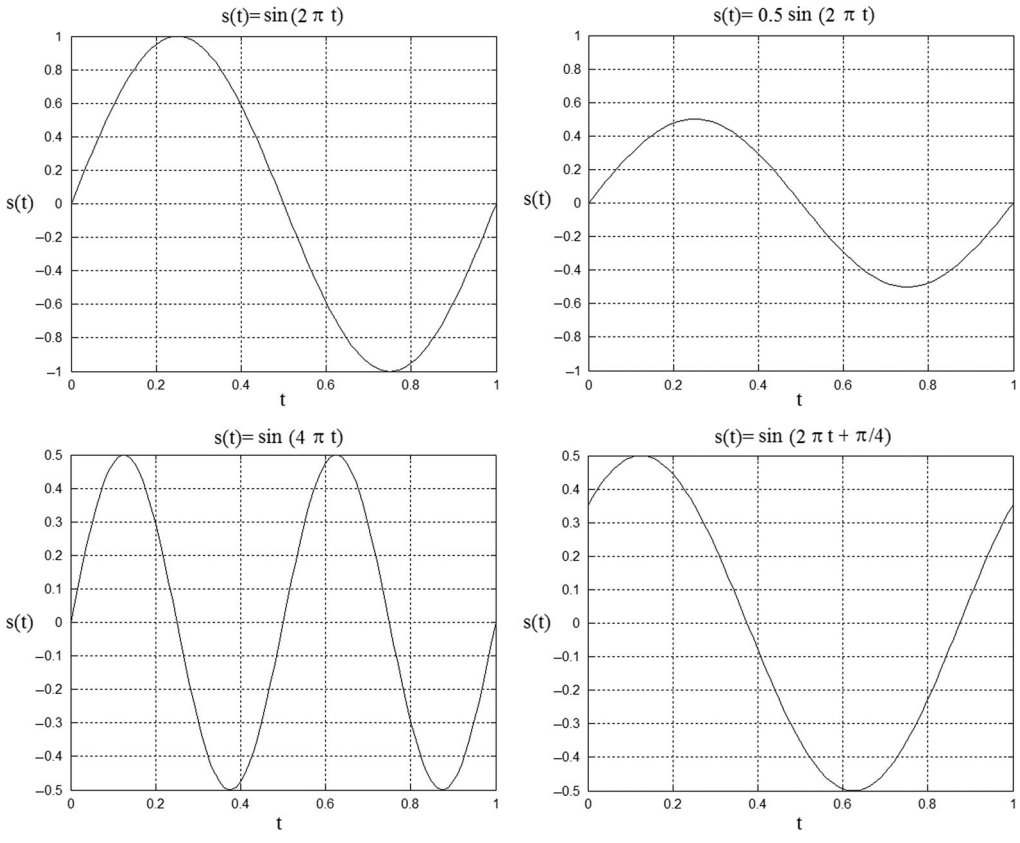


Figure 1.23 Example of various sinusoidal signals characterised by different parameters.

and the frequency band of 802.11b was subsequently proposed. The 802.11 standard started a revolution in the field of information technology, making access to the Internet fully mobile.

1.1.6 International entities of the telecommunications world

The world of telecommunications is full of entities working under various titles and levels: telephone companies, providers, manufacturers, governments, regulatory and standardisation bodies, technical and scientific associations, etc.

Already by 1865, many European governments had joined to create the grandfather of the current International Telecommunication Union (ITU) conceived to standardise communications at a global level, at the time represented by telegraphy. In 1947, ITU became an agency of the United Nations and standardising the then growing telephony sector. It works in three main sectors: radio communication (ITU-R), telecommunication standardisation (8ITU-T) and development (ITU-D). From 1956 to 1993, ITU-T assumed the name CCITT (an acronym derived from the French Comitè Consultativ International Telegraphique et Telefonique) that issued several recommendations. ITU-T is composed of national governments, industry members, associate members and regulatory bodies. Its recommendations are only suggestions that the various governments may or may not accept.

Then, there is the ISO, a voluntary organisation founded in 1946 that produces and publishes international standards and comprising the standardisation bodies of the participating countries. ISO is an active member of ITU-T working to avoid the production of differing international standards in the world of telecommunications. American National Standards Institute (ANSI) is the US

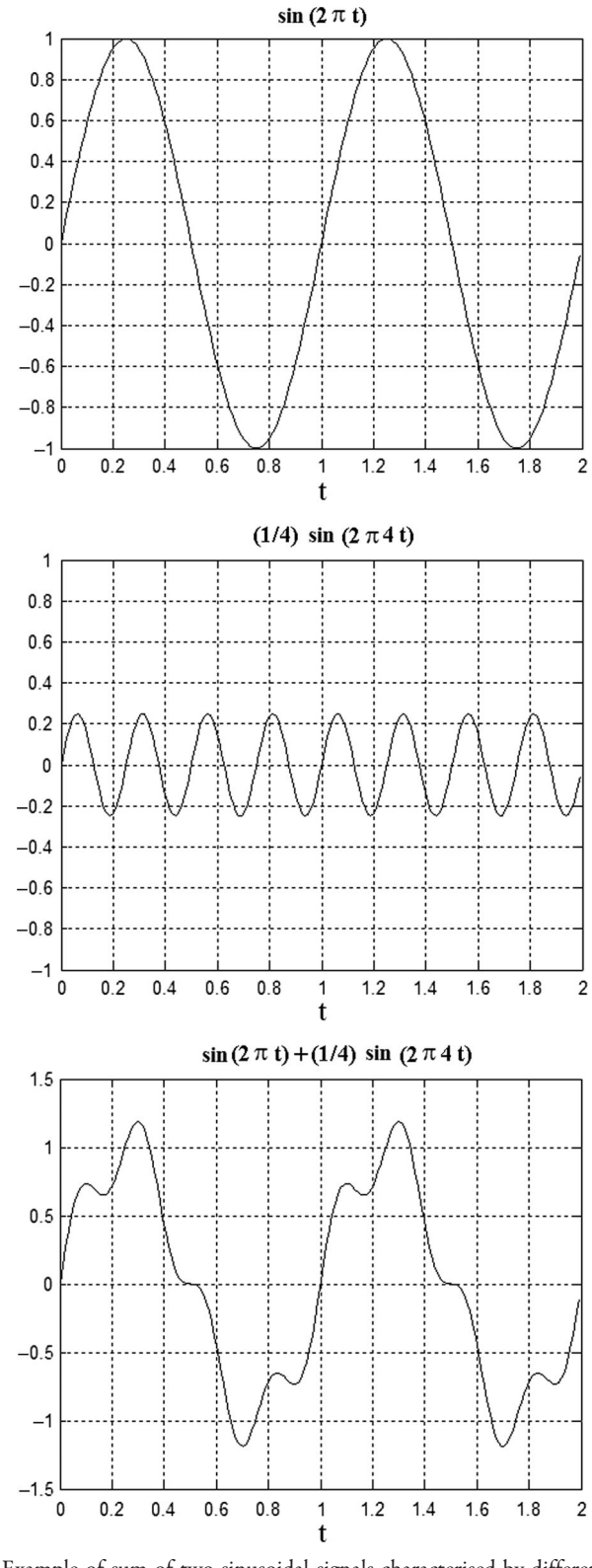


Figure 1.24 Example of sum of two sinusoidal signals characterised by different parameters.

representative at ISO. Then, there is IEEE that is the largest professional association in the world. In addition to producing publications, books and organising conferences throughout the world, it has its own internal standardisation group. For example, committee 802 has standardised different types of LAN, as we have seen, and is still operational (Table 1.3).

With regard to the Internet, in 1983, a committee called Internet Activities Board (IAB) was created that was then renamed Internet Architecture Board. In 1989, IAB was reorganised, the researchers were moved to Internet Research Task Force (IRTF) and the complementary Internet Engineering Task Force (IETF) was created. The Internet Society was also subsequently created. IRTF focuses on long-term research while IETF focuses on short-term technical problems.

1.2 The physical layer

The physical layer represents the lowest level of the ISO OSI model. It defines the mechanical and electric interfaces, timings, encoding of the signal, amplification and regeneration and definition of the transmissive specifications of the medium in terms of networks. The physical layer receives from the upper level a set of bits or bytes and transmits it on the transmission medium such as a stream of independent bits. In the following, after a brief overview of the theory of signals, the various modes of communication in the different physical media currently used in telecommunications are examined.

1.2.1 Signals theory

The signals related to information are transmitted by varying the physical properties of a number of characteristic quantities. In cables, either the voltage or current is typically changed, a function $s(t)$ that is variable over time is associated, in order to model the signal and address it mathematically.

Table 1.3 IEEE 802 working groups.

Number	Argument
802.1	General aspects and architecture of LANs
802.2	Logical link control
802.3	Ethernet
802.4	Token bus
802.5	Token ring
802.6	Dual queue dual bus
802.7	Technical advisory group for wide band technologies
802.8	Technical advisory group for optical fibre technologies
802.9	Isochrone LAN
802.10	Virtual LAN and security
802.11	Wireless LAN
802.12	Demand priority
802.13	Not used for superstitious reasons
802.14	Cable modem
802.15	Personal area network (Bluetooth)
802.16	Wide band wireless
802.17	Resilient packet ring

1.2.1.1 Periodic signals: Fourier series

In data transmission, periodic signals, which have particular importance, are signals that are repeated periodically over time. The characteristics of these signals are the amplitude A, that is the maximum level of the signal, the period T, that is the time interval of the periodicity, the frequency *f*, that is the inverse of the period and phase φ, that is the measurement of the relative position of the signal at a given instant.

Through sinusoidal signals, in physical media, the following are also defined:

1. Wavelength λ that is the distance in metres between two points of equal phase in adjacent periods (for example, the distance between two crests or two troughs of a wave).
2. Propagation speed v that is the speed with which a wave crest or a trough moves in space.

The wavelength, frequency and speed are related by the following ratio:

$$v = \frac{\lambda}{T} = \lambda f \tag{1.1}$$

The speed of electromagnetic waves is approximately 3×10^8 m/s in a vacuum and about 2×10^8 m/s through copper. Visible light has a frequency within the range 10^{14} to 10^{15} Hertz (Hz) for which the range of wavelength, using equation (1.1), is 10^{-6} to 10^{-7} m. The sum of sine waves whose frequencies are multiples of one of these is still a periodic signal. The lowest f_0 frequency is called fundamental. The frequency $f_n = nf_0$ is called nth harmonic. The frequency of the resulting signal is equal to the fundamental frequency.

In general, a signal transmitted in a certain way, upon receipt, is different due to effects attributed to the transmission. Since not all signals are sinusoidal or periodical, it is of fundamental importance to address any signal in terms of defined frequency sinusoidal signals. In this way, we are assisted by the mathematical theory developed by Fourier that allows us to consider signals as the sum of sinusoidal signals.

Given a periodic function $s(t)$, of period T, continues with continuous section and limited derivative, it can be written as a sum of sinusoidal and cosinusoidal functions as:

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(2\pi n f_0 t) + \sum_{n=1}^{\infty} b_n \sin(2\pi n f_0 t) \tag{1.2}$$

where $f_0 = 1/T$ is the frequency of the function.

The coefficients a_0, a_n and b_n are calculable from the following relations:

$$a_0 = \frac{2}{T}\int_0^T s(t)dt \tag{1.3}$$

$$a_n = \frac{2}{T}\int_0^T s(t)\cos(2\pi n f_0 t)dt \tag{1.4}$$

$$b_n = \frac{2}{T}\int_0^T s(t)\sin(2\pi n f_0 t)dt \tag{1.5}$$

The Fourier series can also be written in a more general form, valid for complex functions, such as:

$$s(t) = \sum_{n=-\infty}^{\infty} c_n e^{j2\pi n f_0 t} \tag{1.6}$$

where

$$c_n = \frac{1}{T}\int_0^T s(t)e^{-i2\pi n f_0 t}dt \tag{1.7}$$

being $i = \sqrt{-1}$.

1.2.1.2 Non-periodical signals: the Fourier integral

Even if it is not properly correct, a non-periodic signal can be thought of as a periodic signal tending to infinity. In this case, the fundamental frequency, and therefore the distance between the harmonics, is reduced to zero. When employing the representation of the signal through the Fourier series, it would be represented by a sum of frequencies increasingly closer to each other as the period increases. In this case, the Fourier series transforms from sum to integral. In order that this may be possible, the function $f(t)$ must be integratable, non-periodic and such that it satisfies the following condition:

$$\int_{-\infty}^{+\infty} |f(t)|^2 < \infty \tag{1.8}$$

In this case, the following function is defined transformed by Fourier $S(f) = F[s(t)]$:

$$S(f) = F[s(t)] = \int_{-\infty}^{+\infty} s(t)e^{-i2\pi \mathrm{ft}}dt \tag{1.9}$$

From the Fourier Transform $S(f)$, it is possible to trace back to the original function $s(t)$ using the Fourier anti-transform:

$$s(f) = F^{-1}[S(t)] = \int_{-\infty}^{+\infty} S(f)e^{i2\pi \mathrm{ft}}dt \tag{1.10}$$

The Fourier transform is a function of frequency and represents the contribution of the different frequencies to the signal in a manner similar to the coefficients of the Fourier series for the periodic signals that define the contribution of the various harmonics to the signal amplitude. The spectrum of a signal is not periodic and continuous, that is all frequencies contribute to the amplitude of the signal unlike the spectrum of a periodic signal that consists of a discrete set of frequencies, called harmonics.

A very important function in the theory of signals is represented by the Dirac δ(t) pulse function or mathematical pulse. It is used to represent approximately phenomena such as the high and narrow peaks of certain functions or their discontinuity. It is defined as:

$$\delta(t) = 0 \text{ for } t \neq 0, \int_{-\infty}^{+\infty} \delta(t)\mathrm{d}t = 1 \tag{1.11}$$

Its graphical representation is shown in Figure 1.25 together with other transforms of interest in Figures 1.26 to 1.28.

1.2.1.3 Spectral representation of signals

The graph of the amplitude of a signal in relation to the frequencies that it is composed of is called spectral representation. The rows of the spectral representation indicate the spectral contribution to the amplitude of the signal due to the relative frequencies. If the signal has a non- zero mean value, that is the coefficient a_0 is not zero, it has a continuous component, that is a zero frequency component (Figures 1.29 to 1.33).

It has been seen in the previous sections that a periodic function is expressible as the sum of sinusoidal functions at frequencies that are integer multiples of the frequency of the signal $s(t)$. It

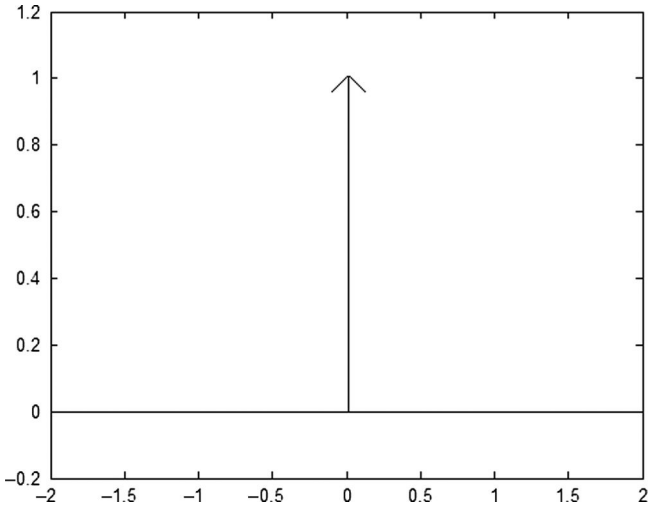


Figure 1.25 Graphical representation of the Dirac pulse function or mathematical pulse.

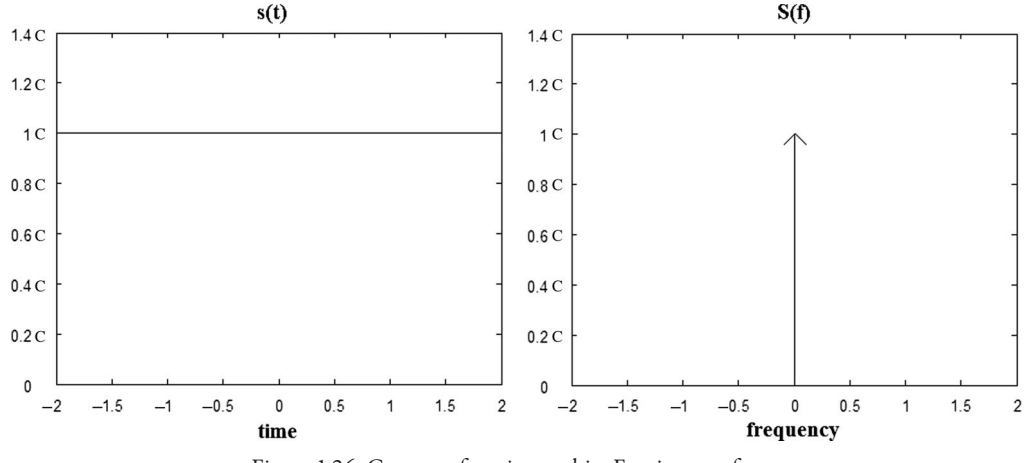


Figure 1.26 Constant function and its Fourier transform.

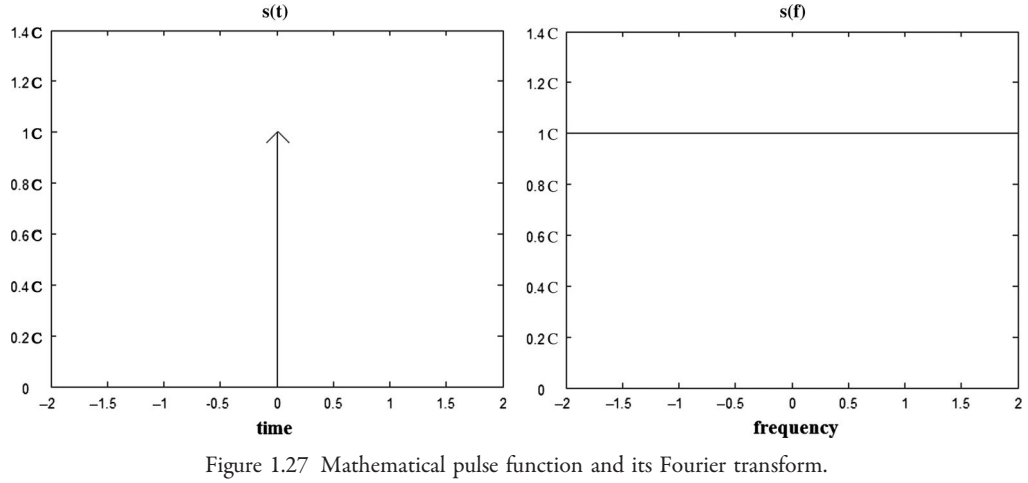


Figure 1.27 Mathematical pulse function and its Fourier transform.

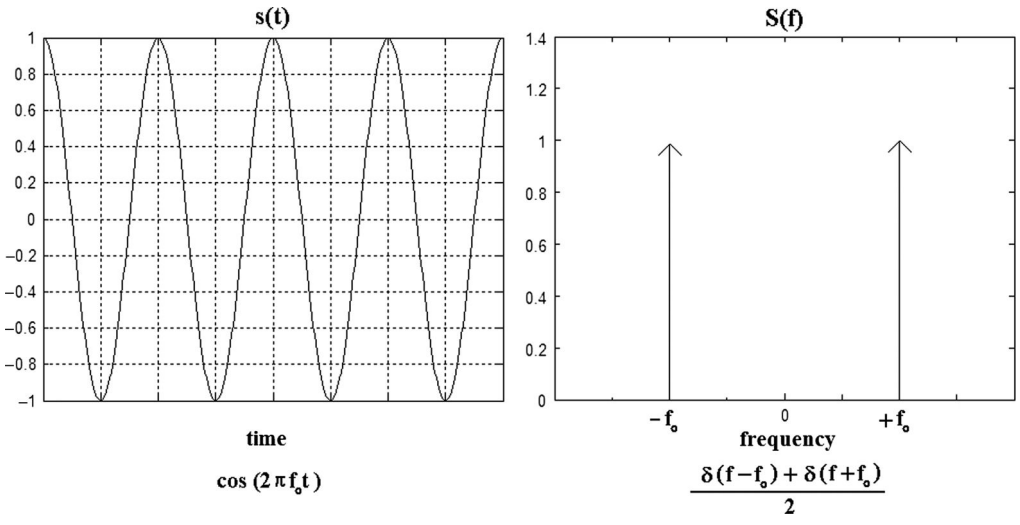


Figure 1.28 Cosinusoidal function and its Fourier transform.

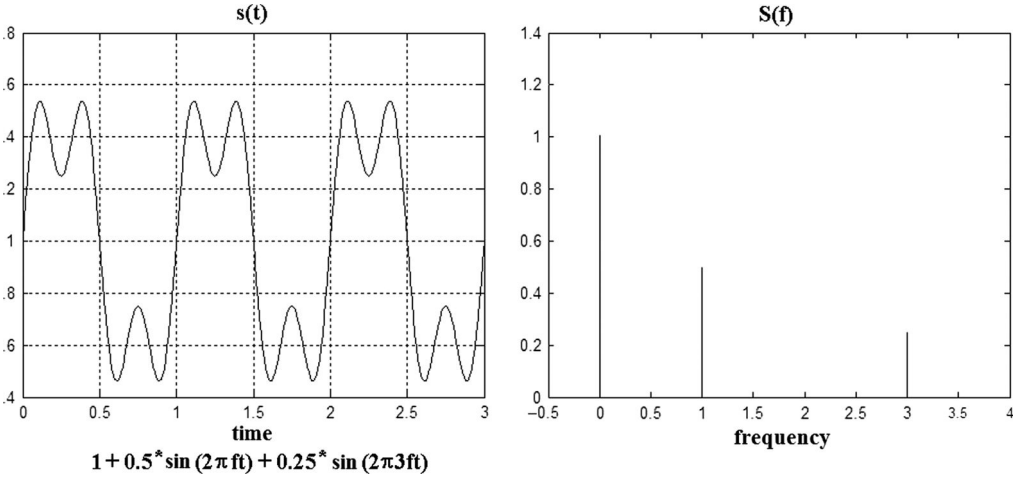


Figure 1.29 Example of periodic signal and its spectral representation.

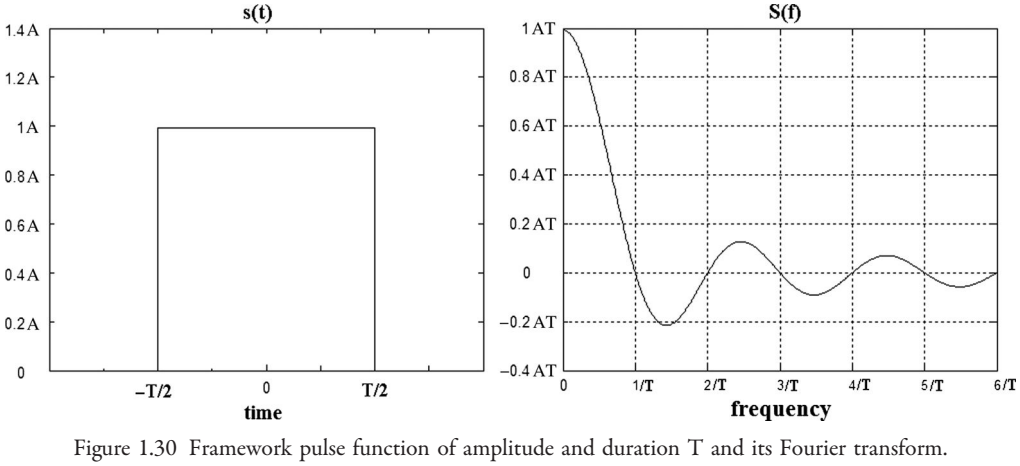


Figure 1.30 Framework pulse function of amplitude and duration T and its Fourier transform.

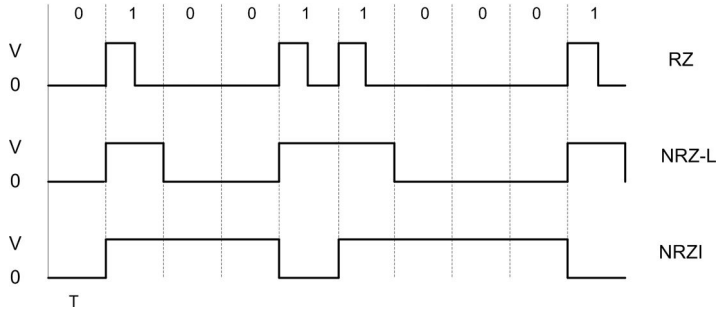


Figure 1.31 Unipolar encodings RZ, NRZ-L and NRZI.

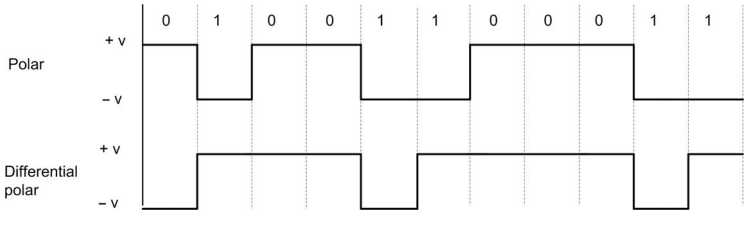


Figure 1.32 Simple and differential polar encodings.

therefore comprises a discrete set of frequencies. It has also been seen that a non-periodic function can be expressed as an integral of sinusoidal functions. Since its components may have any frequency, it is characterised by a continuous spectrum. Consider, for example, the framework pulse function of amplitude and duration T, which is a function equal to A when $-T/2 \leq t \leq T/2$ is 0 elsewhere. Its Fourier transform $S(f)$ is equal to:

$$S(f) = AT\frac{\sin(\pi ft)}{\pi ft} \tag{1.12}$$

As can be seen, its spectrum is infinite descending, its transform being concentrated into lobes of 1/T spectral frequency. Most of the spectrum is concentrated within the first lobe.

1.2.1.4 Power of periodic signals

The average power P of a periodic signal $s(t)$ is defined the amount:

$$P = \frac{1}{T}\int_0^T |s(t)|^2 dt \tag{1.13}$$

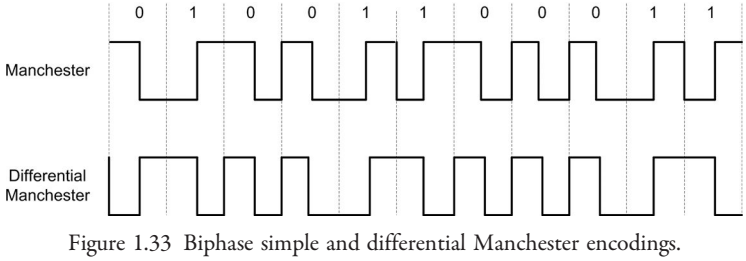


Figure 1.33 Biphase simple and differential Manchester encodings.

Using the Fourier transform, it can be demonstrated that the average power P of the periodic signal $s(t)$ is equal to:

$$P = \sum_{n=-\infty}^{\infty} |c_n|^2 \left(\frac{a_0}{2}\right)^2 + \sum_{n=1}^{\infty} (a_n^2 + b_n^2) \tag{1.14}$$

Often the spectral representation is made by graphing the module of the Fourier coefficients of the development and highlighting the contribution to the power of the signal due to the different harmonics.

1.2.1.5 Bandwidth of a signal and of a transmission channel

The bandwidth of a signal is given by the range of frequencies that make up its spectrum. Signals are usually characterised by an infinite band even if the majority of the power is contained in a small set of frequencies, as has been seen, for example, in the case of the framework pulse of amplitude A and duration T. This small range of frequencies is the effective bandwidth of the signal.

When a signal is transmitted, it is not practically possible to transmit all the frequencies that it is composed of as the transmission medium, the technology that generates the signal or the voluntary choices, impose band limits on the signal itself. As a limited number of harmonics is transmitted, the received signal is different from that transmitted at source: the greater the number of harmonics transmitted, the greater is the similarity between the signal transmitted and the signal received.

If all the Fourier components were attenuated in the same manner, the signal from it would be reduced in amplitude but not distorted. However, in reality, the amplitudes are transmitted without modifications from the zero frequency component (continuous component) up to a certain frequency f_c and the higher frequencies are attenuated. The range of frequencies that pass without attenuation is called band pass, abbreviated to band, which allows the passage of frequencies with an attenuation of power less than 50% is usually taken as band pass.

1.2.1.6 Maximum speed of a transmissive channel

Even a perfect channel has a limited transmission capacity, as it behaves like a low-pass filter (i.e. a filter that only allows the passage of frequencies below a characteristic frequency of the same filter, called cut-off frequency). Nyquist showed that if a signal is transmitted through a channel that behaves like a band B low-pass filter, the incoming signal, and therefore filtered, can be reconstructed in full using 2B samples of the signal per second. Taking a number of higher samples would not be useful because the higher frequencies have already been attenuated. If the signal is composed of discrete L levels, the Nyquist theorem shows that the maximum speed v_{max}, expressed in bits/s, is equal to:

$$v_{\max} = 2B \log_2 L \tag{1.15}$$

It is clear that if the signal is binary, there are only two levels and (1.15) is reduced to $v_{max} = 2B$.

From (1.15), it can be deduced that to increase the speed or capacity of a channel, both band B and the number of L levels of the signal can be increased. However, in practical cases, the band cannot be increased at will due to reasons of cost, practical impossibility or deliberate choice, and neither can the number of signal levels be increased at will because it would make the receiving system more complex and, moreover, would increase the errors upon receipt due to the lower resistance to distortion and noise phenomena typical of real channels.

What is stated above is valid for channels where there is no noise (ideal channel). When, in reality, noise N is always present on the channels of communication, in particular, thermal noise due to thermal agitation of the molecules of the materials making up the channel, this factor must be taken into account. It is measured in a relative manner by taking the ratio of the power of the signal S and the power of noise N, giving rise to those that is defined as signal-to-noise ratio (S/N), also called SNR. It is

mostly measured by resorting to base 10 logarithmic scales ($10 \log_{10} S/N$), expressing it in decibels (dB). This ratio is very useful because it helps to compress a very extended dynamic of the S/N ratio. In fact, an S/N ratio $= 10$ is equal to 10 dB, an S/N $= 100$ ratio is equal to 20 dB, an S/N $= 1{,}000$ ratio is equal to 30 dB and so on.

Shannon explored the discourse on the maximum speed v_{max}, expressed in bits/s, on a noisy channel, of bandwidth B, characterised by a certain SNR and found the following ratio:

$$v_{\max} = B \log_2 \left(1 + \frac{S}{N}\right) \tag{1.16}$$

The value expressed by (1.16) represents an upper limit that cannot be reached by real devices, even resorting to a greater number of levels of the signal or by increasing the sampling frequency.

You might consider increasing the signal level to increase the SNR, but this would involve the occurrence of effects, such as non-linearity, which would increase the rate of errors upon receipt and as such this recourse appears to be impractical. The only possible solution appears to be the increase of band that cannot take place indiscriminately due to the reasons discussed earlier.

1.2.1.7 Alterations of the signal in transmission channels

Transmission of the signals is always accompanied by possible alterations due to the following factors:

1. attenuation;
2. delay distortion;
3. noise.

These changes involve the possibility of mistakes in reception, imposing limits on the maximum distance over which a signal can pass and the maximum baud rate that can be reached on a channel characterised by a certain bandwidth.

With regard to attenuation, this is always present in transmission media and is proportional to the distance to be covered by the signal. With guided media (which will be illustrated in the following), in general, attenuation has a logarithmic trend with distance while with non-guided media (which will also be addressed later), it depends on several factors such as the distance, the moisture in the air, the presence of rain and dispersion. As the signal has to be received with an intensity such that it can be detected by the receiving circuits, you might consider increasing its level of transmission according to the specific needs. However, this is not always possible due to the occurrence of non-linearity effects in the circuits, as discussed earlier. To avoid this, in analogue transmission amplifiers are used on the channel that, however, also amplify noise and as such, it is not possible to enter here an indefinite number, while in digital transmission regenerators are inserted within the channel, which completely regenerate the signal, eliminating the noise component. Furthermore, attenuation also usually depends on the frequency and as such, the different spectral components that compose the signal are attenuated in a different manner, modifying unequally the profile of the transmitted signal. In this sense, in addition to amplifiers, use is also made of equalisers that compensate for the spectral distortion, restoring the original profile of the signal.

With regard to the delay distortion, the different spectral components travel at different speeds within the channel due to the physical limitations of the spectral components. In this case also, use is made of equalisation techniques to compensate for the behaviour of the channel.

With regard to noise, this represents a signal (in reception) that does not belong to the transmitted signal. It can be divided into the following:

1. thermal noise or white noise;
2. intermodulation noise;

3. crosstalk;
4. impulsive noise.

Thermal noise is generated from agitation of the electrons depending on the temperature. It is present in transmissive and receptive circuits and within the channel itself. It has an intensity that does not depend on the frequency and from which the name of white noise is derived. It cannot be eliminated but can possibly be reduced by using high-quality electronic components. In practice, in order to reduce the effects, the level of the signal should be increased to tolerable levels (avoiding non-linearity primers).

Intermodulation noise depends on the interference that occurs when multiple independent signals at different frequencies travel along the same transmission medium. In this case, the effects of non-linearity can generate multiple frequencies (sum or difference between the various frequencies) that can interfere with the signal that actually travels at one of these frequencies. This effect is manifested both as a result of malfunction or ageing of the electronic part and as a consequence of an excess of power of the signal that is being transmitted.

Crosstalk is a phenomenon of electrical coupling between inadequately insulated nearby transmission media. In this case, the signal that is being transmitted is inductively coupled with the adjacent conductors, overlapping with the signal passing through the latter. To counteract this effect, attempts are being made both to improve the insulation of the transmissive medium and to decouple the transmission media through appropriate adjacent windings of conductors that carry the signal.

Impulsive noise depends on the sporadic phenomena that can generate unwanted signals of short duration (from which the name impulsive is derived) in electronic circuits or in the transmission channel. They can be represented by ignitions of electromagnetic devices or electrical power surges or in the areas surrounding the electronic circuits or transmission channel. Unfortunately, it is not predictable in advance and is often very high in intensity. It has a limited effect in analogue transmissions but particularly serious in digital transmissions.

1.2.1.8 Signals and data

Analogue signals are signals that vary their characteristics with continuity while digital or numeric signals are signals that vary their characteristics according to discrete levels. Similarly, analogue data assumes continuous values in a certain range and can be represented by data collected by sensors for pressure, temperature, voltage or electrical current, for voice signals, for video signals, etc. On the contrary, digital data assume discrete values in a certain range. They may be represented by integers, from written text, etc. An analogue datum can be represented with an analogue signal that occupies the same spectrum. A digital datum may be represented with a digital signal that identifies numbers with different levels of pulse amplitude.

It is possible to represent digital data with analogue signals (as in the case of modems) or analogue data with digital signals (as in the case of codecs). The first case is, for example, represented by the communication between computers via a telephone line where numerical data are processed by the modem into analogue signals that are transmitted and reconstructed upon receipt as numeric data from a similar modem. The second case is represented by telephone communication via ISDN line, where voice is digitised by a codec, by sampling, transmitted as numeric data and regenerated upon receipt as an analogue signal. Transmission of signals is called analogue if the signal is transmitted without taking into consideration its meaning: in this case, transmission is limited to sending the signal and amplifying it if it is necessary. The transmission of signals is called digital where the content of the data is also taken into account if operations of contrast of the attenuation are performed: in this case, the signal is not simply amplified, but interpreted and regenerated. Digital transmission offers greater immunity to the alteration of data transmitted over long distances, greater homogenisation of the transmission for various types of data (since both the digital data and the analogue data are transmitted

using the same techniques) and, in general, greater security and confidentiality. Digital transmission is, however, characterised by generally higher costs, increased complexity of electronics used and the need to renew entirely existing communication infrastructures.

1.2.1.9 Encoding of numerical data

Numerical data are usually represented by numerical signals by using sequences of discrete voltage pulses that are characterised by a well-determined duration. Binary data are encoded in such a way as to create a match between bit value and signal levels. It is very important that the receiver knows the instant when the bit starts and ends in order to be able to read it at the right time in order to determine the value of the bit depending on the coding used. This feature is called synchronisation and it should, preferably, be performed by reading the value of the signal at the instant of time corresponding to a half bit to ensure the best results and to minimise the occurrence of errors. It is possible to make different choices of coding, characterised by different performances, which can improve the desired results.

It is very important to take into account the spectrum of the signal, for different reasons: higher frequency components require a higher bandwidth; the absence of a continuous component, in order to be able to use transmission techniques that are more resistant to noise; the concentration of the spectrum in the centre of the band considered to be able to facilitate any multiplexing (which will be described later).

It is also very important time synchronisation (as already mentioned), possible error detection (typical characteristic of upper network layers), the strength of the signal in relation to interference or noise; the complexity of the implementation. There are various coding techniques. In the following, only unipolar, polar and biphasic techniques are described.

There are essentially three unipolar encodings: return-to-zero (RZ), non-return-to-zero level (NRZ-L) and non-return-to-zero invert (NRZI) on ones. RZ encoding provides for the transmission of a signal of duration T for each bit. The signal is equal to zero if the bit is equal to 0 while it is equal to a voltage pulse of duration T/2 if the bit is equal to 1. NRZ-L coding is similar to the RZ, with the difference that the voltage level relating to bit 1 remains high for the entire duration of the period T. NRZI coding is similar to NRZ-L but is differential, that is it changes symbol when there is a bit equal to 1 and remains unchanged when the bit is equal to 0. The unipolar encodings NRZ and NRZI are easy to design and create, usually used for short lines and have a efficient use of bandwidth (if R is the transmission capacity in bit/s, power is concentrated between 0 and R/2).

In unipolar encodings, there is unfortunately a continuous component with all the associated problems that were illustrated above and the long sequences of bits of value 0 (or 1 in NRZ-L) generate a constant signal devoid of transitions that can cause loss of synchronism at the receiver.

To improve the performance of unipolar codings, use is made of polar encodings that can either be of a simple or differential type. In this case, voltage varies between a positive value and an equal negative value. This reduces the effect of the continuous component but leaves the problem of synchronisation.

A further improvement is obtained with bi-phase encodings, the most widely known of which is Manchester encoding. It can be of a simple or differential type. Simple coding uses two voltage levels. Bit 1 is represented by a voltage $-$ V for half T period and by a voltage $+$ V for the remaining half T period. Bit 0 is represented in the opposite manner, that is by a voltage $+$ V for half T period and by a voltage $-$V for the remaining half T period. Differential encoding is similar to simple encoding but represents bit 1 as a variation with respect to the encoding of the previous bit.

The advantages of Manchester encoding are represented by ease of synchronisation, since each bit has a transition in the half that is used by the receiver to synchronise with, there is total absence of the continuous component and ease of error detection in the absence of the determined transition. The disadvantages are represented by the necessity to use a double frequency compared to the bit rate. Manchester encoding is used in the standard IEEE 802.3 (Ethernet) and IEEE 802.5 (token ring), both

on coaxial cable and on twisted pair. A comparative spectral summary between the various encodings is shown in Figure 1.34.

1.2.1.10 Analogue modulation

Representation of the analogue data via analogue signals can be both via base band and through suitable modulation. In base band, the transmitted signal is substantially equal to the analogue data to be transmitted and is characterised by the same frequency band of the analogue datum. In modulation, it is transmitted by effectively modulating a sinusoidal carrier using analogue data as modulating signal. In this paragraph, an illustration of different modulations techniques is given.

When modulation is performed, it generates a resulting signal, which has a band of the same order of magnitude as the modulating signal, the difference being that this band is centred on the frequency of the carrier signal. By using, therefore, a high-frequency carrier, the band required to transmit the data in a desired range can be moved. This can be very useful when intending to send several signals simultaneously on the same medium by moving the bands relating to the various data in different ranges of the available band for transmission and implementing what is referred to as frequency division multiplexing.

Since it is possible to vary different characteristics of the carrier signal according to the modulating signal, modulation can be divided into the following:

1. amplitude modulation, when the amplitude of the carrier signal is varied according to the amplitude of the modulating signal;
2. frequency modulation, when the frequency of the carrier signal is varied according to the amplitude of the modulating signal;
3. phase modulation, when the phase of the carrier signal is varied according to the amplitude of the modulating signal (Figures 1.35 to 1.37).

In amplitude modulation, it has already been stated that the modulating signal $m(t)$ modulates the amplitude of a frequency cosinusoidal carrier f_p generating a signal $s(t)$ equal to:

$$s(t) = [1 + n_a \cdot m(t)] \cos(2\pi f_p t) \tag{1.17}$$

where n_a is the index of amplitude modulation that is chosen in such a way that $|1 + n_a \cdot m(t)| > 0$.

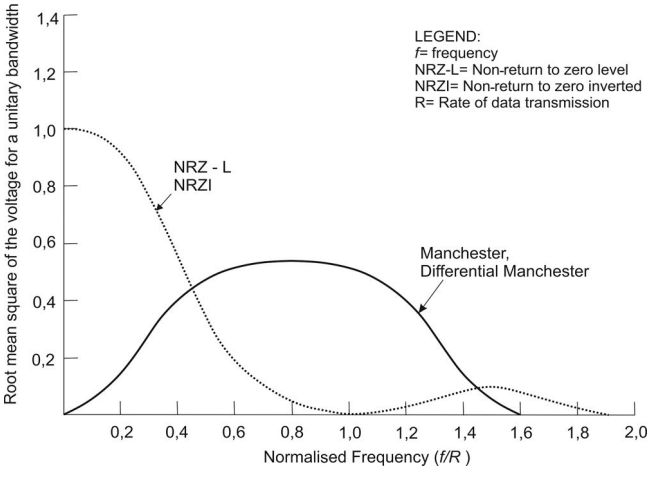


Figure 1.34 Spectral comparison between the various encodings.

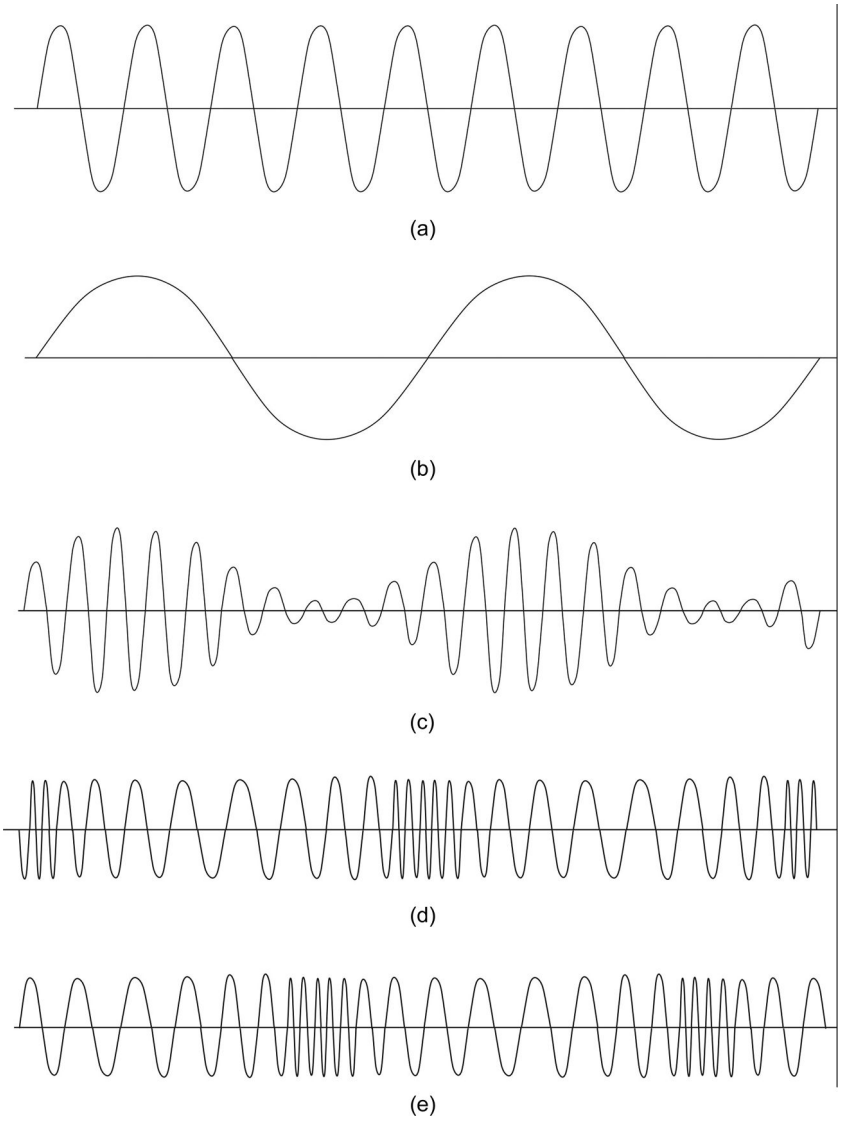


Figure 1.35 Various types of modulation: (a) carrier, (b) modulating signal, (c) amplitude modulation, (d) phase modulation and (e) frequency modulation

If the signal is also a cosinusoidal frequency f_m, that is of the type $\cos(2\pi f_m t)$, by inserting the relevant values in (1.17) and developing it has:

$$s(t) = \cos(2\pi f_p t) + \frac{n_a}{2}\cos(2\pi(f_p - f_m)t) + \frac{n_a}{2}\cos(2\pi(f_p + f_m)t) \qquad (1.18)$$

that is its spectrum is represented by a spectral row at the carrier frequency, plus two symmetrical rows with respect to the carrier located at a distance equal to the frequency of the modulating signal.

In general, an amplitude modulated signal is characterised by a spectrum of the modulating signal doubled and placed symmetrically around the carrier frequency, generating what are called lateral bands.

This means that occupation of the band of the modulated signal is twice that of the modulating signal. However, it is possible to eliminate the lower side band or even the carrier (if the modulating

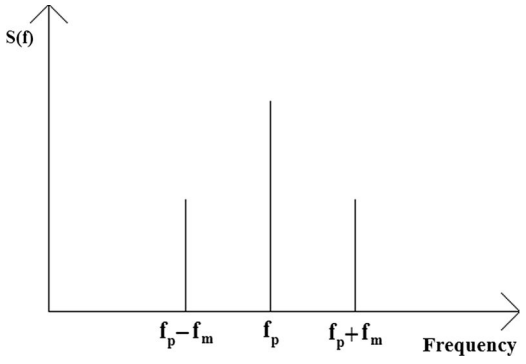


Figure 1.36 Amplitude modulation spectrum with cosinusoidal modulating signal.

signal has no continuous components or is close to zero), resorting to appropriate filters in order to reduce the total band occupied and the total power transmitted.

In the modulation of frequency, it has already been stated that the modulating signal $m(t)$ modulates the frequency of a cosinusoidal frequency carrier f_p generating a signal $s(t)$ equal to:

$$s(t) = A_p \cos\{[2\pi f_p + n_f \cdot m(t)]t\} \tag{1.19}$$

where n_f is the index of frequency modulation.

Given its intrinsic characteristics, the ΔF band occupied as an effect of the modulating signal amplitude is:

$$\Delta F = 2\pi f_{\max} - 2\pi f_{\min} = n_f \max[m(t) = n_f A_m] \tag{1.20}$$

A_m being the maximum amplitude of the modulating signal $m(t)$.

From (1.20), it can be seen how an increase in the amplitude of the modulating signal leads to an increase in the band occupied by the modulated signal, contrary to the amplitude modulation in which this band is constant, only varying the amplitude. Contrarily, the frequency modulation leaves unchanged the amplitude of the modulating signal. In general, the bandwidth B_f of a frequency modulated signal is infinite, but can be approximated, roughly, with the following ratio:

$$\mathrm{Bf} = 2\Delta\mathrm{F} + 2\mathrm{B} = 2n_{\mathrm{f}Am} + 2\mathrm{B} \tag{1.21}$$

In the modulation phase, it has already been stated that the modulating signal $m(t)$ modulates the phase of a cosinusoidal frequency carrier f_p generating a signal $s(t)$ equal to:

$$s(t) = A_p \cos[2\pi f_{pt} + n_p \cdot m(t)t] \tag{1.22}$$

where n_p is the index of phase modulation. In fact, a time delay proportional to the modulating signal is introduced.

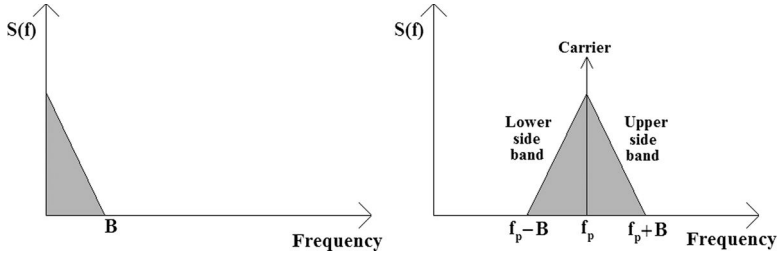


Figure 1.37 Amplitude modulation spectrum with bandwidth B modulating signal.

As in frequency modulation, also bandwidth B_p modulation phase of the modulated signal is infinite but can be approximated, roughly, with the following ratio:

$$B_p = 2(n_f\ A_m + 1)B \tag{1.23}$$

1.2.1.11 Modulation of numerical signals

Modulation techniques are also used to convert numerical data into analogue signals. This is done by modulating a sinusoidal carrier using numerical data. Upon receipt, the signal is demodulated in order to reconstruct the modulating numeric signal. It has already been stated that this type of function is performed by a device called modem (modulator – demodulator) that is used for data transmission over a switched network. This functionality is also used for digital transmission over optical fibre. The three fundamental techniques are amplitude shift keying modulation (ASK), frequency shift keying modulation (FSK) and phase shift keying modulation (PSK).

In ASK modulation, a sinusoidal carrier is modulated into amplitude by multiplying its amplitude by the numerical signal to be transmitted, as shown in Figure 1.38.

The transmitted signal is therefore characterised by the following formula:

$$s(t) = \left\{ \begin{array}{ll} A\cos(2\pi ft), & bit\ \ 1 \\ 0, & bit\ \ 0 \end{array} \right\} \tag{1.24}$$

In FSK modulation, the frequency of a sinusoidal carrier is modulated according to the numerical signal to be transmitted, as shown in Figure 1.39. If the values of the numeric signal are binary, only two frequencies will be used according to the value of the bit to be transmitted.

The transmitted signal is therefore characterised by the following formula:

$$s(t) = \left\{ \begin{array}{ll} A\cos(2\pi f_1 t), & bit\ \ 1 \\ A\cos(2\pi f_2 t), & bit\ \ 0 \end{array} \right\} \tag{1.25}$$

In PSK modulation, the phase of a sinusoidal carrier is modulated according to the numerical signal to be transmitted, as shown in Figure 1.40. If the values of the numeric signal are binary, a phase change for bit 1 and no phase change for bit 0 will, for example, be used.

The transmitted signal is therefore characterised by the following formula:

$$s(t) = \left\{ \begin{array}{ll} A\cos(2\pi ft + \varphi_1), & bit\ \ 1 \\ A\cos(2\pi ft + \varphi_2), & bit\ \ 0 \end{array} \right\} \tag{1.26}$$

Greater efficiency of the channel can be obtained by modulating in such a manner that each symbol transports more than one bit.

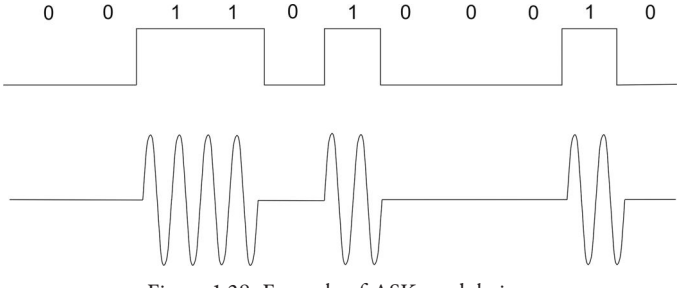


Figure 1.38 Example of ASK modulation.

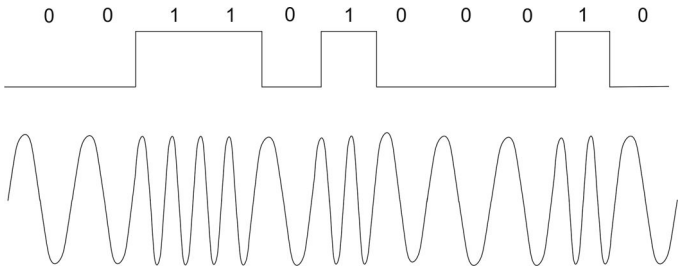


Figure 1.39 Example of FSK modulation.

For example, in PSK quadrature (QPSK), it is possible to use four phase values to transmit two bits at a time, as, for example:

1. 00, phase 0°
2. 01, phase 90°;
3. 11, phase 180°;
4. 10, phase 270°.

These phase values can of course be varied, provided they remain in quadrature between each other.

Continuing according to this diagram, it is possible to create more complex modulations by using a larger number of phase angles, taking into account that increasing of the same, decreases the relative interdistance, making the system more vulnerable to errors.

1.2.1.12 Quadrature amplitude modulation

In quadrature amplitude modulation (QAM) (AM quadrature), the carrier signal is divided into two carriers shifted by 90°, hence the term quadrature is derived. Once the carriers are shifted, they are modulated independently, possibly at several levels and are, therefore, recombined.(Figures 1.41 and 1.42)

To increase the amount of information conveyed, combined modulations can be used in phase and amplitude on the two components and as such may have the following layouts: 4QAM, 16QAM, 64QAM and so on. These techniques are used for digital transmission over analogue signal, such as in modems, digital radio links and in optical kilometre.

1.2.1.13 Sampling and digitising

In order to transmit an analogue datum by means of a digital transmission, the analogue datum must first be transformed into a numerical signal. Conversion takes place in two phases: sampling of the analogue signal and digitisation of the sampled signal.

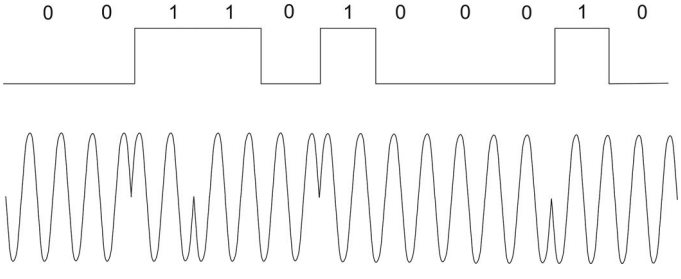


Figure 1.40 Example of PSK modulation.

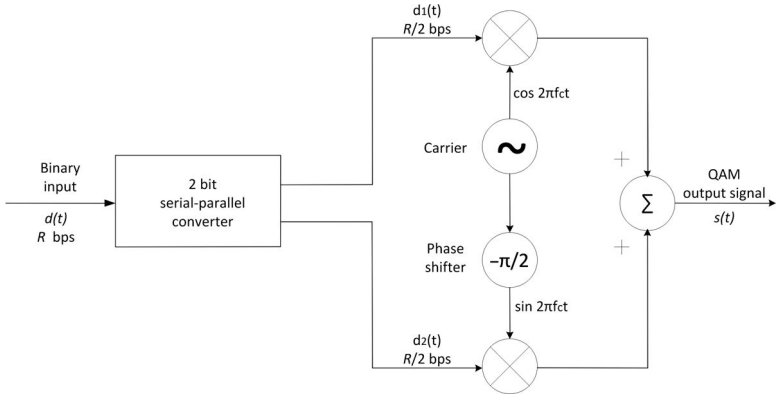


Figure 1.41 Layout of QAM modulator.

In fact, sampling consists of taking the values of the analogue signal with a certain predetermined frequency. In practice, the input signal is used to modulate in amplitude a sequence of set frequency pulses, in which the amplitude of the pulses coincides with the amplitude of the input signal at the sampling instants. The main problem involves determining the right frequency with which to sample in order to be able to reconstruct the original signal from its samples.

The sampling theorem or Nyquist-Shannon theorem states that, given a signal $s(t)$ whose spectrum is characterised by a limited bandwidth B, it is possible to completely reconstruct the same signal, using its samples, if the sampling frequency f_s is greater than or equal to 2B. This can be demonstrated mathematically but this demonstration is not reported for reasons of space.

In practice, the spectrum of the sampled signal consists of replicas of the spectrum of the original signal shifted by multiples of the frequency of the pulse signal used to sample it. If the spectra of two adjacent replicas of the original signal do not overlap, a low-pass filter upon receipt can be used to isolate a single replica of the signal, obtaining a signal whose spectrum is equal to the spectrum of the original signal. The non-overlapping condition can be expressed as: $B \leq f_s - B$ that is $f_s \geq 2B$. The situation is shown schematically in Figure 1.43.

In practical applications, the sampling frequency must be at least higher than 2B to provide a necessary range in order to prevent any possible effects of non-ideality of filters mitigating necessary parts of the signal. It is clear that the sampling theorem is closely related to the law of Nyquist on maximum transmission capacity of a channel without noise.

The analogue signal obtained with the sampling is then digitised using different techniques. The purpose of this operation is to be able to convert the analogue signal to a numerical signal and to be able to use the techniques of numerical transmission and the related benefits (noise immunity thanks to signal regeneration during transmission; ability to use time division multiplexing, as will be shown in the following; homogenisation of the transmission of signals; etc.).

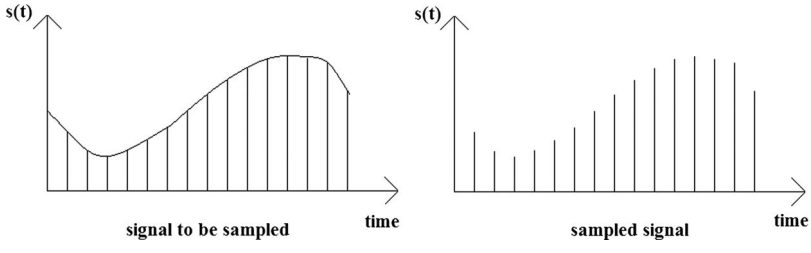


Figure 1.42 Example of signal to be sampled and sampled signal.

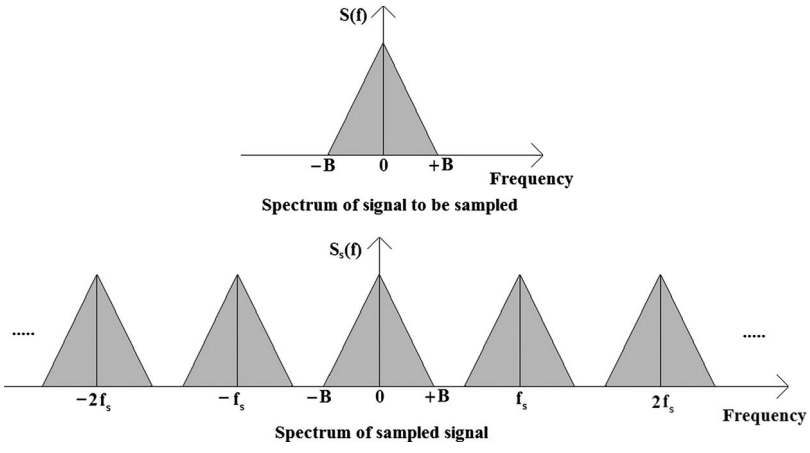


Figure 1.43 Example of signal spectrum to be sampled and sampled signal spectrum.

During digitisation, the levels of amplitude of the sampled signal are normalised according to well-defined levels: the greater the number of levels, the better is the approximation of the signal through the digital signal. As binary logic is employed, the number of levels is a power of 2 and as such with 2-bit representation there will be 4 levels (2^2), with 4-bit representation there will be 16 levels (2^4), with 8-bit representation there will be 256 levels (2^8) and so on.

1.2.1.14 Multiplexing

Multiplexing is a technique that is used to carry different information streams over one and the same transmission medium. It is used when there is a need to transmit various communications, each characterised by an occupation of limited band, and if there is a transmission channel equipped with a wider band. The part of the band occupied by each communication is called channel.

There are three main multiplexing techniques:

1. frequency domain frequency division multiplexing (FDM);
2. wavelength domain wavelength division multiplexing (WDM);
3. time domain time division multiplexing (TDM).

In frequency division multiplexing (FDM), it is assumed that there is a series of signals, each characterised by a limited band B and by a transmission medium with band limited by the frequencies f_1 and f_2, being $f_2 - f_1 >> B$. In this case, using the techniques of amplitude modulation and eliminating the carrier and the lower side band, as previously seen, it is possible to modulate sinusoidal signals at the frequencies $f_1 + B, f_1 + 2B, f_1 + 3B$, etc., until reaching and occupying the entire band of the transmission channel $f_2 - f_1$. Signals, thus, modulated occupy discrete intervals within the transmission band of the medium and can be transmitted simultaneously without interfering with each other. Upon receipt, the various signals are separated by operations of filtering demodulation (Figures 1.44 to 1.56).

In practice, the different signals to be transmitted modulate different frequency carriers, called subcarriers. The modulated signals are then combined, generating an overall signal in base band. Subcarrier frequencies are selected in such a manner as to minimise the overlap of the combined signals. The overall signal, which is an analogue type, is generally used to modulate a carrier to translate the same signal over a frequency that can be supported by the transmission medium. When the signal reaches the receiver, it is demodulated back into base band. Demodulators and filters suitable for the various subcarriers are then used that separate and return the original signals.

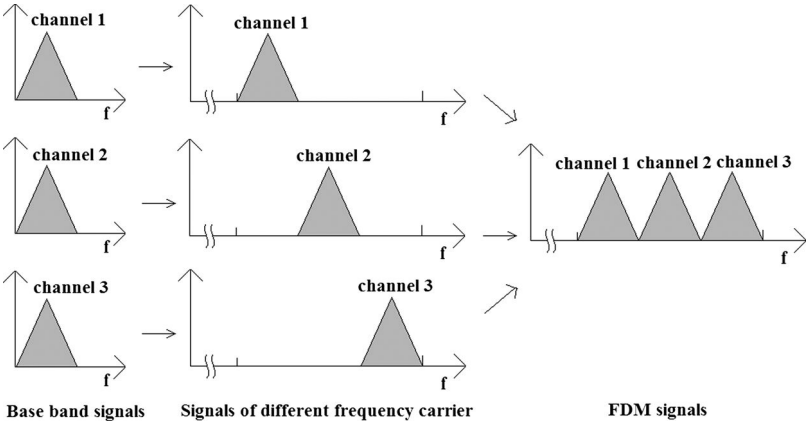


Figure 1.44 Example of frequency division multiplexing.

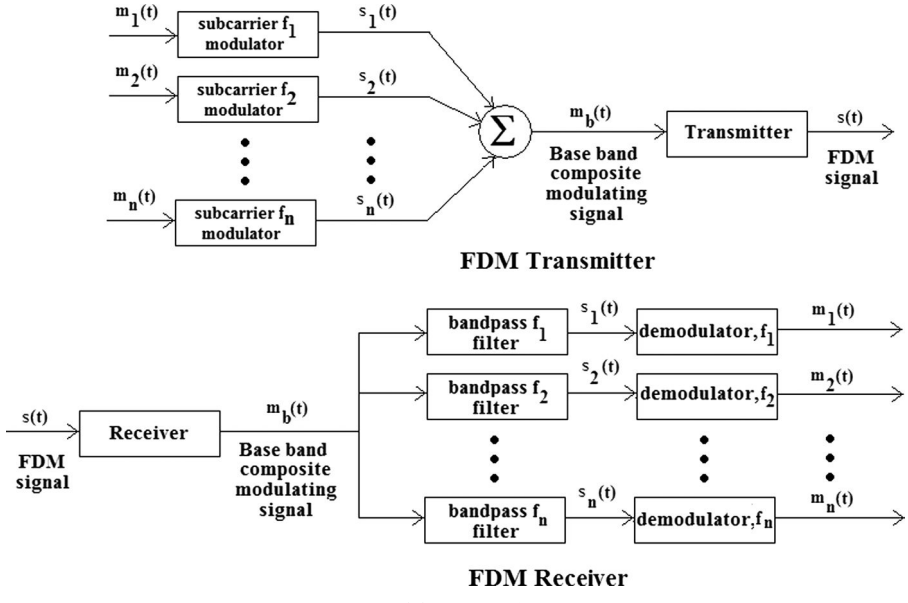


Figure 1.45 Layout of frequency division multiplexing.

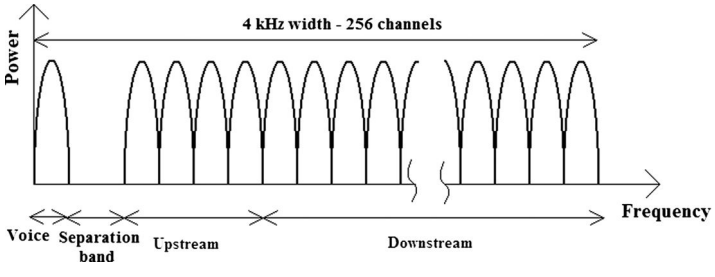


Figure 1.46 Usage diagram of the ADSL band.

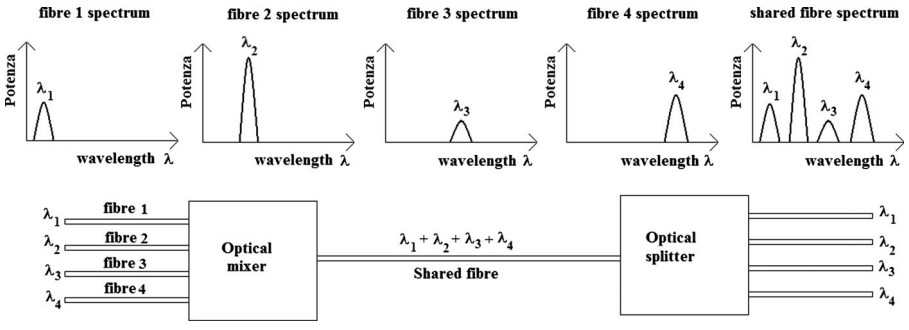


Figure 1.47 Wavelength division multiplexing layout.

If the upper sole side bands of the original signals are assumed to be used, the bandwidth of the overall signal is roughly equal to the sum of the higher side bands of the individual signals. In practice, the band occupied by the overall signal is greater than the sum of the individual bands in order to obtain a separation between the different channels, reducing the mutual interference due to possible non-ideality of the filters.

An example of FDM is represented by the ADSL system that is used to provide higher bandwidth digital access, on telephone lines, which is not possible with a modem. Telephone lines consist of a

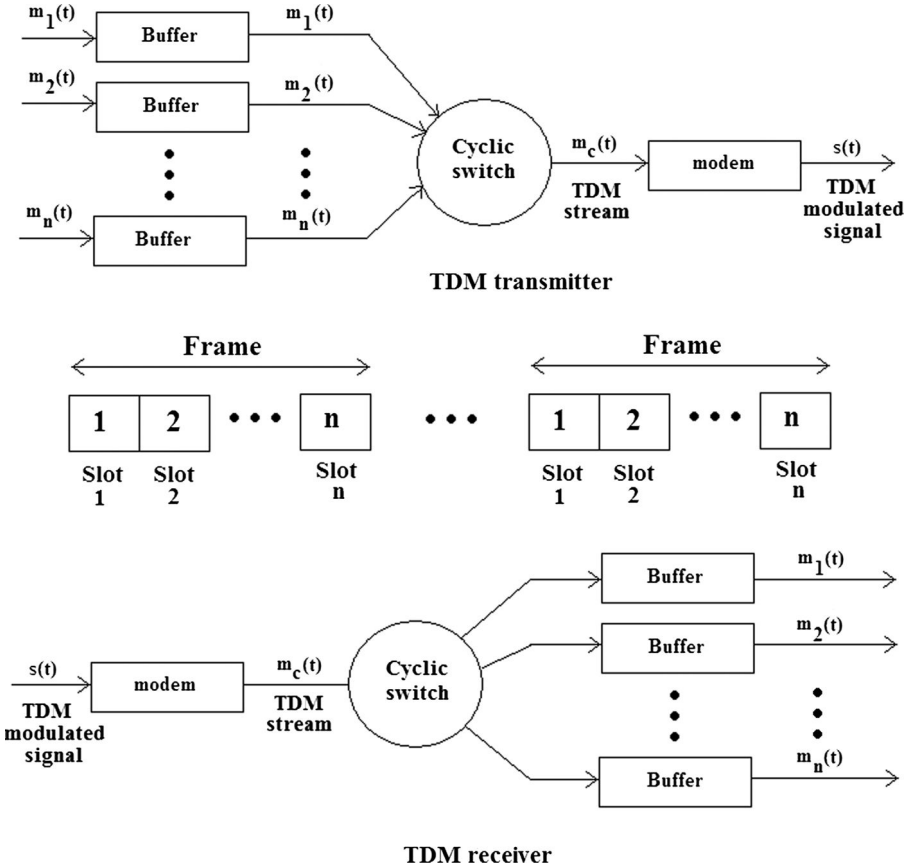


Figure 1.48 Layout of time division multiplexing.

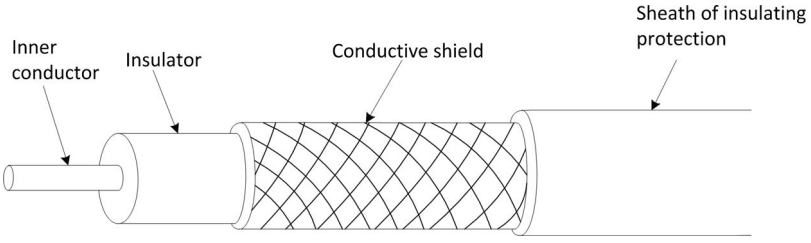


Figure 1.49 Composition of a coaxial cable.

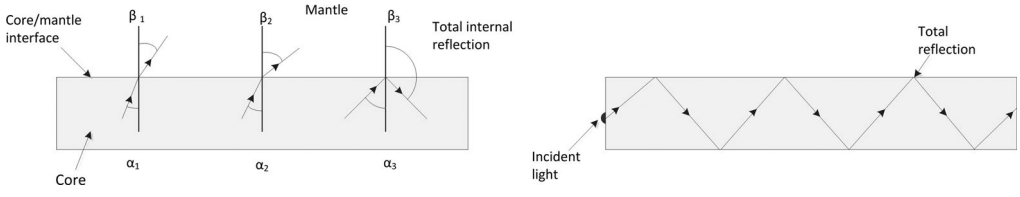


Figure 1.50 Total refraction and reflection.

copper twisted pair that is used for voice transmission. Given the limited bandwidth of voice (<4 kHz), a low-pass 4 kHz filter is applied on the twisted pair. The twisted pair is, however, able to guarantee a bandwidth of the order of megahertz, depending on its length between the telephone terminal and the nearest switching centre that can vary from a few hundred metres to several kilometres. The spectrum available on the twisted pair is, therefore, divided into 256 channels whose band is 4 kHz and is able to operate at speeds of the order of 60 Kbps each. Channel 0 is reserved for voice. The subsequent four channels are not used to avoid interference between telephonic transmission on the channel and data transmission on the upper channels. The upper channels are

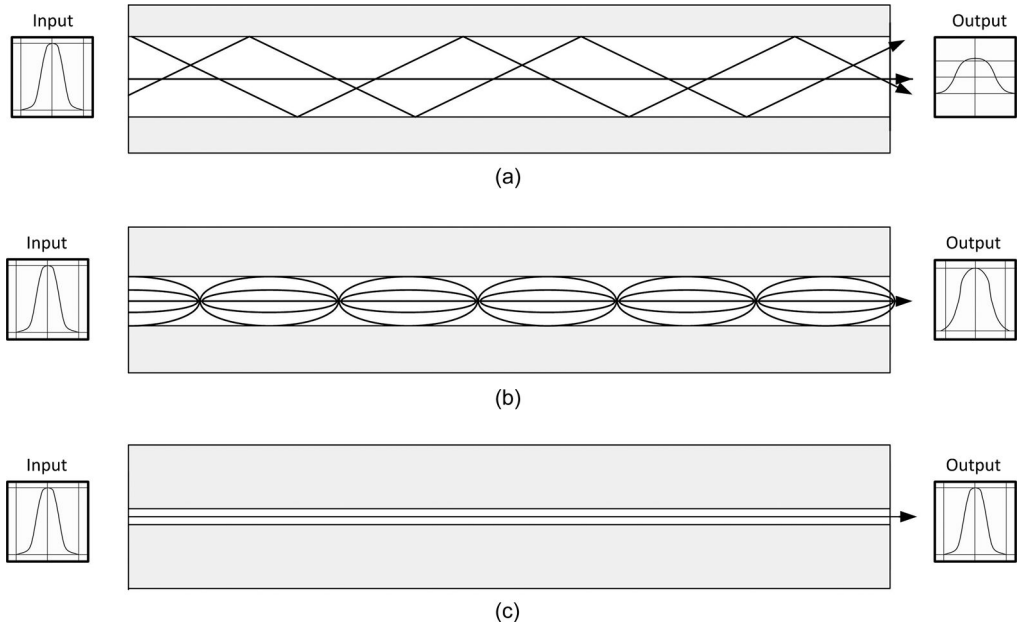


Figure 1.51 Light propagation for various types of fibre: (a) Multimodal, (b) with gradual profile index, and (c) monomodal.

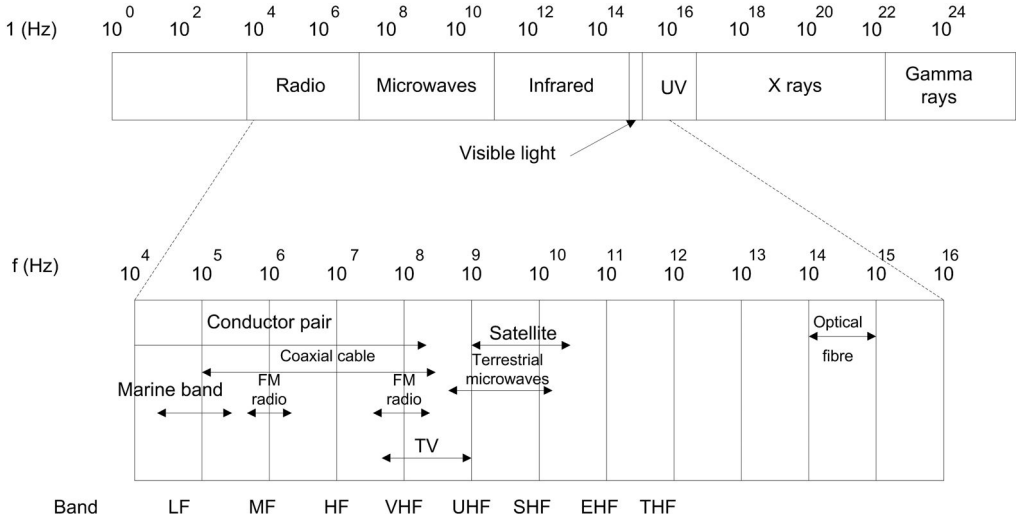


Figure 1.52 The electromagnetic spectrum.

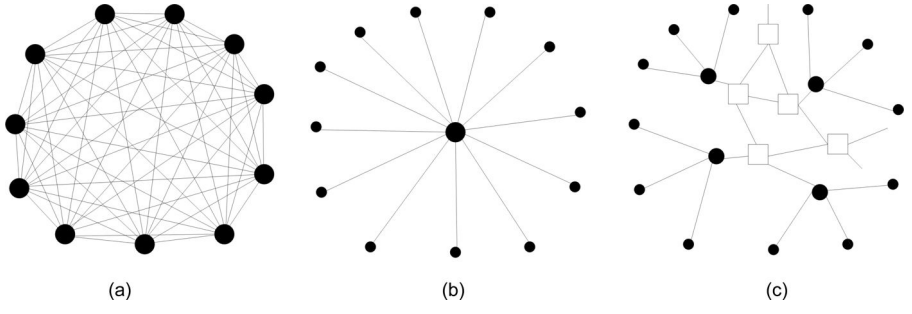


Figure 1.53 (a) Examples of fully interconnected telephone network, (b) centralised switching centre and (c) 2 level hierarchy.

used for data traffic, some for the outgoing traffic (upstream) and others for the incoming traffic (downstream). When the ADSL modem receives data, it separates it into parallel streams to be transmitted on different channels available; it generates an analogue signal in base band for each stream using 15 bit/baud rate 4,000 baud/s QAM modulation, and transmits it over the different channels using frequency modulation. Although, in theory, the bandwidth available would allow traffic of over 10 Mbps, not all channels are capable of transmitting at full bandwidth and usually the service provider decides its use. Usually, a greater number of channels is used for incoming traffic (downstream) while a

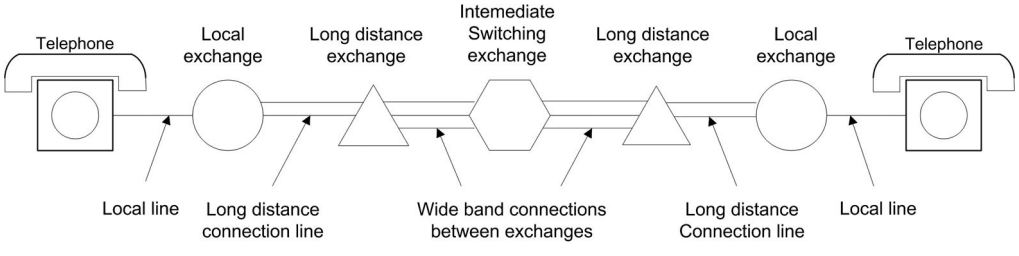


Figure 1.54 Connection diagram of a medium distance call.

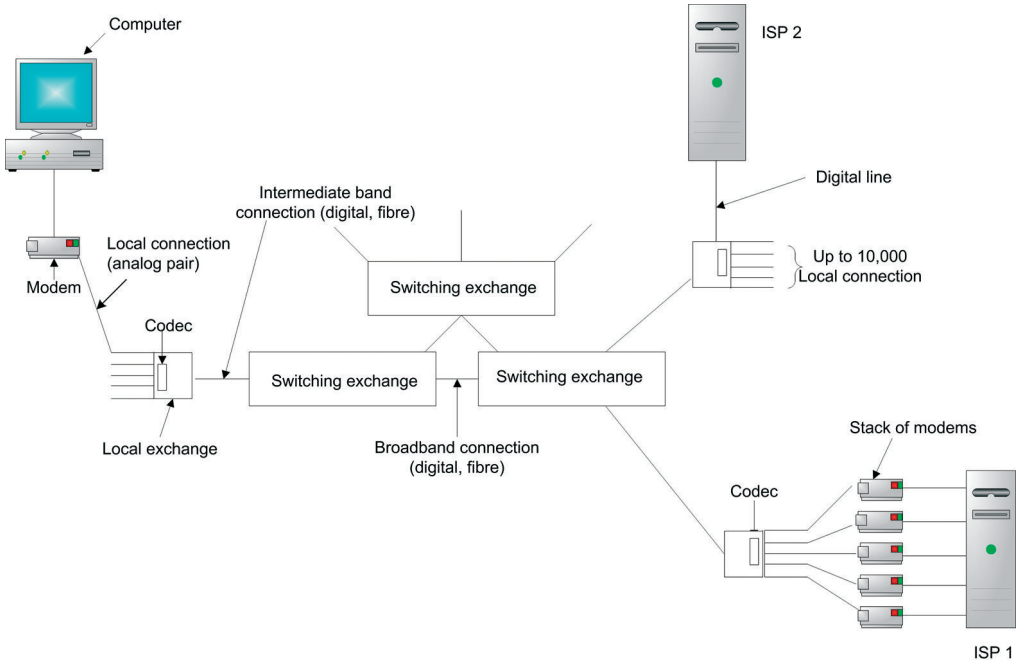


Figure 1.55 Analogue and digital connection diagram on a computer using analogue telephone line.

lower number is used for outgoing traffic (upstream) from which the term asymmetric service is derived.

In wavelength division multiplexing (WDM), the various signals modulate different wavelengths that are transmitted simultaneously, usually in optical fibre (the operation of which will be shown below).

Optical fibre is able to transmit electromagnetic signals with typical wavelengths of around 850, 1,300 and 1,550 nm. Each of these frequencies is able to transmit with a band of about 100 nm. Considering, for example, the wavelength of 800 nm, the lower wavelength band is $\lambda_1 = 800$ nm, which corresponds to a frequency $f_1 =$ (speed of light)$/\lambda_1 \approx 2.50 \times 10^{14}$ Hz while the upper wavelength of the band is $\lambda_2 = 900$ nm, which corresponds to a frequency $f_{2=}$ (speed of light)$/\lambda_2 \approx 2.22 \times 10^{14}$ Hz. The bandwidth available is $B = f_2 - f \approx 0.28 \times 10^{14}$ Hz $= 28{,}000$ GHz that represents a significant value in the telecommunications sector.

In the case of fibre, the different wavelengths relating to the various signals are sent to an optical combiner that mixes them together. Upon receipt, a reverse system, called splitter, separates amongst

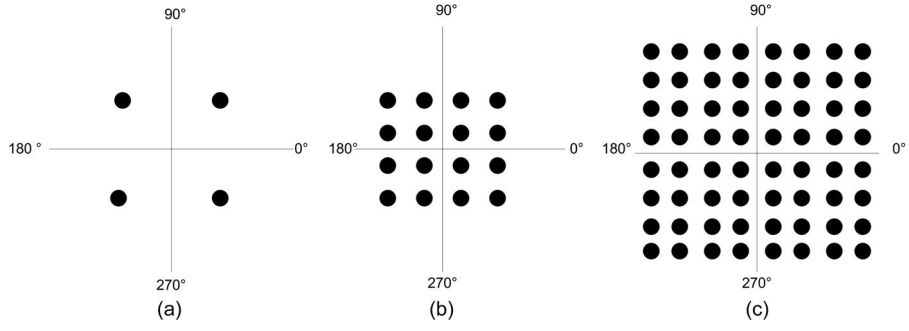


Figure 1.56 (a) QPSK constellation diagram, (b) QAM-16 and (c) QAM-64.

them the different components. In practice, the system is very similar to the FDM system, with the difference that wavelengths instead of frequencies are used, thereby taking advantage of passive devices, such as diffraction gratings, which do not introduce any additional noise. Using this technology, it is currently possible to transmit dozens of 10 Gbps channels through a single optical fibre and transmission capacity can be increased without having to increase the generation frequency of optical pulses, the latter being a factor that represents a major limitation for the transmission of data on optical fibre.

Time division multiplexing (TDM) uses a high-speed digital channel to simultaneously transmit a set of digital communications characterised by a lower speed.

Unlike FDM and WDM systems that bring together the various frequencies or wavelengths, in TDM, the bits of the various communications alternate systematically. In practice, the availability of the channel in time slots is divided and the different time periods are in turn dedicated to the different streams to be transmitted. Each time slot is named slot and is able to contain one or several bits of information relating to each independent stream. The overall flow is subsequently organised into frames: a frame represents a collection of time slots that contains at least 1 bit for each transmission. The flow relating to a single transmission is called a channel, as in the previous cases.

Not all the input data can be digital: in this case, the analogue input signals can be sampled and digitised as shown above. In addition, not all the input signals need to be characterised by the same transmission rate, as it is possible to modulate the duration of each time slot according to the transmissive requirement of each channel.

There is also the so-called synchronous TDM that in input has a certain number of transmission flows to which one channel is statically assigned: in practice, every transmission is assigned one time slot. When there is no incoming data on a particular channel, transmission continues unaltered and the relevant slot does not carry data.

As the frames are being continuously transmitted, the receiver must be able to identify the beginning of each frame to be able to hook in and maintain synchronism. To facilitate this operation, the frame contains the control bits that hold sequences of values discretely present in the data field in order to facilitate the operation of recognition and synchronisation. During the initial moments, the receiver tries to identify the synchronisation bits: once they are found in a number of consecutive frames, it hooks into them and starts decoding, always maintaining control over the synchronisation bits of received frames. If during transmission synchronisation is lost, the receiver again repeats the synchronisation operation just illustrated.

1.2.2 Transmission over guided media

It has already been said that the physical layer is responsible for transporting bits from one computer to another. To do this, different physical media can be used, each characterised by a different band pass, delay, cost and mode of installation and maintenance. The transmission media may be divided into guided media (copper cables, fibre optics, etc.) and non-guided media (waves radio, laser, infrared, etc.). These means are explained in the following sections.

1.2.2.1 Twisted pair

A copper twisted pair represents one of the older but still used transmission media. It is composed of two copper conductors, with thickness of around 1 mm and wound around one another in a helical structure. The two cables are interwoven in such a manner that the electromagnetic fields produced eliminate each other, reducing the overall irradiation and also the uptake of external disturbances. It is very widespread in phone systems, being used for the connection, sometimes of a few kilometres, from the end-user to the switching centre. It is one of the easiest and cheapest transmission media to handle. The twisted pair is used both for transmitting analogue as well as digital signals. The bandwidth

depends on the diameter of the cable and the total length and, in most cases, for sections of a few kilometres, a few megabits per second can be reached. Since attenuation is far from negligible, use may be made of amplifiers each 4/5 km for analogue signals and 2/3 km for digital signals. When using cables composed of various loops, in order to reduce crosstalk, pairs with different winding pitches are used.

There are different types of twisted pairs but two main types are used in the transmission of data that are the unshielded twisted pair (UTP) and shielded twisted pair (STP). Standards provided for cables generally comprising four pairs of wires, individually insulated and wrapped in pairs. There are various categories of cables, classified on the basis of the types of windings. These categories are defined on the basis of bandwidth capacity within defined distances (typically 100 m). There is therefore category 3 UTP that works up to 16 MHz, category 4 UTP that works up to 20 MHz, category 5 UTP that works up to 100 MHz and faster standards such as categories 5e, 6 and 7. Category 3 UTP twisted pair are also called target quality and are used both for telephony and data transmission up to 10 Mbps. Category 5 and 5e twisted pair UTP are used on local networks for speeds of up to 1 GBps. STP twisted pairs are, as already said, shielded and obtained by winding the pairs with conductive shield in order to reduce the effect of external disturbances. Shielding allows this category of twisted pair to work at a higher frequency (up to 300 MHz) at equal attenuation. STP twisted pair, although having better transmission characteristics, is also characterised by a greater rigidity (and therefore less ease of use) and by a higher cost and as such is only used in environments where there are strong electromagnetic disturbances.

1.2.2.2 Coaxial cable

The coaxial cable is composed of a copper core covered by an insulating coating, covered in turn by a cylindrical conductor usually made from a metal shield, which is in turn covered by a protective plastic sheath.

Being shielded, at equal attenuation, it can reach greater distances compared to the twisted pair with a larger bandwidth. Furthermore, screening ensures a greater resistance to external disturbances. It is capable of operating at frequencies of up to 1 GHz. There are two types of coaxial cable, classified according to their impedance mostly due to historical reasons than to technical reasons. The first, 50 Ω, that is used for digital transmissions and the second, 75 Ω, that is used for analogue transmissions.

1.2.2.3 Optical fibre

The optical fibre represents a fundamental component of an optical transmission system. This system is composed of a light source, a medium (the optical fibre, specifically) and a detector. The presence of an optical pulse is used to indicate, by convention, a logic 1, while the absence of a pulse is used to indicate a logic 0. By connecting to one end of the fibre light source, if the connection is performed by following certain well-defined criteria, the light pulses travel trapped within the fibre reaching the other end where they strike a detector that converts them back into electrical impulses.

The light remains trapped inside the fibre due to a law of physics called refraction/reflection. If the light travels from a medium (silicon of the fibre) to another (air around the fibre), the angle of incidence, which depends on the refractive indices of the two media, is different from the angle with which it emerges (refraction angle). As the angle of incidence increases so does the angle of refraction. If the angle of incidence exceeds a certain limit, and if the refractive indices are suitable, there may be total reflection, that is to say, light no longer emerges in the second medium but remains trapped in the first, since it can propagate for long distances, if the attenuation is relatively low, by subsequent reflections on the walls of the first medium.

If a light source is applied at one end of the fibre, there will be various light rays that affect with different angles the silicon/air interface and will be reflected at different angles. If the cross-section of

the fibre is appropriately reduced, there will be a single ray able to propagate: in this case, reference is made to single-mode fibre, opposite to the first fibre that is called multimode fibre. The single-mode optical fibre can work at faster speeds than multimode optical fibre, being able to reach 50 Gbps for approximately 100 km without amplification, but it is characterised by a higher cost.

Optical fibre is essentially composed of silicon, that is glass, appropriately treated in order to reduce its absorption to a maximum. For optical communications, three wavelength bands are used, centred at 0.8, 1.30 and 1.55 μ, where there are windows of reduced absorption and bandwidths of 25,000 and −30,000 GHz. The last two bands are characterised by a lower absorption than the first band that is in any case necessary because at that wavelength the same material can be used to build both the electronic part and the part relating to the light source.

As a pulse, even though laser, is composed of various wavelengths, the latter travel within the medium with slightly different velocities due to a phenomenon called chromatic dispersion, causing a temporal broadening of the pulse and requiring an increase of the temporal distance of the subsequent pulses (lower transmission speed) to avoid overlaps, and therefore disturbances and errors in transmitted signals.

To generate the light signal, two types of light sources are used, light emitting diodes (LEDs) and laser diodes, whose characteristics are listed in Table 1.4.

To convert the electronic pulse received into an electric pulse, suitable photodiodes are used, whose response times are in the order of nanoseconds, which allows speeds of the order of gigabit per second to be reached.

1.2.3 Wireless transmission

Wireless transmission allows communication without the use of permanent cables. In future, it is expected that communications will take place exclusively in this manner or via optical fibre.

There follows a summary of how this technology operates, with referencing to subsequent chapter 6 regarding security aspects.

1.2.3.1 The electromagnetic spectrum

When electrons are made to oscillate, electromagnetic waves are generated that propagate into space. The number of oscillations per second of a wave is called frequency and is measured in hertz, while the distance between two consecutive maxima or two minima in space is called the wavelength and is indicated by the letter λ (lambda). If connection is made to an oscillating electronic circuit of an antenna of appropriate size to the wavelength of the same electric oscillation, irradiation of the relative electromagnetic waves is promoted and the same can be picked up at a distance by a tuned receiver connected to a suitable antenna: wireless communications is based on this principle.

Table 1.4 Comparison between the characteristics of a LED diode and a laser diode.

Feature	LED diode	Laser diode
Transmission velocity	Low	High
Kind of fibre	Multimodal	Multimodal or monomodal
Distance	Short	Long
Duration	Long	Short
Temperature sensibility	Scarce	High
Cost	Low	High

In a vacuum, all electromagnetic waves, regardless of their frequency, travel at the same speed $c \approx 3 \times 10^8$ m/s. In nature, there is no greater speed than the speed of light. We have seen that between speed c, frequency f and wavelength λ there is a fundamental relationship:

$$c = \lambda f \tag{1.27}$$

Since c is known, it is enough to know the wavelength to calculate the frequency and vice versa.

The entire electromagnetic spectrum is usually divided into bands, according to the frequency or wavelength.

In wireless communications, radio waves, microwaves, infrared and visible light are used, starting from very low frequencies, by modulating them suitably in amplitude, frequency and phase. Ultraviolet rays, X-rays and gamma rays could also be used but this is avoided both because of the difficulty of generating the same and due to their risk to human health, given their high specific energy.

The amount of information that electromagnetic waves can carry depends, of course, on their band. The technology currently in use allows the carrying of a few bits for every hertz at low frequencies up to 8 bits per hertz at high frequencies. In most cases, a narrow frequency band is used even if, ultimately, techniques are used, called spread spectrum techniques that have particular advantages. These techniques are known as variable frequency spread spectrum and direct sequence spread spectrum, widely used in the military and also recently used in the civil sector.

In variable frequency spread spectrum, the transmitter jumps frequency hundreds of times per second and is very difficult to intercept and to disrupt. It also offers a high resistance to disturbances by multi-path because when the reflected secondary paths arrive in delay, the receiver has already changed receiving frequency, making it immune to this type of phenomenon. This technique is, for example, the one used by the Bluetooth communication system. In direct spread sequence spectrum, a higher frequency signal is used to expand the spectrum of a signal to be transmitted using appropriate spread codes. This technique is used in latest generation cellular systems and in some types of WLAN. These techniques will be discussed below in more detail, given their importance in today's communication.

1.2.3.2 Radio transmissions

Radio waves are relatively easy to generate and to manage. They can travel for relatively long distances and also passing through obstacles: for this reason, they are used intensively in communications. Given their relatively low frequency, they tend to propagate in all directions and for this reason, the transmitter and the receiver must not necessarily be aligned. Radio waves vary their characteristics according to the frequency and in any case their intensity decreases with distancing from source. At low frequencies, radio waves tend to propagate in all directions and to pass easily through obstacles while with an increase of the frequency they tend to become directional, bouncing off obstacles and being absorbed by rain. Radio waves are subject to disturbance and interference from unintentional sources such as electric motors and for this reason transmitters are subject to frequency limitations and concessions by authorities.

At low frequencies (very low frequency (VLF), low frequency (LF) and medium frequency (MF)), radio waves tend to follow the ground, being propagated for hundreds of kilometres.

At higher frequencies (high frequency (HF) and very high frequency (VHF)), radio waves tend to be absorbed by the ground but are reflected by the higher layers in the atmosphere (typically the ionosphere which is a layer composed of charged particles that is located between 100 and 500 km in height) and return to earth: being able to follow this path many times, they may cover considerable distances.

1.2.3.3 Microwave transmissions

Once a certain frequency, typically around 100 MHz, has been surpassed, radio waves begin to travel in a straight line, and can therefore be focused in specific directions, resorting to suitable antennas that

allow a high SNR to be achieved but that must, of course, be well aligned. Microwaves extend in such area.

Since these waves travel in a straight line, they are not able to reach destinations that are too far away and located beneath the horizon: in this case, suitable repeaters are used or the height of antennas is increased. The distance between repeaters depends on the square root of the height of the antenna. Microwaves, unlike lower frequency radio waves, are not able to pass through obstacles and at frequencies higher than 4 GHz, where the wavelength of the same is a few centimetres, they are absorbed by water and thus by rain, representing a significant problem when trying to communicate at those or at higher frequencies.

Microwaves are used for telephone transmission over long distances, in cellular telephony, for television, in satellite communications and in radar.

With microwaves, the so-called Industrial, Scientific and Medical (ISM) bands are also employed that are appropriately confidential international bands for industrial, scientific and medical applications. These bands, in order to avoid interference between the various transmitters, involve the mandatory use of expanded spectrum emission techniques. The Bluetooth system and 802.11 LAN, for example, operate on these bands.

1.2.3.4 Infrared and millimetre wave transmissions

At frequencies higher than microwaves, the infrared band is encountered, which, depending on the closeness to the visible band, are called near, medium or remote infrared. They are commonly used in remote controls for various electronic devices for household applications. The main problem of infrared is that it cannot pass through obstacles, at best reflecting against the same if they do not exhibit absorbent characteristics. This can however also be an advantage, since emissions remain confined within an environment as they are unable to reach neighbouring environments and reducing interference between devices.

Since infrared propagate strictly in a straight line, they are difficult to intercept unless the same beam is interrupted (and in this case transmission would be interrupted, revealing the presence of an intruder in communication) and do not require any government licence. Infrared is used for local communication between computers and their peripherals.

1.2.3.5 Light wave transmissions

Transmission via light waves has very ancient roots and was widely used in the past. Currently, it still finds applications thanks to the advent of laser (light amplification by stimulated emission of radiation) that, due to its ability to emit monochromatic radiation in highly concentrated and collimated beams, can reach discrete distances, with line of sight and high precision. This concentration of the beam also makes perfect alignment between the transmitter and receiver difficult when at distances exceeding 100 m.

Laser transmission is used to create a connection between buildings that are relatively close to each other and with line of sight, this communication however being disrupted by fog and rain, as well as by the turbulent convective motion of air.

1.2.4 Satellite transmission

This type of transmission takes place between the earth and suitable transceivers located on satellites with variable period orbit depending on their height. The range of coverage on earth can be very large (continental) or limited to a few hundred kilometres. As most satellites are powered by photovoltaic panels, the electrical powers available are relatively small meaning that the related powers of emission, in the region of microwaves, are modest. On the ground, in order to increase the level with which

satellite signals are received, use is made of parables directed with precision towards the satellite, which concentrate the signals received. The orbital period varies with height, in accordance with the laws of gravitation: the lower the latter, the shorter is the period of rotation. This ranges from 90 min for satellites placed in low orbits to 24 h for geostationary satellites located approximately 35,000 km at height and up to a month for satellites placed at 384,000 miles (earth – moon distance).

As there are bands around the earth composed of charged particles within which satellites are unable to orbit, which would otherwise be damaging to them, they are mainly located in three orbital areas (low, medium and high), depending on the type of service that they have to offer. It is evident that the greater the height, the smaller is the number of satellites required to cover certain terrestrial areas.

1.2.5 Fixed telephone network

The public switch telephone network (PSTN) was created to allow connection and communication between telephone apparatuses. It was only later, given the impossibility of laying high distance lines, that it was also used for computer remote connection and then for data transmission. This different purpose explains why with a remote voice connection, used as a data connection, a maximum 56 kbps can be achieved instead of a direct cable connection that easily allows speeds of the order of gigabit per second to be reached.

The telephone originally allowed the connection of pairs of apparatuses. Given the proliferation of cables within cities, switching centres were introduced where an operator, once called, sought to effect commutation and connection with the desired users. Thereafter, with the development of electronics, automatic switching centres were developed that could be controlled directly by users through dialling, on their apparatus, of the desired telephone number. These switching centres were subsequently organised according to hierarchies, depending on the switching level at which they operated. If two users were connected to a switching centre of the same level, the switching occurred locally. Alternatively, the connections were activated through higher level switching centres. The type and the typology of switching centre varies from country to country.

Local connections are currently made via category 3 twisted pair, while switching centres use coaxial, fibre optic and microwave cables. Initially, voice communication was performed analogically, but, with the advent of digital, analogue mode is limited, at most, to the connection between telephone and local switching centre.

To send digital signals over an analogue line (this obviously excludes the use of ADSL technology that is typically for digital use), a conversion must be performed through a modem (which has already been discussed). At the switching centre, data are reconverted into digital and routed through any long-distance lines. Alternatively, there is ISP that can be connected to the network via analogue line (in this case, it will use an appropriate modem bank) or directly through digital line.

Since attenuation, distortion and noise are generally present on the lines, it can be difficult to directly transmit high-speed square waves, and the high-frequency spectral components of which they are made up would be rapidly attenuated over long distances, distorting the signal itself. For this reason, modems that fit the specifications of the digital world to that of analogue telephone lines are used in order to transmit data correctly. Unfortunately to reach high speeds, it is not possible to increase the sampling speed at will, since the bandwidth of 3000 Hz of the analogue phone line at most permits sampling at 6000 Hz, according to the Nyquist theorem.

The number of samples per second is measured in baud: every baud corresponds to a transmitted symbol. If the 0 symbol is represented by 0 V and symbol 1 by 1 V, the maximum frequency is 3000 bauds (theoretical). If, instead, the values 1, 2, 3 and 4 V are used, for each value (and thus symbol), it is possible to combine a pair of bits duplicating the baud rate. Similarly, using four phase values (as

already shown in the QPSK), a pair of bits can be associated with each value, transmitting again 2 bits per symbol, doubling the frequency of bits to baud rate.

Given the scarcity of bandwidth that is available on analogue telephone line, if a direct correspondence bit/baud were used, it would not be possible to exceed 3000 theoretical bits (or bauds) per second. In order to increase the speed of transmission, more phase values are used (QPSK) and more amplitude values (multilevel QAM) as shown in the relevant diagrams called constellation diagrams.

It has already been said that in the presence of several points in the constellation, the relative distance between the same decreases and vulnerability to errors due to noise inevitably present on the line increases. Several standards have been proposed over time for modems. Currently, the maximum attainable speed via modem on an analogue telephone line is 56 kbps. Modems enable communication in both directions in what is called full-duplex mode. If communication takes place in both directions but not at the same time, this would be referred to as half-duplex mode, while if it happened in only one direction, this would be known as simplex mode.

To increase the speed of transmission and to use the full potential of the twisted pair that connects the switching centre with the end user, an ADSL system is used referred to above, which straddles the vocal filter installed by telephone companies to only allow the passage of voice stream, whose frequencies are below 4 kHz.

In this sense, the phone filter is removed and the so-called NID (Network Interface Device) is installed in its place, in the vicinity of the end user. This device contains a bandwidth divider or filter (splitter) that sends signals over the lower band at 4 kHz towards the traditional phone apparatus while signals with a higher frequency are sent to the ADSL modem that is connected to the computer for digital communication. On the other side of the line, within the switching centre, an analogue splitter is installed that splits the signal into speech, to be sent to the telephonic section and to be properly routed over traditional line, and into signal, to be sent to an analogue device to the ADSL modem, called Digital Subscriber Line Access Multiplexer (DSLAM), which sends the data signal to the ISP.

1.2.6 The cellular telephone network

There are two types of wireless phones – cordless phones and cellular phones. Cordless phones are characterised by a limited range (a few tens of meters) and substantially operate as a full-duplex transceiver. Given the limitation of scope and use, we will not focus on such technology, instead examining cellular telephony, given its intensive daily use.

Cellular telephony can be divided into three historical generations:

1. first generation or analogue voice, also called 1G;
2. second generation or digital voice, also called 2G;
3. third generation or voice and digital data, also called 3G.

However, there is also an intermediate generation (the so-called 2.5), which will be outlined below.

1.2.6.1 First-generation cellular network

All cellular telephone systems use a division of the coverage area into cells, hence the specific name. Each cell uses a specific group of frequencies that is not used by the adjacent cells in order to avoid mutual interference. Each cell refers to a fixed system of reception and transmission, called radio base station, connected to the fixed network. The radio base station transmits appropriate antennas, located as high as possible, to increase coverage.

In first-generation systems, a channel corresponds to each frequency and thus a possible phone call. To increase the communicative capacity of the system, the size of coverage of the cells is reduced, a factor that also allows reduction of the output power of mobile terminals, making them lighter and

easy to handle. The cells usually have a roughly circular shape. The various radio base stations that contribute to generation of the coverage cells are connected by a device called mobile switching centre (MSC), which connects and coordinates the various units and is possibly connected to a higher MSC. An MSC is like a fixed telephony switching centre and is connected to the other MSC units and to the fixed telephone network to suitably forward calls.

Every cell phone is always located in the coverage of a specific cell. When the mobile phone moves from one cell to another, the signal of the original cell inevitably fades and the phone looks for a new signal in the destination cell: once the new signal is linked, the phone is transferred under the coverage of the destination cell. This process is called hand-off and is coordinated by the various MSCs. If a telephone call is in progress, the phone must be able to maintain at the same time both the frequency of the coverage cell that it is leaving and the frequency of coverage cell which it is entering in order to avoid interruption of the conversation. This transition process is called soft hand-off. If the old station releases the channel prior to acquisition of the new one or if the new radio base station does not have frequencies available at the time of transition, the conversation is interrupted, even though the phone is under the coverage of the new cell: in this case, this is referred to as hard hand-off. The available channels are divided into channels necessary for the telephone conversation and control channels needed to manage the system. Their division depends on the type of first-generation system.

Every phone contains its own serial number and its own phone number. When the handset is accessed, or in any case periodically, it broadcasts the two numbers that are received from the nearest radio base station that authenticates it, links it and connects it to the network. When the phone has to make a call, it transmits its own data and number to be called on the control channel. If the radio base station has at that moment a free channel, connection is made and the phone is alerted to the fact that this connection is available on the assigned channel, allowing it to commence the conversation. If the channel is not available, the operation needs to be repeated one or several times, until a channel becomes free. If there are incoming calls, the MSC checks under which coverage cell the phone is located, issuing to the relevant radio base station a broadcast presence confirmation request. If the phone responds on the control channel, then the relevant radio base station communicates to the phone the channel to tune into in order to commence communication. The first cellular generation was totally analogue with clear transmission and risky in terms of communication security.

1.2.6.2 Second-generation cellular network

Second-generation cellular networks are fully digital. There are various second-generation standards since, as was the case with first-generation cellular network, a process of global unification had yet to be developed. There is therefore the system Digital Advanced Mobile Phone System (D-AMPS) in the United States and in modified version in Japan and the Global System for Mobile (GSM) communications in Europe and in the rest of the world. The second-generation system works at higher frequencies, being the band of lower frequencies in any case already occupied by first-generation systems. In second-generation systems, voice is digitised and suitably compressed to reduce the number of bits to be transmitted. Communication between phone and radio base takes place on two different frequencies. For transceiving, the technique of time division multiplexing (TDM) is used by dividing each frequency into several frames/s and each frame into time slots within which to allocate the various users wishing to converse.

1.2.6.3 CDMA technology

There is a technology that works in a completely different manner compared to the TDM technique used by both the D-AMPS system and the GSM system: it is known as code division multiple access (CDMA). This technology, instead of dividing the range of available frequencies into sub-channels, directly uses the whole range for all the conversations in progress by adopting suitable expedients. In

fact, every conversation is assigned an appropriate key and the same can only be decoded if the right key is used, otherwise it is perceived as background noise. In CDMA technique, each bit time is divided into short intervals called chips. Usually, a range of 64 to 128 chips for each bit is used. Each transmitter is assigned a unique code called chip sequence: to transmit a bit equal to 1, the station sends the chip sequence while to transmit a bit equal to 0 the station sends the complement of the chip sequence. If each transmitter must send a certain number of bits per second and n is the number of chips for each bit, the amount of information, and therefore bandwidth, will increase in times, expanding the spectrum: for this reason, the CDMA technique is a spread spectrum technique.

In practice, assuming there is 1 MHz of bandwidth available and 100 transmitters, using the technique of frequency division multiplexing (FDM), we would have 1 MHz/100 = 10 kHz that, assuming transmission of 1 bit per hertz, would assign 10 kbps for each station. With CDMA, less than 100 chips per bit need be used to have available, for each station, a higher data rate than that of the FDM. In this way it is possible to solve the problem of channel management, since each station transmits over the whole 1 MHz band available using a different code for each station.

In order for the CDMA technique to operate correctly, all of the sequences must be, two by two, mutually orthogonal, that is their internal normalised product must be equal to zero. To generate sequences orthogonal to each other, the so-called Walsh codes are used. For simplicity of calculation, it is assumed that each component of the sequence equal to 1 is represented with 1, while each component equal to zero is represented with -1. Mathematically, given two vectors **U** and **V**, consisting of n components each, these would be called orthogonal if their internal normalised product $\boldsymbol{U} \cdot \boldsymbol{V} = 0$, which means, in terms of components, that:

$$\boldsymbol{U} \cdot \boldsymbol{V} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{U}_i \boldsymbol{V}_i = 0 \tag{1.28}$$

If the compliment of V is indicated with $\bar{V}$, it is easy to verify that if $\boldsymbol{U} \cdot \boldsymbol{V} = 0$ then $\boldsymbol{U} \cdot \bar{\boldsymbol{V}} = 0$. Given how each sequence has been defined in terms of the number of components equal to $+1$ or -1, it is easy to verify that the internal normalised product of each sequence by itself is equal to 1, that is:

$$\boldsymbol{U} \cdot \boldsymbol{U} = \frac{1}{n} \sum_{i=1}^{n} U_i U_i = \frac{1}{n} \sum_{i=1}^{n} U_i^2 = \frac{1}{n} \sum_{i=1}^{n} (\pm 1)^2 = 1 \tag{1.29}$$

It can also be shown that $\boldsymbol{U} \cdot \boldsymbol{U} = -1$.

At every bit moment, each station can transmit (sending its chip sequence, a 1 must be transmitted or for its negation a zero must be transmitted) or not transmitted. It is assumed that all of the stations are synchronised, emitting possibly related sequences of chips in the same instant. It is clear that if several transmitters emit simultaneously, their chip sequences will overlap. The receiver, to listen to a well-determined transmitter, need simply run the internal normalised product of the overall signal received for the chip sequences of the desired transmitter, appropriately insulating its content. Take as an example of 4 transmitters A, B, C and D, and suppose that the first transmits 1, second transmits 0, third transmits 0 and fourth transmits 1. The relevant signal $\boldsymbol{S}$ that reaches the receiver will be $\boldsymbol{S} = \boldsymbol{A} + \bar{\boldsymbol{B}} + \bar{\boldsymbol{C}} + \boldsymbol{D}$. If the receiver wants to listen D exclusively, it must perform the internal normalised product of the overall signal $\boldsymbol{S}$ for the chip sequence $\boldsymbol{D}$, that is $\boldsymbol{S} \cdot \boldsymbol{D} = (\boldsymbol{A} + \bar{\boldsymbol{B}} + \bar{\boldsymbol{C}} + \boldsymbol{D})\boldsymbol{D} = \boldsymbol{AD} + \bar{\boldsymbol{B}}\boldsymbol{D} + \bar{\boldsymbol{C}}\boldsymbol{D} + \boldsymbol{DD} = 0 + 0 + 0 + 1 = 1$, which is precisely the value transmitted by D, being the normalised products of the other orthogonal sequences between each other and thus equal to zero.

In an ideal CDMA system, one that is noise free, the capacity, that is the number of transmitters, of the channel can be increased at will. In practice, a major problem is the imperfect synchronism between the various transmitters that increase the level of background noise, disturbing decoding of the desired signal. Another requirement is that the signals are received with the same level and this is very difficult in wireless systems where many transmitters are placed at a variable distance from a radio base station

unless use is made of an automatic control system, by the radio base station, of the power emitted by the various transmitters on the basis of the signal level with which they are received.

1.2.6.4 Third-generation cellular network

Cellular telephony was devised with the purpose of providing users in every part of the world with the same type of services, that is high-quality voice telephony, message transmission of any type, multimedia applications, Internet access and many other applications. Initially, two spread spectrum systems were developed: the wide-CDMA (W-CDMA) in Europe and the CDMA2000 in the United States. The W-CDMA standard is currently widely used with the name of Universal Mobile Telecommunications System (UMTS). The latter is designed to maintain full compatibility with the GSM system, being able to pass without problems from a coverage cell of one technology to another with different technology without losing communications in progress. Before definition of the award-winning 3G technology, intermediate standards called 2.5G that are Enhanced Data rates for GSM Evolution (EDGE) and General Packet Radio Service (GPRS) were developed. EDGE is nothing more than a GSM capable of transmitting several bits per baud, being able to use nine different patterns of modulation and error correction. GPRS is instead a packet network built over D-AMPS and GSM that allows the transmission of IP packets on the voice system used in second-generation telephony, using data slots dynamically as time slots assigned to voice communication.

1.3 Data link physical layer

The main purpose of data link layer (also called data link level) is to provide the network layer with services for the delivery of data to the node directly adjacent on the network. The task of the data link layer is therefore to organise the transfer of data between two adjacent pieces of equipment and to provide an interface defined to allow the network layer to access the services provided. Adjacent apparatuses mean apparatuses connected by a channel that on the one hand transmits the bits and receives them on the other hand, in the order of transmission. The data link level uses the services of the physical layer for the delivery of data to its equal process on the receiving computer, but logically communication occurs directly with the process of the remote link data layer.

The structure and characteristics of the channel does not relate to the data link layer but to the physical layer: it does not matter if there is a cable, a fibre, a sequence of different media with interposed repeaters, electrical/optical converters, modems, multiplexers, antennas or other, as illustrated in the sections relating to the physical layer. To accomplish its functions, the data link layer receives data from the upper network layer (packets), organises them in frames, possibly breaking into multiple frames the block of data received from the upper level, adds to each frame a header and trailer and passes everything to the physical layer for transmission.

Upon receipt, the data link layer receives the data from the physical layer, carries out the necessary checks, deletes header and trailer, recombines the frame and passes the received data to the upper network layer.

The data link layer provides the network layer with services of:

1. data transmission without acknowledgement and without connection;
2. reliable data transmission without connection;
3. reliable data transmission with connection.

The class of unreliable service without connection is suitable for high-quality lines. Error control and the retransmission of incorrect frames involve inefficiency in terms of the number of bits transmitted in relation to the data with a reduction of the necessary rate and increase in the likelihood of error. Control may, where appropriate, be entrusted to the upper levels benefiting the efficiency of

the data link level. These services are generally used on local network. It has also already been stated that unreliable services are additionally used for voice and video traffic.

The class of reliable service with connection is suitable for lines more frequently subject to errors: delegating control and retransmission to higher levels (which generally transmit packets consisting of several frames) in the event of high probability of error could cause the retransmission of several packets, while retransmission of the single frame may be sufficient at the level of data link. The class in question is typically used on long-distance lines (WAN connections), even if the optical fibre significantly reduces this problem. The data link level must then be able to offer different classes of service in order to meet different needs. The services are implemented through a series of communication rules (protocols) between the data link levels of adjacent computers.

In order to perform its functions, the data link layer level addresses the following aspects:

1. framing: organisation of the flow of bits into frames, with control for synchronisation, insertion and removal of header and trailer;
2. fragmentation and reordering of frames upon receipt;
3. error handling: error-correction codes or error identification codes must be used and retransmission of incorrect frames should be managed;
4. flow control: a fast transmitter must be prevented from overloading a slow receiver.

It has already been said that the data link level organises data transmissions in blocks (frames) in order to provide monitoring of transmissive errors because the physical layer cannot guarantee the transfer free of errors, which will need to be managed by the level in question. To do this, the data link layer organises the bits into frames and performs checks for each frame. The frame structure must allow the recipient to identify its limits (synchronisation): rules must therefore be adopted to limit it. Since multiple frames are usually involved, the same must be numbered in an appropriate field of the header in order to reorder them upon receipt.

We have seen that the physical layer cannot guarantee delivery of bits without errors. The data link layer must therefore use algorithms to ensure that the frames sent are all received without errors, duplicate-free, in the correct order. Usually, a form of acknowledgement is used that the recipient sends to the sender to confirm correct receipt of the frames. This is done by sending appropriate packets of positive acknowledgement (ACK) or negative acknowledgement (NACK). The check should also include a mechanism for correcting or identifying transmission errors. The complete loss of a frame, or the loss of an ACK, leaves the transmitter awaiting the ACK, and as such a timer must be included for the automatic retransmission of frames. The loss of an ACK implies the retransmission of a frame already received correctly; this eventuality must therefore be identified and the duplicate is discarded through, for example, numbering of the frames. There are different mechanisms used for this function that depend on the protocol being used.

It may also happen that a source is able to transmit at a higher rate than the ability to receive at destination. Without a suitable control, this would imply that the destination would begin to discard frames transmitted correctly due to lack of resources (processing time and buffer). The protocol must be able to manage this situation and provide mechanisms to slow transmission. Typically, the protocol provides control frames with which the recipient can inhibit and re-enable the transmission of frames, that is the protocol determines when the sender can send frames. There are, in this sense, different techniques, which differ in complexity and efficiency of use of the line. There are two strategies for managing physical level transmission errors:

1. Using error correction encoding (forward error correction): the coding used is able to identify the incorrect bits in the frame and correct them upon receipt. Such encoding is typically used on lines with a high rate of error, for which coding overhead is more cost-effective than retransmission of the frame that has a high probability of continuing to be wrong.
2. Using error identification encodings: the coding is able to understand if there has been an error

during transmission; as a result of the error, the protocol requests retransmission of the frame or does not perform any operation and waiting for expiration of the timer in transmission. Such encoding is typically used on lines with low error rate, in which retransmission of the incorrect frame is more cost-effective than the overhead of error-correcting coding.

1.4 Medium Access Control sub-layer

Almost all the geographic and metropolitan networks consist of point-to-point connections due to technical, economic or legal reasons. In fact, the impossibility of laying a proprietary cable along a geographic stretch forces the designer of the geographic network to rely on the lines of telecommunications companies. Even if it was possible to use a proprietary line, in many cases economic factors come into play that favour the hiring of already existing lines with respect to the creation of a proprietary infrastructure. In the case of LANs, these limits do not exist as the area to be interconnected is, usually, entirely under the control of the company or enterprise concerned, as in the floor of a building, a tower block or a campus: in this situation, the designer has a lot of freedom in choosing the technology and the best protocol on the basis of cost, performance, ease of management, robustness, etc. The limited extension of the local networks allows the use of protocols that utilise a transmission medium common to all the interconnected stations (broadcast channels), such as:

1. local networks with transmissive bus (Ethernet, token-bus);
2. shared ring local networks (Fiber Distributed Data Interface (FDDI), token ring);
3. broadcast communication wireless networks, that is non-directional (802.11a/b/g).

The fundamental characteristic of these technologies is represented by the cost-effective availability of a high-performance network and with very low error rates: ratios of 1 to 1,000 are in fact normal on error rates of channels in use over a local network with respect to the channels typical of geographic networks and this allows the creation, as indeed is the case, of data link level protocols that provide unreliable services and delegating controls on the integrity of the data to the higher levels. This consideration does not apply to wireless connections, which indeed are extremely noisy, despite their unique feature that is represented by the support to the mobility of the connected station.

In the networks in question there is a problem that in point-to-point networks does not exist: determining which station acquires the right to use the transmission medium in competitive circumstances (i.e. when more than one station wants to transmit data at the same time interval). The problem is similar to the management of verbal communication between groups of people: there must be a mechanism to adjust the communication because the overlapping of transmissions makes all transmissions incomprehensible. The function of defining the assignment of the shared channel is performed by protocols that are part of the data link layer. The complexity of this issue and its relative independence from issues relating to the transfer of frames means that the matter is treated as a sub-layer independent of the data link level, which is called MAC.

When the physical layer was illustrated, it was shown how the same channel can be used by different users through FDM, WDM or TDM multiplexing techniques. These techniques have the characteristic of being efficient and functional when the traffic of the users is regular and predictable, so as to be able to allocate for each transmission, the resources it requires and that it will generally use for most of the time. In conditions of irregular traffic, the users that have allocated bandwidth or time slots, and that do not use them, constitute inefficiency for the use of the channel.

Computers connected to a local network do not usually produce a regular flow of data to be transmitted, but rather alternate periods of inactivity with moments in which the amount of data to be transferred is high: differences of three orders of magnitude between the peaks of traffic and the average rate produced by a computer connected to the network can easily be found. In these contexts,

the techniques of multiplexing appear to be very inefficient. Another reason for inadequacy is represented by the complexity to be addressed of adding new stations to the local network: if the multiplexing of a configuration is such as to best exploit the resources of the transmission medium, a new inclusion involves the need for an overall reconfiguration of the distribution of the time slots or the frequency bands.

To solve this problem, dynamic assignment of the channel is used. By dynamic assignment, it is meant a mechanism by which the channel is assigned from time to time to the station that needs it, according to specific criteria of the protocols, seeking an efficient way to resolve the disputes. Analysis, both from a logical and quantitative perspective, of management protocols of the above problem is based on certain assumptions that are as follows:

1. Model of the station: it is assumed to be in the presence of N independent stations, each with a user program that generates frames at a constant frequency; when the frame is generated, the station does nothing until the frame has been successfully transmitted (the analysis of models in which the station can use the waiting time to run other programs, thus changing the frequency of generation of the frames depending upon the delay in transmission, is vastly more complex).
2. Existence of a single channel: there is a single transmission channel, through which all communications pass; there is no other way for a station to be able to receive information if not through this channel (as such, it is not possible to establish a mechanism to request allocation of the channel).
3. Existence of the collision: two frames transmitted simultaneously overlap, the signal becomes distorted and incomprehensible; the overlap of a single bit also causes failure of the transmission of both frames. This event is called collision and is the sole cause of transmission error considered. Collisions should be detected by all the stations connected to the medium.
4. Moment of the beginning of the transmission. There are two mutually exclusive alternatives: continuous time and time divided into slots. In continuous time, a station can start to transmit a frame at any time. In time divided into slots, the stations are synchronised between each other and the frames can only be sent at the beginning of the time slots. In this situation, a slot may contain 0, 1 or several whole frames: in the first case, none transmits, in the second case, transmission is successful and in the third case, a collision occurs.
5. Detection of occupation of the channel. Also, in this case, there are two alternatives: the stations, before starting transmission, are able to understand if someone is already in transmission (such as Ethernet); the stations do not perform detection of occupation of the channel. Only after transmission can it be understood if the transmission has been successful (e.g. wireless).

In this sense, there are several known protocols, such as Aloha, time slot Aloha, persistent carrier sense multiple access (CSMA) and non-persistent CSMA, which are not reported due to space limitations. For further information, the reader can refer to bibliography section provided at the end of this book.

A further improvement of performance has taken place with the CSMA/CD (protocols (CD = collision detection). These protocols provide that a station, where a collision on the frame that is transmitting has been detected, interrupts transmission of the frame, thus reducing occupation of the channel with, in any case, invalid frames. In the event of collision, the station waits a random amount of time and tries again. These protocols are the basis of many protocols used on LANs, including Ethernet. The main characteristic of these protocols is represented by the duration of time during which the channel is disputed between stations ready to transmit.

Given the characteristics of the protocols, it is clear that the type of transmission at level two could not be anything other than half duplex, as two simultaneous transmissions cannot coexist on the transmission medium. It can also be observed how the protocols so far illustrated are not able to offer reliable services since collision avoidance does not guarantee that the frame arrives intact. In principle, an acknowledgement technique could be used, but these protocols have an efficiency that collapses

depending on the number of frames transmitted per unit of time. In addition, for cable communications, the high reliability of the physical medium (due to the short distances) makes it more efficient to delegate control to the higher levels rather than weigh down the load with acknowledge frames. In this sense, protocols have been developed to adjust access to the medium that does not involve collisions, such as, for example, the booking protocol and the round robin protocol. By booking protocol it is meant a protocol by which a station announces to all its intention to communicate, before commencing the transmission itself. Since all are aware in advance that the station is about to transmit, nobody interferes with the transmission.

In round robin protocol, each station is given, in turn, the ability to transmit. At its turn, a station transmits the frames available, generally for a predetermined maximum period of time, and then the turn passes to the next station. Control of the sequence may be centralised (a master station that conducts polling of the other stations) or distributed (through exchange of a token), in both cases following a predetermined sequential order. A typical protocol in turn is represented by the token ring (IEEE 802.5 standard) and its variant for bus (IEEE 802.4). Given its importance, being now a reference standard, the rest of the following paragraph will address Ethernet technology.

Ethernet came about as CSMA/CD from a DEC/Intel/Xerox collaboration, standardised in 1978. A few years after, the standard IEEE 802.3 was published, subsequently inherited by the ISO as 8802.3, with minimal differences that were then combined. Normally, Ethernet and IEEE 802.3 are used as synonyms. Ethernet, meant as a technology, has developed since the first 10 Mbps version, which was followed by a new 100 Mbps standard, then one of 1,000 Mbps and a subsequent one of 10 Gbps. The standard provides four different types of wiring for Ethernet:

1. 10Base5: thick type coaxial cable, with a maximum length of 500 m;
2. 10Base2: thin type coaxial cable, with a maximum length of 185 m;
3. 10BaseT: twisted pair, with a maximum length of 100 m;
4. 10BaseF: optical fibre capable of connections up to 2,000 m.

The nomenclature used indicates the speed (10 Mbps), the fact that the transmitted signal is base band and (in the first two cases) the length expressed is in hundreds of metres. The 10Base5 wiring is performed using thick type coaxial cable of maximum length 500 m. Is it possible to connect to the cable, at a distance of 2.5 m, vampire sockets connected to the stations. A transceiver is attached to the socket, called vampire, which is the analogue module that controls the cable to detect collisions. The transceiver is connected to the network interface on the computer network by a power cable that can be up to 50 m. The transceiver cable generally consists of five pairs, of which two are dedicated to two way traffic, two to the control and an optional one to powering of the transceiver itself. There are transceivers to which up to eight stations can be connected. The coaxial cable must be terminated at the two ends of a 50 Ω "cap" in order to eliminate reflections. The main problems of this type of cable are represented by the rigidity of the cable, the difficulty in identifying the source of any problems (excessive lengths, defective sockets, total or partial interruptions of the conductor) and by the technical difficulty of the inclusion of new stations via the vampire sockets.

The 10Base2 wiring uses a thin type coaxial cable and also ended in two 50 Ω impedance caps. The connections are made via Bayonet Neill Concelman (BNC) connectors in the form of a T that allows connection to the T of a station interface (or a thin cable that leads to the interface of the station, but of a much shorter length, as it introduces reflections). In this solution, the transceiver resides directly in the network interface of the connected station. The advantages with respect to the 10Base5 are represented by the easy handling of the addition of new stations (if the T is already in place, otherwise the cable has to be interrupted) and by the reliability of the connectors (greater than that in the case of 10Base5, but in any case source of possible problems).

The 10BaseT wiring makes use of a wiring diagram based on copper twisted pairs. Each station is connected via a UTP cable (category 3 or higher) to a device with multiple ports named hub. The hub does not process data, but it is the shared medium from a logical point of view: copper wires are

connected by the electronics within the hub in such a way as to simulate the shared medium. The hub performs the functions of a repeater, which regenerates the signal and sends it to all the connected lines (except the one from which it has received the frame). If there is a simultaneous transmission of two or more stations connected to the hub, there will be a collision. The use of this wiring technique offers many advantages from a practical point of view, represented by the simplicity of wiring (often being able to utilise existing telephone wiring), by the simplicity in adding, removing, or moving of connected stations, by the mechanical reliability of the physical medium and simplification of troubleshooting. The disadvantage of this solution is represented by the limited distance that is 100 m for the UTP category 3 and 200 m for the UTP category 5.

The 10BaseF wiring uses a pair of optical kilometre, with a maximum length of 2 km. This wiring is generally used to interconnect buildings or distant locations. The specifications provide the possibility of hub interconnection, stations and repeaters. To increase the distance covered by the network, it is possible to connect multiple cables to each other via repeaters. From the data link layer point of view, the only difference of a structure with repeaters is the transmission delay introduced by their presence. The standard provides limits on the extensibility of the network through repeaters that are: between two transceivers there can be no more than 4 repeaters; between two transceivers there can be no more than 2.5 km of relative distance. The use of repeaters allows the development of different topologies for the wiring of a building.

On the shared medium, the condition of lack of transmission is necessarily identified by the absence of signal. Encodings that use the 0 V signal to identify a bit are not therefore possible. The need to transfer clock information together with the signal led to the invention of Manchester encoding, already illustrated in the previous paragraphs. The Ethernet standard uses Manchester encoding with $+0.85$ V and -0.85 V signals (other protocols, such as token ring, make use of the differential Manchester encoding). The Ethernet address is normally shown as a sequence of 6 bytes, each represented by a pair of hexadecimal digits, separated by a couple of points or by a hyphen. The maximum size of the data field of the Ethernet frame is 1,500 bytes.

As the number of stations connected increases, so does the inefficiency of the protocol that handles them. To solve this problem a device is used called switch. The switch is an object consisting of a high transmission speed internal card (backplane) on which various line cards can be grafted, each containing different connectors. The connectors are designed for 10BaseT twisted pairs; each connects a station (or a hub or another switch) to the network. When a station sends a frame, the latter reaches the switch. The switch is able to identify to which port of which card the station to which the frame is intended is connected: if the recipient station is connected to the same port of the same card, the switch removes the frame; if the recipient station is connected to a different port on the same card, the frame is sent on that port; if the recipient station is connected to a different card, the frame is transmitted within the destination card through the backplane and from there sent on the port connected to the recipient station. The backplane card works using a proprietary protocol, developed by the producer, generally with a capacity much higher than 10 Mbps.

It can often happen that two stations connected to the same card transmit simultaneously. In first-generation switches, the card is in fact a hub: all the lines are electrically connected to form a single collision domain and the simultaneous transmission causes a collision managed according to the contention protocol. Collision, of course, only concerns those stations connected to the card in question; in this case, only one transmission for each card is possible, but different cards can transmit frames in parallel. More modern switches have a buffer for each port: the frame is stored and sent on the port of destination as soon as possible (transmission mode called store and forward). In this case, there is no chance of collision because each port can transmit and receive at the same time without affecting the transmissions of others; the switch manages storage of a frame on buffer if it cannot be forwarded immediately. In this way, there is a full band full-duplex communication.

It is possible to use a number of ports of a switch as line centraliser: a port can be connected to a hub or to another switch, so as to separate collision domains. This technology allows overall efficiency

in high-load conditions to be greatly increased by, in fact, eliminating the problem of collisions or limiting it within distinct branches containing a small number of stations. Using switches in cascade, tree topologies can be created making the topological structure of the network very flexible and its development over time simpler.

To discover on what port the frame has to be transmitted, the switch must create and keep updated a table on the association between destination address and port address. The manual construction of this table would be too costly in terms of network management and a suitable mechanism for self-learning was designed. Initially, this table is empty and the switch must forward each frame received on all the ports that are connected. Since the frames contain the address of the sender, for every frame that arrives, the switch learns that the station that has sent the frame can be reached through the port from which the same frame arrived. With the passing of time the switch fills the table and can perform its function even more efficiently. All of the broadcast and multicast frames will continue being transmitted on all the ports that are connected (except that of origin), as well as the frames intended for addresses not in the table. The addition of connected stations is handled by the switch automatically through the mechanism of self- learning.

The functional limits of the switch are determined by its ability to transmit the frames at the necessary speed. Since the switch allows a full-duplex transmission on all the ports of each card, the backplane may limit the ability to support the traffic generated. In modern high-quality switches, the backplane is constructed in such a way as to ensure an outflow sufficient for full bandwidth transmission of all its ports at the same time. In older switches or in those of lower quality, the ability of the backplane is in any case very high and it is used the fact that it is rare for all stations to transmit at full bandwidth at the same time. Another problem is the limitation of the buffers. In fact, if it is assumed that two stations transmit at full bandwidth towards a third station, the switch receives a traffic of 20 Mbps in input, but has only 10 Mbps in output towards the destination; in such a situation, it is not possible to unravel all the traffic. In all cases, the excess frames will be removed from the switch and it will be the responsibility of the upper levels of the stations concerned to manage the situation with retransmissions and flow control.

In 1992, IEEE met the 802.3 committee to develop a 100 Mbps protocol based on Ethernet technology. The work was developed according to the fundamental guideline of maintaining compatibility with existing LANs, keeping, therefore, the same frame format, the same interfaces and the same procedural rules. An increase in the speed by a factor of 10, with the same minimum length of the frame, requires that in order to detect collisions, the maximum cable length should be shortened by a factor of 10. This would not have allowed maintenance of the pre-installed wiring structures and therefore the solution was to give up the coaxial cable. FastEthernet provides as possible topologies only connections via hubs or switches, using as means of transmission:

1. UTP category 3: 100Base-T4 (maximum 100 m);
2. UTP category 5: 100Base-TX (maximum 100 m);
3. optical fibre: 100Base-FX (maximum 2,000 m).

A 100 Mbps Manchester encoding requires 200 Mbaud, a value that is unfeasible for twisted pairs at the required distances: it was therefore decided to change the encoding. Modern apparatus, which manages the clock more accurately, and small distances allow forgoing of the benefits of the Manchester encoding.

The 100Base-T4 represents the standard for UTP category 3 that provides for the use of four twisted pairs with 25 MHz ternary signals (supported by the cable at 100 m distance). A twisted pair dedicated to transmission in one direction is used, one for that in the opposite direction, two that can be switched; a ternary signal is transmitted: with three twisted pairs there are 27 symbols that can transfer 4 bits of information with little redundancy; 25 MHz for 4 bits provides the 100 Mbps required, but not full duplex.

The 100Base-TX uses UTP category 5 at 100 m and is capable of supporting a frequency of 125 MHz. The standard provides for the use of two pairs (one for each transmission direction) using a two level coding called 4B/5B. Each group of 5 clock periods contains 32 combinations: 16 are used to transmit 4 data bits, some of the other for control functions; the 16 combinations dedicated to data were suitably chosen to ensure an adequate number of signal transitions in order to facilitate synchronisation upon receipt; 4 bits each 5 clock periods at 125 MHz provides the desired 100 Mbps, for each cable pair, guaranteeing full-duplex communication.

The 100Base-FX uses a connection made through a pair of multimode kilometre (one for each direction) capable of a maximum distance of 2,000 m that use 125 MHz 4B/5B encoding converted into optical signal. The standard thus defined allows use of the same rules of Ethernet protocol. For the copper connections, tree topologies via a hub or switch are possible. Each hub constitutes a collision domain; the collision is managed with the dispute mechanism regulated by the Ethernet algorithm. The maximum cable length for operation of the algorithm based on detection of the collision is 10 times less than the limit for Ethernet, thus equal to 250 m, compatible with the maximum length of UTPs. For fibre connections, the length of the specifications exceeds the maximum allowable for the correct management of collisions, for which 100Base-FX may only be used with switches.

All switches can use mixed speed connections, with 10 or 100 Mbps port. The port speed can usually be negotiated by the two interfaces at the time of ignition of the machines, as well as the mode of transmission (half duplex or full duplex). In this way, a migration of the network from Ethernet to FastEthernet can be planned without having to change all the switching apparatus and interfaces at the same time. The old network interfaces, manufactured according to the Ethernet standard, are not able to negotiate but switches can understand alone and automatically configure the port appropriately. Quality switches can be configured manually to define the mode of operation of the ports. Fibre ports do not have these features: for fibre connections if the switch technology is changed, so must the interface.

1.4.1 Wireless networks

Given their importance, in the following will be explained the main types of wireless networks currently used and represented by WLANs, by broadband wireless and by the Bluetooth system.

1.4.1.1 Wireless LANs

WLANs are a strong competitor of Ethernet, allowing connection with great ease, without cables, in any place in which cover with this system has been provided.

As we have seen, they can operate in the presence or absence of a radio base station or AP. The most widespread standard is represented by IEEE 802.11 that allows operation in both modes. Some general information has already been provided. The following paragraph will provide further information on this standard.

The protocols of the 802 family all use a similar structure, including Ethernet. As can be seen from Figure 1.57, which shows the stack of protocols of the 802.11 standard, the physical layer corresponds to the ISO OSI physical layer but the data link layer is divided into two or more sub-layers: the MAC sub-layer defines what method to use for allocating the channel while the logical link control (LLC) sub-layer is aimed at making the different variants of 802 indistinguishable to the network layer.

In 1997, this standard had three transmission techniques based on infrared, over low-power radio waves using the technique frequency hopping spread spectrum (FHSS) and on the low-power radio waves using the technique direct sequence spread spectrum (DSSS). The last two techniques operate on the ISM band at 2.4 GHz. In 2001, two new techniques called orthogonal frequency division multiplexing (OFDM) and high-rate DSSS (HR-DSSS) were introduced into the standard. These techniques, although relevant to the physical layer, are addressed in the following paragraph under

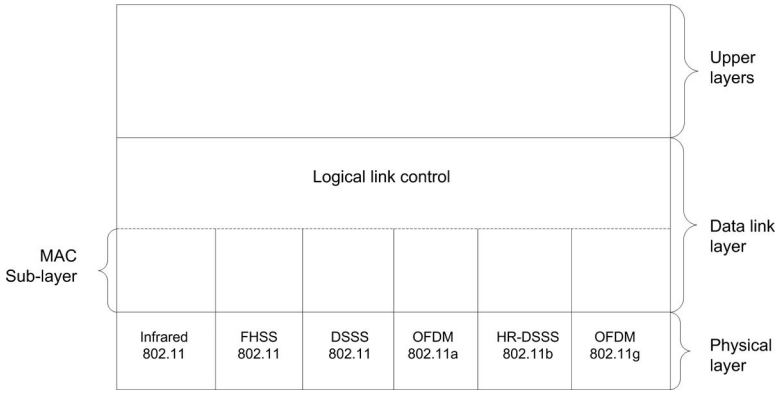


Figure 1.57 Diagram of the protocol stack of the 802.11 standard.

MAC layer of the data link layer due to their close connection with the latter. All five techniques provided by 802.11 allow the sending of frames by one equipped computer to another but use different speeds and technologies and for this reason they will be briefly illustrated below.

In infrared technique, use is made of a widespread emission that uses the wavelengths of 0.85 and 0.95 micron with speeds of 1 and 2 Mbps. This technique is not widely used as infrareds are not able to pass through obstacles, limiting coverage of the system that uses it. In addition, infra reds are subject to interference by solar radiation.

In FHSS technique, 79 channels are used, each with a width of 1 MHz, allocated within the ISM band, using a pseudo-random number generator (which will be described in greater detail in Chapter 2) to produce the hopping sequence of frequencies. In order to synchronise the transmitter with the recipient in the hop sequences, the same must use the same seed for the pseudo-random generator (this concept will be explored later). Rotation time is defined as the time in which the system emits on a specific frequency, which must be less than 400 ms. FHSS technology ensures a certain level of security as a potential intruder, in order to intercept communication, would need to know the hop sequence and the rotation time, which are the parameters only available to users of the network. It is also fairly robust in relation both to the multiple paths that occur when the distances between transmitter and receiver start to increase and in relation to electromagnetic interference. Its only disadvantage is the limited bandwidth. With DSSS technology speeds of 1 or 2 Mbps are reachable. It uses a system similar to CDMA (shown above) with the difference that for each transmitted bit 11 chips of an appropriate sequence, called Barker, are used.

In OFDM technology, used by the first high-speed LAN called 802.11a, a speed of up to 54 Mbps on ISM bands placed around 5 GHz was achieved. This technology works in a manner similar to the ADSL system (explained above), using 52 frequencies, 48 of which are dedicated to the transmission of data and four are dedicated to synchronisation. Given emission on several frequencies at the same time, this technique is similar to those of spectrum diffusion such as FHSS and DSSS or CDMA. Using multiple bands has the advantage of greater immunity to disturbance concentrated in frequency.

In HR-DSSS technology, a speed of 11 Mbps is achieved, in the 2.4 GHz band, using 11 million chips per second. It is also known by the name of 802.11b. It is slower than 802.11a but is characterised by a flow rate seven times higher than the latter. In 2001, a new standard was developed, called 802.11g, which uses the OFDM modulation technique of 802.11a, but works on the restricted ISM band located at 2.4 GHz like 802.11b. Subsequent to this standard many others have been developed but will not be shown for reasons of space. Reader can refer to bibliography section for further information.

Two typical problems of LAN wireless systems are represented by the hidden station (shown above and addressed here) and by the exposed station. These problems arise from the fact that not all stations

are in the radius of coverage of the others and transmissions that take place in certain areas of the network may not be picked up in the other areas of the network (Figures 1.58 to 1.60).

In the problem of hidden station, C is transmitting to B and A, not being able to know that B is busy, thinks it can transmit freely towards B. In the problem of exposed station, B wants to transmit to C and checks the channel finding it occupied by A that perhaps is transmitting to a fourth station different from C. For these and many other reasons, the 802.11 standard does not use CSMA/CD, as in the case of Ethernet, but a specific coding technique.

Wireless networks are usually noisy and unreliable, in contrast to wired networks, due to the presence of numerous sources of noise (such as microwave ovens) that operate on the same ISM frequency band. To avoid this problem, the 802.11 standard provides for fragmentation of the frames into subframe, each equipped with its own control field. In this way, in case of an error in transmission, the entire frame need not be transmitted again but only the subframe that has not correctly reached the destination, optimising performance of the system.

In the 802.11 standard, there are nine types of services divided into two categories. In one category, five distribution services are provided that relate to the management of belonging to the cell and the interaction with the stations that are located outside of the cell, whereas in the other category, four station services are provided that relate to the activities within the single cell. The distribution services are:

1. association, which is a service available to mobile stations for connection to radio base stations;
2. separation, which is a service that enables the mobile station or the radio base station to separate itself, ending communication;
3. reassociation, which is a service that allows mobile stations to vary their radio base station at will;
4. distribution, which is a service that establishes in what manner direct frames are routed to radio base stations;
5. integration, which is a service that manages translation from the 802.11 format to the format required by another network for relevant forwarding of data.

The services stations are:

1. authentication, which is a service that only allows authorised mobile stations to exchange data with radio base stations;
2. invalidation, which is a service that enables a mobile station to leave the network after being authenticated;

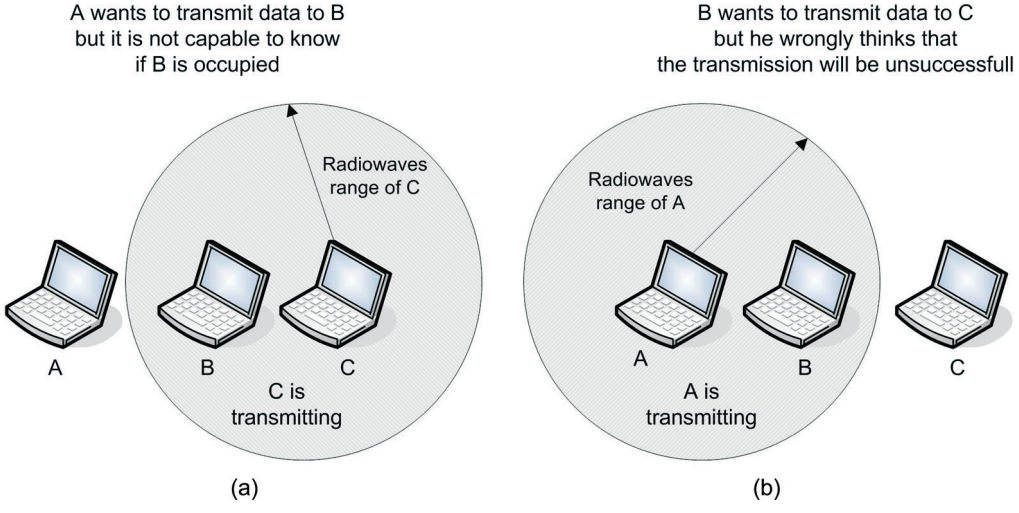


Figure 1.58 (a) Problem of hidden station and (b) problem of exposed station.

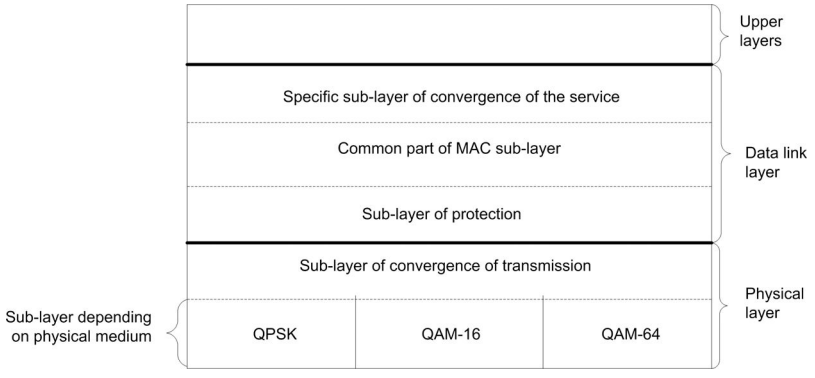


Figure 1.59 Diagram of the protocol stack of 802.16.

3. confidentiality, which is a service that encrypts and decrypts (as will be shown in more detail in Chapter 2) transmitted data to make transmission secure;
4. data transfer, which is a service that in fact manages data transfer, leaving detection management and error correction to the upper layers.

1.4.1.2 The 802.11 standard

There are several standards of the 802.11 family that will be illustrated in the following.

The 802.11 a standard

In 2001, protocol 802.11a was ratified and approved in 1999. This standard uses the frequency space of around 5 GHz and operates with a maximum speed of 54 Mb/s, although in reality the real speed available to users is about 20 Mb/s. The maximum speed can be reduced to 48, 36, 24, 18, 9 or 6 Mb/s if electromagnetic interference so dictates. The standard defines 12 non-overlapping channels, eight dedicated to internal communications and four for point-to-point communications. Almost every nation has issued a different directive to adjust the frequencies but after the world conference on radio communication of 2003, the US federal authorities decided to make the frequencies used by the 802.11a standard free, according to the criteria already seen.

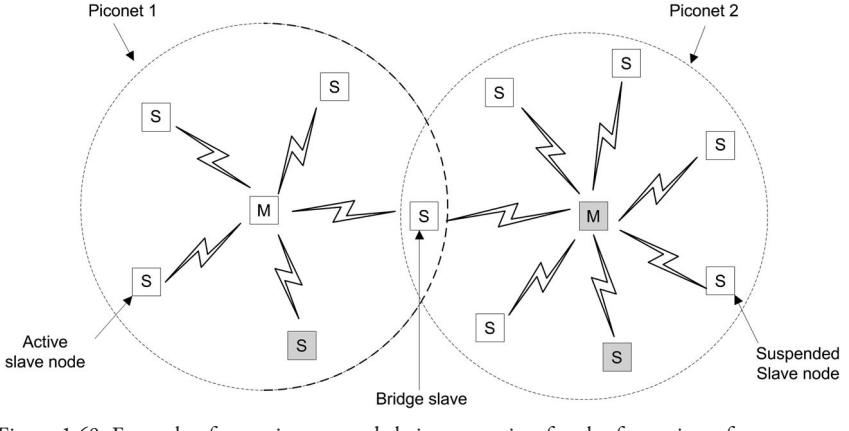


Figure 1.60 Example of two piconets and their connection for the formation of a scatternet.

This standard did not win the approval of the public since 802.11b was already widespread and in many countries use of 5 GHz frequencies is still reserved. In Europe, the 802.11a standard was not authorised for use given that those frequencies were reserved to high-performance radio LAN (HIPERLAN), and it was only in the mid-2002 that these frequencies were liberalised after which it was possible to use 802.11a. There are dual-standard or tri-standard cards able to also accept (in addition to standard a, b) even the tri-standard cards as g. Obviously, there are also multi-standard points of access.

The 802.11 b standard

802.11b can transmit a maximum of 11 Mbit/s and uses CSMA with collision avoidance (CSMA/CA) as a method of transmitting information. A good part of the available bandwidth is used by CSMA/CA. In practice, the maximum transfer obtainable is 5.9 Mbit/s in TCP and 7.1 Mbit/s in UDP. Metal, water and, in general, solid obstacles drastically reduce the range of the signal. The protocol uses the frequencies of around 2.4 GHz.

Using external directional antennas with high gain, it is possible to establish point-to-point connections of the radius of several kilometres. Using receivers with a gain of 80 dB, up to 8 km can be reached or if weather conditions are favourable even greater distances but these are temporary situations that do not allow reliable coverage. When the signal is too noisy or weak, the standard provides for a reduction in the maximum speed to 5.5, 2 or 1 Mb/s to allow the signal to be decoded correctly.

Proprietary extensions were developed that, using multiple coupled channels, allow an increase in transmission speed to the detriment of the compatibility with devices produced by other manufacturers. These extensions are normally called 802.11b+ , and bring the theoretical bandwidth to 22, 33 or even 44 Mb/s. 802.11b and 802.11g divide the spectrum into 14 sub-channels consisting of 22 MHz each. The channels partially overlap over each other in frequency and thus between two consecutive channels there is strong interference. The two groups of channels 1, 6, 11 and 2, 7 and 12 do not overlap each other and are used in environments with other wireless networks. The only channels that can be used throughout the world are 10 and 11 given that Spain has not granted channels from 1 to 9 and many nations are limited to the first 11 sub-channels.

The 802.11 g standard

This standard was ratified in June of 2003. It uses the same frequencies of the 802.11b standard, that the bandwidth of 2.4 GHz and provides a theoretical bandwidth of 54 Mb/s which in fact is reflected in a net band of 24.7 Mb/s, similar to that of the 802.11a standard. It is fully compatible with standard b but, when it has to work with b devices, it must obviously reduce its speed to that of standard b. Before official ratification of the 802.11g standard, which took place in the summer of 2003, there were independent producers who provided the equipment based on non-definitive specifications of the standard. The main manufacturers however preferred to adhere to the official specifications and when these were published, many of their products were adapted to the new standard. Some producers introduced further variants called g+ or Super G into their products. These variants used a coupling of two channels to double the bandwidth available, even though this induced interference with other networks and was not supported by all the cards.

The 802.11 i standard

IEEE 802.11i (also known as WPA2) is a standard developed by the IEEE specifically to provide a layer of security to communications based on the IEEE 802.11 standard. The document was ratified on 24 June 2004 and is a superset (extension) of the previous standard Wired Equivalent Privacy (WEP),

which was shown to be subject to conceptual project errors. Before the 802.11i standard, the Wi-Fi Alliance had introduced Wi-Fi Protected Access (WPA) a subset of the 802.11i specifications. The WPA was introduced to buffer any security emergency due to WEP and represents only one transitional standard while 802.11i was completed and perfected. The Wi-Fi Alliance decided to call the specific 802.11i with the name of WPA2 to make it easy for the common user to identify the cards based on the new standard. 802.11I uses as cryptographic algorithm Advanced Encryption Standard (AES) unlike WEP and WPA that use RC4.

The architecture of 802.11i uses the following components: IEEE 802.1x to authenticate (the EAP protocol or an authentication server can be used), the RSN protocol for keeping track of associations and CCMP to ensure the confidentiality, integrity and certainty of the sender. The authentication process takes place by means of a four-way handshake. This standard will be illustrated in more detail in Chapter 6.

The four-way handshake process is based on two considerations. AP must authenticate itself and the session key used to encrypt messages must still be calculated. First, EAP should exchange its private key (PMK) with AP. But this key should be detected as little as possible and only on a secure channel because it is the keyword that protects all communications and as such the four-way handshake comes into play. Second, EAP transmits to AP a temporary PTK key. The PTK is generated by concatenating PMK, AP nonce (Anonce), STA nonce (Snonce), AP MAC address and STA MAC address. The product is sent to a hash cryptographic function. The protocol uses the temporary GTK key to decrypt the multicast traffic.

The 802.11 n standard

The reasons for which almost all the major manufacturers of wireless systems threw themselves into the presentation of devices based on the 802.11 *n* standard was the strong public demand for higher performance than that allowed by standards such as 54 Mbps 802.11g. Current wireless Wi-Fi systems have, in fact, many limits if compared with wired connections, first of which is the speed that is lower and more variable, due to several factors. The second limitation is the range that is sometimes not sufficient to adequately cover large enclosed spaces such as an apartment, in fact, for example, that is not appreciated in an office. A solution to these problems is a new standard that implements appropriate technologies to improve performance and it is for this reason that all the efforts of producers have concentrated on 802.11n.

Designed in any case to be compatible with previous standards, 802.11n allows connections to be created with a speed that is far higher than in the previous 802.11g standard. Due to the frequencies adopted, the devices with 802.11n standard can operate on the 2.4 GHz band, the same one used by 802.11g and 802.11b, but also on the 5 GHz band or on both depending on the implementation made by the manufacturer. The modulation chosen for this protocol is OFDM. One of the solutions found to overcome the problems of range and speed, however, is the technology called multiple input multiple output (MIMO). MIMO is a technology that allows improvement of the quality of radio transmissions and/or an increase in the baud rate through the use of multiple antennas both in transmission and in receipt. The term MIMO, however, does not in general define the way of using multiple antennas that varies greatly depending on the application and physical transmission channel that is available.

MIMO technology is particularly interesting when applied to wireless communication, since it offers considerable improvements in throughput and in transmission distance without resorting to additional bandwidth or higher transmission power but through increased spectral efficiency (more bits per second per hertz of bandwidth) and higher connection reliability. The three complementary aspects that cohabit in each MIMO system are diversity, spatial multiplexing and beamforming.

Diversity relates to the transmission and/or reception of the same symbol through different antennas. If the antennas are physically separated by a distance that makes the signals received

independent of each other, it is probable that when an antenna does not have a good SNR, transmission can still be correctly decoded from the signal of adjacent antennas. Since this is a type of spatial redundancy, the baud rate of the signals does not increase but increases its reliability.

Spatial multiplexing was historically the first use designed for MIMO systems, so that more information could be encoded and divided on different antennas and transmitted simultaneously on the same band. In this situation, the receiver antennas see a resultant field that is the superposition of all the signals in transmission, which will typically have passed along different paths and will have been subject to multiple different reflections. It can be shown mathematically that if the various paths are actually independent, it is possible to reconstruct all of the information transmitted from all the signals present on receiving antennas. Spatial multiplexing allows, on transmission channels with a high SNR and very rich in scattered reflective objects, a considerable increase in the baud rate. Spatial multiplexing is feasible as long as in receiving there is a number of antennas greater than or equal to the number of flows of information transmitted in parallel. Conversely, the use of MIMO in diversity does not place restrictions on the number of antennas in transmission and reception.

Beamforming instead, in general, allows directing of the radiated power in a non-uniform manner into space, but in preferential directions in which, for example, users can be localised. In MIMO systems, when the transmission channel is known to the transmitter, a transmission encryption called "eigen-beam forming" (i.e. beam forming based on spatial modes of the channel) can be adopted, which optimises power usage and maximises transmission of the information between all the transmitting and receiving antennas. Eigen-beam forming is especially usable when it is possible to estimate the transmission channel with great precision and when the change in time of the channel itself is not too fast. When applicable, eigen-beam forming also provides a considerable increase in the capacity of the channel with respect to Single Input - Single Output (SISO) systems.

The MIMO technologies used in commercial systems are based on a combination of the three above-mentioned aspects. Given this genesis, it is clear that it is not possible to give a MIMO system maximum diversity and at the same time maximum degree of spatial multiplexing. The system must therefore be designed as a compromise between the two aspects, which are in opposition and are regulated by the so-called "diversity-multiplexing trade-off" (i.e. compromise between diversity and multiplexing). The use of eigen-beamforming, in turn, implies a certain degree of diversity and spatial multiplexing. In the recent apparatuses of wireless broadband access and in mobile phones, adaptive MIMO systems are being designed that will allow movement in such a manner as to choose at any time the type of MIMO modulation that is more suited to the conditions of the transmission channel at a given moment.

Continuing with the 802.11n standard, to further improve performance, a doubling is added, from 20 to 40 MHz, of the bandwidth of the channel compared to the 802.11g version for the physical layer, and a higher maximum baud rate, always in relation to that allowed by the 802.11g. Transmitters and receivers, moreover, use techniques of pre-coding and post-encoding. All of these combined elements meet the needs according to the 802.11n design to significantly increase the different performances compared to those possible with the previous standards. In fact, a number of components already on the market and based on draft version declare speed of 300 Mbps and very significant ranges. To create 802.11n devices, single chips are now available that incorporate all the necessary aspects.

From a constructive point of view, the 802.11n devices, depending on the number of streams, use two transmitters and two receivers for the standard version in terms of speed that pass to four receivers and four transmitters for the higher speed. With regard to the transmission power, the values are essentially the same as found in standard 802.11g, but the overall power is divided into equal parts between the different antennas.

The system however is very flexible: if the device that transmits replicates the same signal on all the channels, a greater range capacity and stability of the connection is consequently obtained instead of an increase of the baud rate, with respect to a single channel. The data are transmitted over a single

channel and in the same way as provided by the 802.11 protocol, which allows combination of the normal packets with the MIMO system.

With regard to the operation modes, 802.11n provides different systems. The first system is called legacy that operates practically as the previous ones. This means that the devices must operate in the same frequency range of the other standards and support channels with a bandwidth of 20 MHz. The second system, however, is a mixed type and allows communications between the previous cards and those 802.11n ones and, finally, the third mode is called native. The first two modes however offer other advantages, such as the increase in range compared to 802.11g.

One of the problems, on the other hand, is interference with 802.11 b and g devices that was remedied by moving the operating frequencies and the mode of operation due to special algorithms. This is in addition to the technology of aggregation of the frames that is a system that allows improvement of aspects such as efficiency and speed in the presence of small packets, allowing the reduction of latency when a device tries to communicate with the AP. Efficiency, moreover, was also improved by reducing the use of bandwidth for control data.

Other standards of the 802.11 family

In addition to the revisions of the original standard, designed to optimise bandwidth capacity (as in the case of standards 802.11a, 802.11g, called 802.11 physical standards), other standards were published that aimed to specify the elements to ensure better interoperability security. Table 1.5 shows the rest of the standards and significance of the 802.11 family.

1.4.1.3 Broadband wireless

Broadband wireless was conceived to solve the problem of the last mile, that is, for fast connection in the stretch that connects the telephone exchange with the end user that, traditionally, is ensured by a normal telephone twisted pair. To increase the connection speed, this twisted pair should be replaced with optical fibre, performing a large number of excavations for all users to be reached.

Broadband wireless solves the problem because using an appropriate radio base station makes it possible to serve a large number of users that can connect by simply pointing their dish towards the same radio base station. The reference standard was developed in 2002 by the IEEE with number 802.16 that was also called, briefly, MAN wireless or local wireless connection. This standard was heavily influenced by the ISO OSI model.

Even if the aims of this standard are the same as those of 802.11 (to provide broadband wireless communication), the first goal of all is that 802.11 is aimed at providing services to mobile users that can move from cell to cell while 802.16 provides services to entire fixed buildings. In addition, 802.11 does not communicate in full-duplex mode unlike 802.16. In addition, 802.16 was designed to ensure a higher speed and greater bandwidth than 802.11 and for this reason it operates between 10 and 66 GHz, frequencies that are easily absorbed by the rain, making the system more vulnerable to errors. While 802.11 was designed to provide mobile Ethernet connections, 802.16 was designed to provide telephony and multimedia services to fixed users. The stack of protocols of 802.16 is very similar to that of 802.11 but it is characterised by a greater number of layers.

The lowest substrate is dedicated to transmission, using narrow-band radio communication with different modulation schemes. Above this sub-layer, there is a sub-layer of convergence that serves to hide the technological differences from the upper layer.

The data link layer is composed of three sub-layers: protection, MAC and convergence. The protection underlayer is responsible for encryption, decryption and key management. The MAC sub-layer is responsible for management of the channel, being oriented towards connections in order to ensure quality of service at the level of telephone and multimedia communications. The convergence sub-layer is intended for interfacing with the top network layer.

Table 1.5 Additional 802.11 families.

Name of standard	Name	Description
802.11c	Bridge from 802.11 with 802.1d	The rule 802.11c does not affect the general public. It is merely a modification of the standard 802.1d in order to establish a bridge with 802.11 frames (data link layer).
802.11d	Internazionalization	The 802.11d standard is a supplement to the standard 802.1 whose purpose is to allow international use of local networks 802.11.Is to allow different devices to exchange information on the list of frequencies and powers authorized in the country of origin of the material.
802.11e	Improvement of the quality of service	The 802.11e standard aims at giving some possibilities in matter of quality of service at the level of the data link layer. Thus, this standard has as purpose to define the needs of several packages in terms of bandwidth and transmission time in such a way as to allow especially a better transmission of voice and video.
802.11f	Roaming	The standard 802.11f is a recommendation to the intention of the sellers of access points to improve product interoperability. Proposes the Inter-Access Point Protocol roaming protocol that allows a user to change his itinerant access point in a transparent way during a move, regardless of the brand of access point in the infrastructure network. This possibility is called roaming.
802.11 h		The 802.11 h standard aims to bring the 802.11 standard of the European standard (HIPERLAN 2, where h802.1 h) and comply with the European regulations regarding frequency and energy savings.
802.11Ir		The norma802.11r has been developed in such a way to use infrared signals. This standard is technically out of date.
802.11j		The standard 802.11j is the Japanese regulation such as 802.11 h is the European regulation.

With the use of the frequency band between 10 and 66 GHz, thanks to the greater directionality of these waves, a single radio base station can be used, with antennae pointed towards the different areas to be covered, without which their beams interfere with each other. In addition, since the level of the signal emitted decreases rapidly as distance from the radio base station increases, worsening the SNR, three modulation schemes are used depending on the distance of the end user from the radio base station. They are, in ascending order, QAM-64 for short distances, QAM-16 for medium distances and QPSK for long distances.

1.4.1.4 Bluetooth

The Bluetooth standard was devised to allow short-distance connection between various devices. It takes its name from the Viking king Harald Blaatand (Bluetooth) that unified Denmark and Norway.

This standard has the same purposes as 802.11, although over lesser distances and, unfortunately, interferes at the level of communication with the latter.

Bluetooth was designed to operate on piconets, networks consisting of a main node, called master, and with not more than seven active nodes located at a distance of less than 10 m. Several piconets may be located in the same area and may be connected to each other giving rise to what is called scatternet.

The system can manage up to 255 nodes of which only seven are active at the same time, the others are in hibernation mode controlled by the master, in order to reduce power consumption. A device in a state of suspension can only respond to a request for activation by the master.

The system is based on centralised TDM, where the node master controls the clock and enables the various nodes to communication. Communication only occurs between the master node and the slave nodes and not between the various slave nodes. Bluetooth, in contrast to the other standards, accurately defines the type of specific applications that it can support (13 in all), which are as follows:

1. generic access;
2. discovery of the service;
3. serial port;
4. exchange of generic objects;
5. LAN access;
6. dial-up access;
7. fax;
8. cordless telephony;
9. intercommunicators;
10. wireless headset;
11. object sending;
12. file transfer;
13. synchronisation.

Bluetooth is characterised by different protocols that are not strictly ordered in layers, following neither the ISO OSI organisation, the TCP/IP model, nor the 802 model.

As can be seen from Figure 1.61, on the lowest part is the physical radio transmission that also deals with modulation. Above there is the base band layer that is characterised by functionality similar to the MAC sub-layer but includes certain features of the physical layer. It manages the manner in which the root node controls time intervals and the grouping of these intervals into frames. Above there is a group of protocols linked to each other. The link manager, for example, is dedicated to the management of logical channels between devices, power management, authentication and QoS. Logical Link Control Adaptation Protocol (L2CAP) is used to hide transmission details from the upper layers.

The intermediate upper layer contains a series of protocols dedicated to various services and functionality. The upper layer contains the applications and the various profiles.

With regard to the physical layer, Bluetooth is a low-power system of emission, with maximum range of 10 m, which operates on the ISM band at 2.4 GHz. This band is divided into 79 channels of 1 MHz each. The modulation used is FSK capable of reaching a speed of 1 Mbps, even if most of the traffic is dedicated to controlling the quality of communication. The channels are used in a homogeneous manner by resorting to distribution of the frequency spectrum, with 1,600 hops per second and a rotation time of 625 ms: the hopping sequence is decided by the master node and all the slave nodes perform this. For Bluetooth operating on the same band of 802.11, but with a higher hop speed, there is more probability that a device that operates with the first system will interfere with the second system than vice versa.

1.4.1.5 Other types of wireless networks

There are other types of wireless networks, which will be explained later.

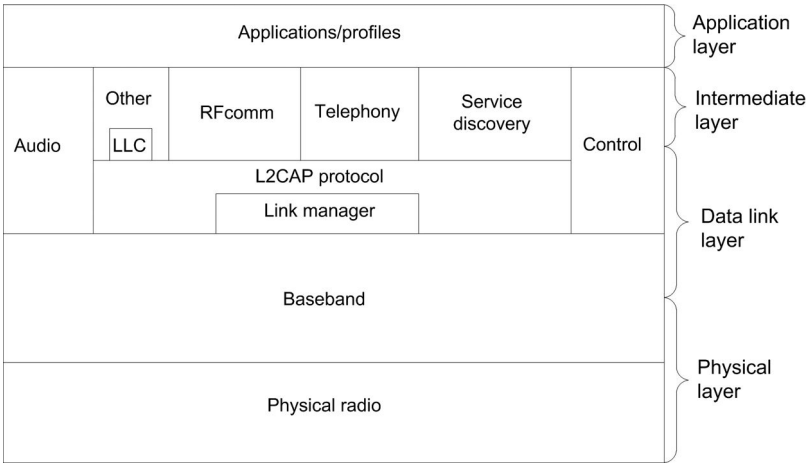


Figure 1.61 Bluetooth architecture in version 802.15.

HIPERLAN

HIPERLAN is the name of a WLAN standard, which describes a series of alternative European solutions to the US IEEE 802.11 standards.

Following on from 2000, European Telecommunications Standards Institute (ETSI), the standards body for communications, given the technological difficulties of innovation of the 802.11, defined a set of standards for transportation of IP, ATM and UMTS backhauling data.

A committee, divided the project into four solutions (TR 101 031 V1.1.1):

1. HIPERLAN;
2. HIPERLAN type 2;
3. HiperACCESS (type 3);
4. HiperLINK (type 4).

But the market was already saturated with Wi-Fi products and few manufacturers invested in a new technology for transportation of immobile data, though even of superior performance. The products on the market with this technology are generally mid-high range, with higher costs than Wi-Fi.

The standards define the physical level and in part the data link level, leaving in some cases freedom for the implementer on the mechanism of contention of the channel.

Some of the basic concepts of the HIPERLAN were then included in the WiMAX standard (802.16). HIPERLAN proposes a local wireless solution for IP transportation, defining a physical level at 5 GHz with FSK modulations and giving manufacturers freedom on the data link level. Its architecture provides the presence of one or several forwarder stations (the aim is to send to its neighbours those frames with different destinations from its address), one or more non-forwarder stations (which are limited to receiving messages) and bridge stations (to connect multiple HIPERLAN/1) networks. Each forwarder and non-forwarder station must update a series of data bases in order to perform routing. The type 1 HIPERLAN standard redefines the physical layer and starts from the data link layer. It specifies the levels of access to the medium (MAC), access to the channel (CAC) and physical level (PHY). This technology also implements a system of QoS at the MAC level and a system of access priority to the channel at CAC level. Access control to the channel is regulated by the protocol Elimination Yield – Non-pre-emptive Priority Multiple Access (EY-NPMA), which allows there to be a relatively low number of collisions.

HIPERLAN/2 also uses the first type and it is being proposed for both the links: point-to-point and point-to-multipoint over short distances, defining both the physical level, always at 5 GHz but also with modulations including OFDM, both the data link level, with a mechanism of contention of the channel in time division multiple access (TDMA) and time division duplexing (TDD). The solution is designed for the transportation of IP data and ATM frames and ensures QoS.

HiperACCESS (TR 102 003 V1.1.1) illustrates a system in OFDM on non-defined frequencies for high-speed data transportation and with low latency for medium distances, in configuration point-to-multipoint applications to sectors using TDMA and frequency division duplex (FDD); the aim is multimedia applications and UMTS infrastructures.

Finally, HiperLINK defines a point-to-point solution on 17 GHz with broadband for long distances (at least 155 Mbps).

WiGig

WiGig is a standard communication technology that operates at a multi-gigabit speed without operating license on the band of 60 GHz, allowing devices to communicate without wires at multi-gigabit speed.

The creation of WiGig was announced on 7 May 2009 by Gigabit Wireless Alliance (also known as WiGig Alliance), an organisation that promotes the adoption of WiGig. The full version 1.0 WiGig was announced in December 2009.

Devices enabled for WiGig tri-band, operating between 2.4, 5 and 60 GHz, are offered data transfer speeds of up to 7 Gbit/s, more than 10 times faster than the 802.11n standard while maintaining compatibility with existing Wi-Fi devices.

WiGig 1.0 includes the following features:

1. It supports data baud rates of up to 7 Gbps (more than 10 times faster than 802.11n).
2. It is compatible with the IEEE 802.11 standard.
3. The physical level allows high performance for WiGig devices, ensuring interoperability and communication at gigabit speeds.
4. It allows robust communication at distances of more than 10 m.

WiGig competes with wirelessHD (WiHD) in certain applications.

In May 2010, WiGig announced the beginning of collaboration with Wi-FiAlliance, in order to cooperate in the development of Wi-Fi products that work at a frequency of 60 GHz.

According to forecasts, 60 GHz Wi-Fi should allow a baud rate 10 times higher than current Wi-Fi connections.

In this way, a good part of the 60 GHz products (if not all) will be able to ensure retro-compatibility with already existing Wi-Fi standards of 2.4 and 5 GHz.

Retro-compatibility will not only be an important element to sustain the device park currently on the market but it will also be necessary to fill the one that seems to be an important drawback of WiGig, represented by the fact that the new standard is able to operate at a maximum distance of 20 m.

WirelessHD

WiHD defines a specification for the interfaces of next-generation wireless digital networks for a high-definition wireless transmission signal for consumer electronics.

The consortium behind the standard concluded specifications in January 2008.

WiHD allows uncompressed digital transmission of marked data, audio and video, essentially performing the equivalent of wireless High Definition Multimedia Interface (HDMI) (WHDI).

It is characterised by the following features:

1. It works on the band from 7 to 60 GHz.
2. It allows uncompressed digital transmission of video, audio and high-definition signals.
3. The specifications were designed and optimised for the connection of wireless screens, reaching a speed rate of 4 Gbit/s per PC and mobile devices. This technology promotes a data transmission rate up to 25 Gbit/s (compared to 10.2 Gbit/s for HDMI 1.3), allowing higher resolutions and better performance.
4. Working at a frequency of 60 GHz usually requires a line of sight between the transmitter and the receiver. The WiHD specification improves this limitation: in point-to-point solution, non-line-of–sight (NLOS) up to 10 m can be reached.

Wireless high-definition interface

Wireless high-definition interface (WHDI) is a new technology of MIMO short-range wireless data transmission that seeks to replace cables from video and televisions and other screens, with a high-definition television (HDTV) wireless connectivity throughout the home.

WiHD 1.0 specification works with transfer rate up to 4 GBps. It is characterised by the following features:

1. Data transfer rates of up to 3 Gbit/s, on a 40 MHz channel, and data baud rate up to 1.5 Gbit/s.
2. A 20 MHz channel of 5 GHz band without licence.
3. The signal reaches beyond 30 m, also passing through the walls.

1.4.2 Switching in the data link layer

A component much used for data link layer switching is represented by the bridge. Such a device, in order to operate correctly, examines the addresses of the data link layer in order to perform routing of the frames. Not having to examine the contents of frames, it can route IPv4 packets (current Internet standards), IPv6 packets (future Internet standards) and so on.

The router (which will be described later), on the contrary, examines the addresses of the packets and routes them on the basis of these.

The bridge is used all the times when there are several LANs, near and far, and where they need to be merged together or, on the contrary, when a single LAN is going to be divided into physical subnetworks in order to divide the load. It is also used when the distances involved are very high; in such case, the LAN is divided into functional sections connected to each other through a bridge. Bridges also improve the reliability of networks as, by analysing critical addresses, they can avoid out-of-control nodes filling the network with useless packets that would saturate the same. Bridges also improve the security of networks that can be programmed to send only certain types of traffic on more vulnerable sections of the network (Figures 1.62 to 1.72).

Bridges tend to operate in a transparent manner, since any computer can be moved from one segment of the LAN to another without the need for any hardware modification.

Bridges, when they have to operate between networks of different formats, must, in most cases, perform data formatting operations that inevitably require computing capacity. Moreover, if they connect networks with different speeds, they must store the data of a high-speed network in an appropriate memory buffer in order to be able to transfer them onto the network with lower speed always risking saturation of the buffer by the high-speed network.

Another problem is represented by the networks that use long frames: in this case, the bridge not being able to break these frames into frame characterised by a smaller length, discards them directly.

Another problem is represented by security since both 802.11 and 802.16 use cryptography on the data link layer while Ethernet does not, and this inevitably reduces the level of security when the traffic is transferred from one type of network to another.

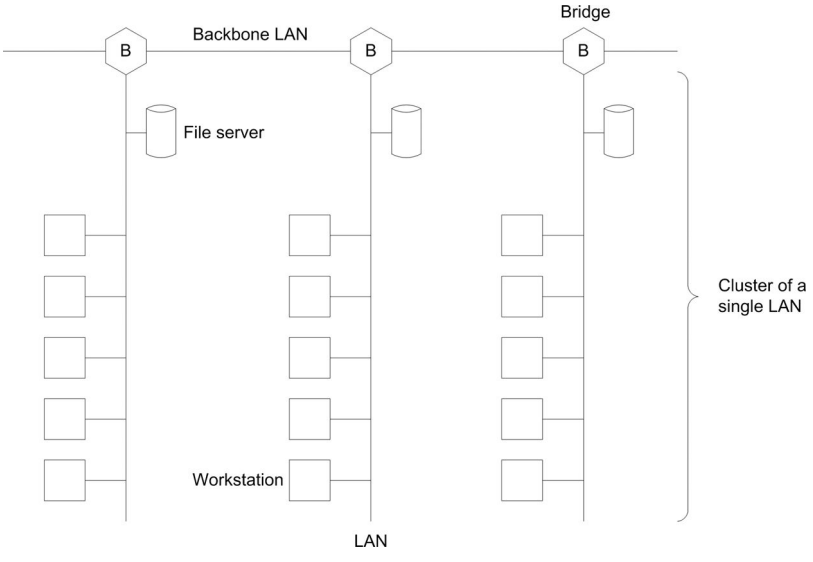


Figure 1.62 Example of connection of multiple LANs via bridge.

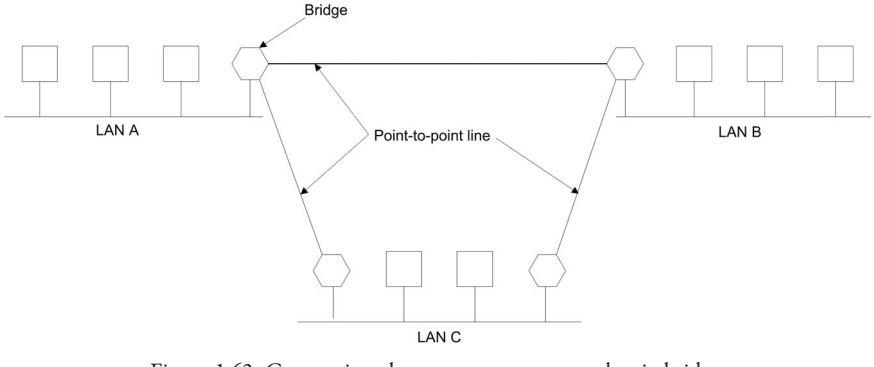


Figure 1.63 Connections between remote networks via bridge.

Another problem is represented by the loss of QoS that occurs when the traffic passes from 802.11 and 802.16 networks (able to manage it) to Ethernet-type networks (which are not able to manage it).

Bridges are used for the connection of remote networks. This is done, for example, by connecting the various bridges in pairs, to the ends of a dedicated point-to-point line (possibly rented from an external communication company).

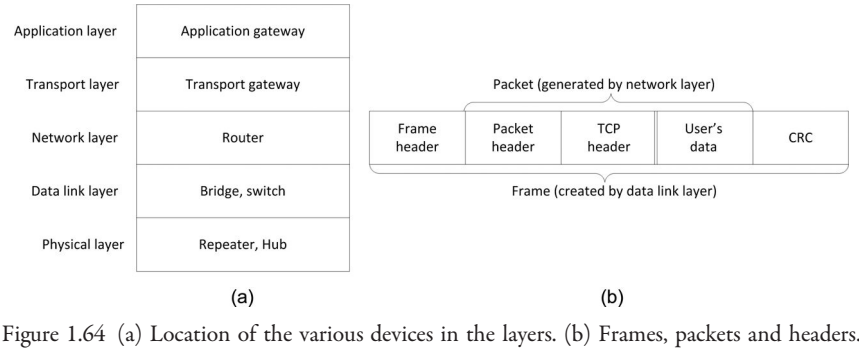


Figure 1.64 (a) Location of the various devices in the layers. (b) Frames, packets and headers.

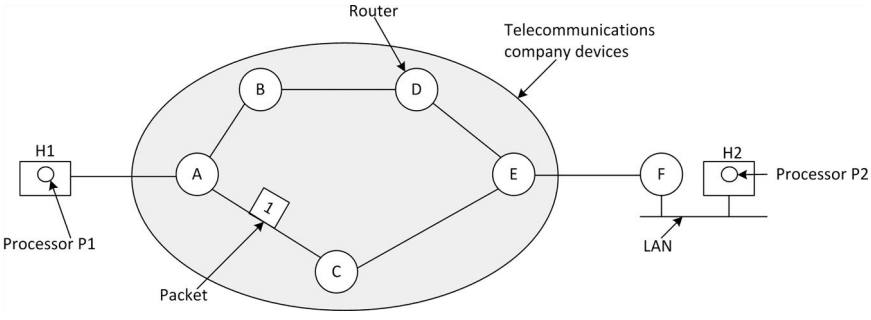


Figure 1.65 (a) Hub connection, (b) bridge and (c) switch.

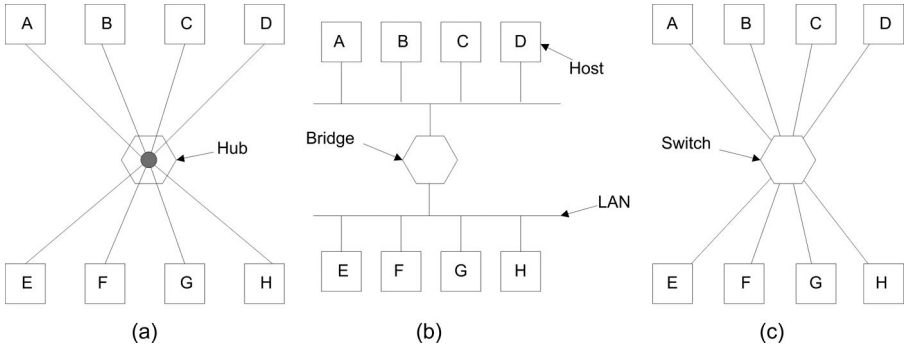
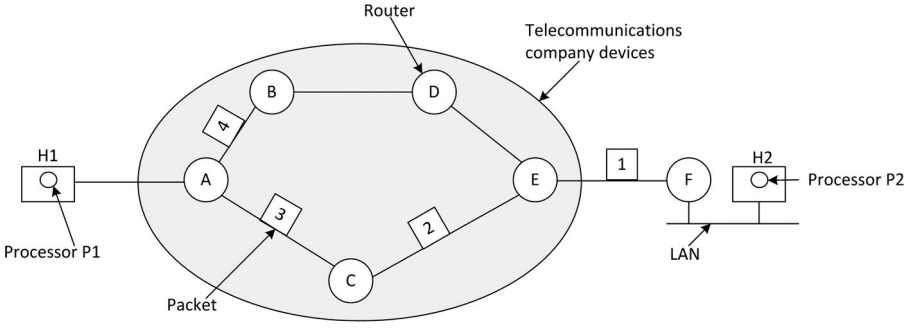


Figure 1.66 Example of architecture in which the layer network operates.



A table

Initial		Later	
A	-	A	-
B	B	B	B
C	C	C	C
D	B	D	B
E	C	E	B
F	C	F	B

Destination line

C table

A	A
B	A
C	-
D	D
E	E
F	E

E table

A	C
B	D
C	C
D	D
E	-
F	F

Figure 1.67 Example of routing in a datagram subnet.

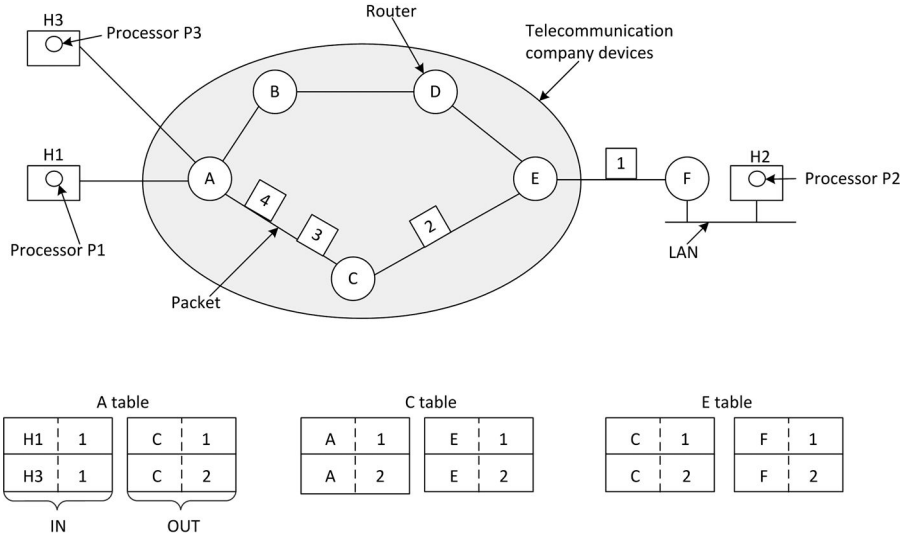


Figure 1.68 Example of routing in a virtual circuit subnet.

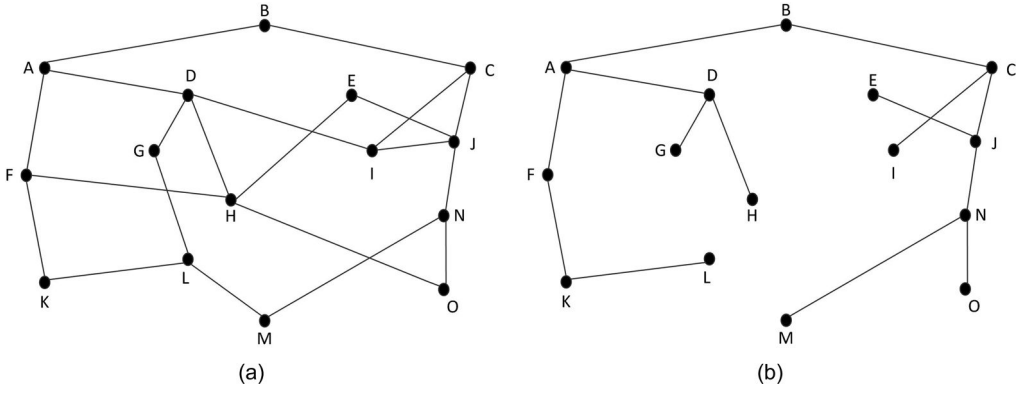


Figure 1.69 (a) Example of subnet and (b) corresponding sink tree.

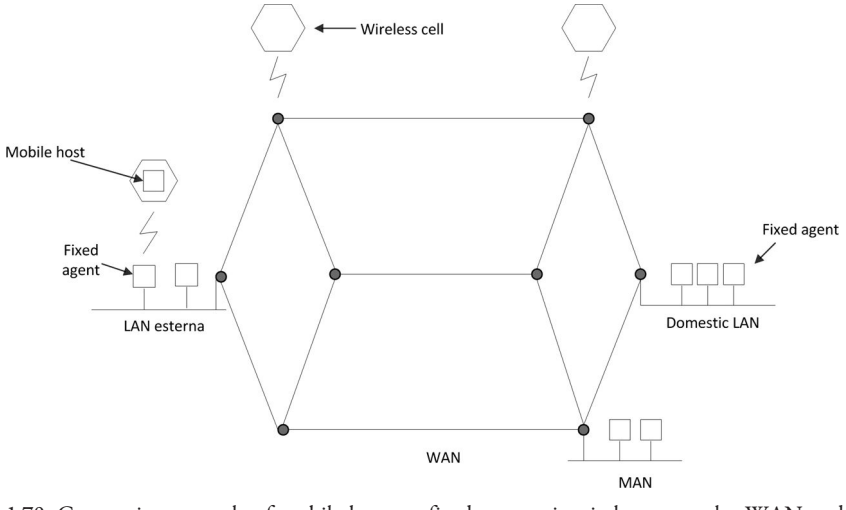


Figure 1.70 Connection example of mobile hosts to fixed agents via wireless networks, WAN and LAN.

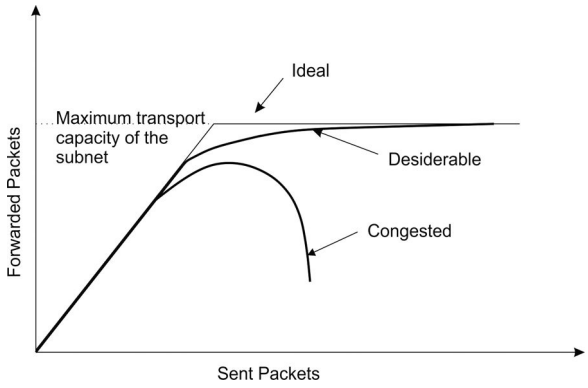


Figure 1.71 Qualitative development of the number of packets forwarded according to the number of packets sent in the ideal case, desirable case and the case relating to congestion.

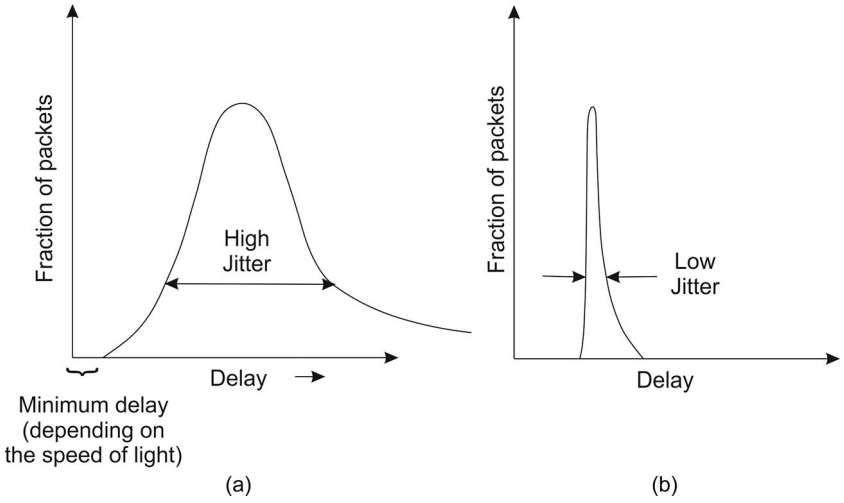


Figure 1.72 Situation of (a) high variability and (b) reduced variability of the delay.

For the switching of packets, various devices are used that operate at various levels of the network layers. They are repeaters, hubs, switches, bridges, routers and gateways. Operating on different layers, they use different parts of information to perform switching. We have seen how information, passing from the upper levels to the lower ones, is encapsulated in data structures typical of each layer.

The repeaters are located in the physical layer and are substantially analogue amplifiers that reason in terms of voltage to be amplified and are connected to two lengths of network cable. Ethernet, for example, supports up to four repeaters by increasing the maximum distance from 500 to 2,500 m.

Hubs are characterised by a number of electrically connected input lines. In this way, frames that reach one of the inputs are transmitted on all the others. If two frames arrive at the same instant, they collide, as is the case with a coaxial cable: in this case, we say that the hub forms a single collision domain. The input lines all operate at the same speed and do not amplify signals. The hub, as well as repeaters, do not analyse the addresses of the input data, instead only perform operations on a physical level.

Rising from the physical layer to the data link layer, we find bridges and switches.

It has already been said that the bridge connects multiple LANs and, when a frame arrives, it analyses the related destination field and, by controlling its internal routing table, sends it on the

correct line. When using a bridge, each line represents a collision domain, unlike the hub where that domain is unique.

Switches operate in a similar manner to bridges, the difference being that the first is used especially to connect individual computers while the second deals with sections of the network. When a computer wants to send data to another computer, the switch should operate actively while the bridge, if the two computers are on the same network, deletes such data. Having to connect every single computer with others, switches are equipped with a greater number of ports with respect to bridges that should connect only sections of network. In the switch, each door represents a single collision domain and for this reason, when this device is used, data are not lost because collisions never occur. The only problem is the arrival of an excessive quantity of data with respect to the capacity of the buffer that can cause saturation of the memory of the latter and loss of data.

The device called router works at the top network layer. When a packet reaches such a device, the header and trailer of the same are deleted and the content of the same is correctly routed on the basis of the header of the same packet. If the packet is IP-type, the header of the same packet is 32 bits, if address type is of IPv4 kind, or 64 bits, if address type is of IPv6 kind. Routers will be described in greater detail in the paragraph relating to the network layer.

Transport gateways are found at the upper level that allows connection between computers that use transport protocols oriented to different types of connections.

At an even higher level application gateways are found, which are able to understand the format and content of transported data and are able to translate messages from one format to another different format.

Often, there is the need to combine groups of machines, in a homogeneous way, on the same LAN but this is not always possible because of their different physical location. This was made possible via software thanks to virtual LANs (VLANs), suitably standardised by committee 802. They are based on appropriate VLAN switches or even on hubs positioned on the outer perimeter. Virtual networks that are created are usually represented with a colour code, assigning to each machine belonging to a well-determined VLAN a very precise colour to distinguish it from other machines and from other VLANs to which the machines belong.

1.5 The network layer

The layer of network takes care of the movement of data packets from source to destination, managing the different passages through the various routers that the path may require. Unlike the data link layer, which deals with moving only the various frames from one extreme to another of the cable, the network layer represents the lowermost layer that takes care of transmission from one point to another. To do this, the network layer must be aware of the topology of the network that must use and choose the optimal routes within it, trying to better manage the communications lines available, avoiding overloads or underuses.

The typical mechanism that is used for the forwarding of packets is the so-called store-and-forward.

In this mechanism, a host that has data to transmit sends it to the closest router through its LAN or via a point-to-point connection of the telecommunications operator. The various packets are stored until they have arrived in full and then checking for correctness is performed. Once they have arrived correctly, they are transmitted to the next router that performs the same operation and so on until the packets reach the final destination.

The network layer provides its services to the transport layer by using the interface between the two layers. The services of this layer are designed to ensure that:

1. the same are not linked to the technology of the routers used;
2. the transport layer does not have access to details of the network layer such as the number and type of router;

3. the network addresses of the transport layer use a consistent structure through the various LANs and WANs.

There are two different orientations concerning the services that the network layer should provide. One orientation requires from this layer the services without a connection (as in the case of the Internet) while the other requires from this layer the services oriented to the connection (as in the case of ATM).

If the service is connectionless, packets are transmitted on the subnet in question individually and routed individually, since a preliminary configuration is not necessary. The packets are in this case called datagrams and the relative subnet is called datagram subnet. If, on the contrary, the circuit is connection-oriented, before starting the connection, a so-called virtual circuit is identified that connects the source router to the destination router and in this case the subnet is called virtual circuit.

In the service without connection, the data to be transmitted are conveniently divided into packets, adequately addressed and forwarded on the same network. The various routers contain appropriate routing tables that vary depending on the status of traffic of the network itself, owing to which the various packets can be forwarded on different paths to reach the final destination where they will arrive not necessarily according to the sending order. The algorithm that organises the routing tables and decides on the forwarding path is called routing algorithm.

In a service with connection, the data are always divided into packets, suitably addressed, but the forwarding path is established before initiating transmission and remains the same throughout the transmission itself, in a manner similar to telephone communications system. Once transmission is completed, the virtual circuit used is released. To route data correctly, each packet is labelled with a number that identifies the virtual circuit on which the same must be forwarded.

Datagram and virtual circuit subnets each have advantages and disadvantages, depending on the determined objectives (Table 1.6).

Table 1.6 Comparison between datagram and virtual circuit subnets.

Problem	Datagram subnet	Virtual circuit subnet
Circuit setting	Not necessary	Necessary
Addressing	Each packet is labelled with the destination address and the source address	Each packet is labelled with the number of virtual circuit on which it is transmitted
State information	The routers do not store information about the state of connections	Each virtual circuit needs space in the tables of the routers to connect
Routing	Each packet is forwarded independently from other	The route is selected when setting up the virtual circuit and all the packets follow that path
Effect of faults on routers	No effect with the exception of packets lost due to the failure	All packets passing through the failed router are terminated
Quality of service	Difficult	Simple if we are able to assign, in advance, sufficient resources to the virtual circuit
Congestion control	Difficult	Simple if we are able to assign, in advance, sufficient resources to the virtual circuit

1.5.1 Routing algorithms

It has already said that the routing algorithm represents that part of software of the network layer that deals with the choice of output line on which to route data. If it is a datagram subnet, such algorithm performs the choice for each packet as the conditions of the subnet can vary with time while if it is a virtual circuits subnet, this choice is performed during establishment of the circuit itself. In the latter case, reference can also be made to session routing.

A distinction is often made between routing, which deals with the choice of the path to be followed, and forwarding, which represents the operation performed upon arrival of the packet.

The routing algorithm should be able to adapt to changes in network topology that can occur over time. The routing algorithm should also be stable and converge towards the balance in the shortest possible time.

Routing algorithms can be divided into two classes: adaptive algorithms and non-adaptive algorithms. Non-adaptive algorithms perform estimates of the traffic and of the topology to make the best choice, and the generic path between two points is calculated in advance and off-line by downloading this information subsequently into routers. Adaptive algorithms vary their estimates in real time, using as information traffic and network topology, varying the forwarding paths continuously, on the basis of information they receive. There are different techniques and algorithms used for routing.

A fundamental axiom of reference is the so-called principle of optimality, which states that if node K is located on the optimal path that connects N with M, then the optimal path that connects K with M follows the same route. A direct consequence of this principle is that the series of optimal routes that connect all of the sources to a certain destination give life to a tree where the root node is represented by the destination node. This tree is known as a sink tree, which is a structure that contains no cycle and each packet is forwarded to it in a finite and limited number of hops.

It is clear that routers can lose their functionality and therefore vary the structure of the tree. In addition, each router can have its own path management policy. The fact remains that the principle of optimality and sink tree structure represent a valid reference model for the assessment of the performance of routing algorithms.

One possible technique is represented by routing based on the shortest path that will not be shown in details for the sake of brevity.

Another technique is based on a static algorithm called flooding, in which each packet is sent on all lines with the exception of the line of origin. This mechanism of course generates a large number of duplication packets that could in theory become infinite, with the risk of congesting the network. For this reason, it is necessary, when using this algorithm, to implement a limitation mechanism. In this sense, there is a variant, called selective flooding, in which packets are sent only on lines that, by and large, are sent in the right direction. The flooding algorithm, given its redundancy, is very useful for applications where a high degree of reliability is required, since, given the large number of duplicate packets that circulate on networks, nearly every single packet will arrive at destination.

Another technique is based on the distance vector that represents a dynamic algorithm that takes into account the instantaneous load of the network. This algorithm provides that each router has a table, or vector, indicating the best known distance for each destination and the line that goes to that destination. These tables are updated thanks to a mechanism for the exchange of information with neighbouring routers. This algorithm has formed the basis of Arpanet and has also found application in the Internet with the name of Routing Information Protocol (RIP). This algorithm, although functional and convergent towards the correct solution, is characterised by response times that can be very long. In particular, it is very reactive to positive news but poorly reactive when the news is negative.

Another dynamic technique is represented by routing based on the state of connections, which represents the replacement, in Arpanet, of routing based on the distance vector. This algorithm is based on the following assumptions:

1. Finding neighbouring routers and relative network addresses.
2. Evaluating the delay or cost of each nearby router.
3. Assembling a packet containing all the information acquired.
4. Sending this packet to all routers.
5. Calculating the shortest path towards other routers.

This algorithm has proven to be very efficient.

It is obvious that the size of routing tables grows proportionally with the size of the network. This growth leads to a high consumption of internal memory of the router, an increase of the processing time for data analysis and a bandwidth consumption for the due exchange of information on the state of the network by routers. To reduce these problems, it is worth considering adoption of a mechanism of hierarchical routing, similar to what is performed in telephone networks. In hierarchical routing, various routers are inserted in appropriate regions where each router knows only the information of the routers that belong to the same region. In this way, the various routers of the network are not obliged to know all the details of the same network. When the sizes of the network are high, use must be made of multilevel splitting, grouping the regions in a cluster, the clusters into areas, the areas into groups and so on.

Often, in practical applications, it is necessary to send messages to all users of a network. In this case, we are discussing about broadcast that can be carried out in different ways. The simplest method is to send a message to every user who in any case wastes bandwidth and forces each user to have a full list of other users. Another method is represented by the flooding algorithm that nevertheless generates too many packets and wastes bandwidth. A further method is represented by routing to multiple destinations (multi-destination routing) in which each packet contains a list of destinations. When the router receives this packet, it checks all the destinations contained therein and sends it on the relevant lines, ensuring that the packets of each line contain only the addresses of that line. Ultimately, the final packet will contain only the address of the last user or the last users of this line. Another method is based on the spanning tree. In this case, each router is able to know which of its lines belong to the spanning tree, and copy the broadcast packet on each such line except for the input line. This method uses the band in an optimal manner and produces the fewest number of packets. To make this method operational, each router must know the spanning tree, which is simple to implement in routing based on the status of connections but difficult to implement in routing based on the distance vector.

In some cases, different processes must be operational on different groups. In this case, the messages of a component of a group must be sent to all members of the group. This mode of transmission is called multicast, and requires a system that creates and destroys dynamically the various groups, appropriately updating the various routers. In this mode, each router creates its own spanning tree that covers all other routers. Different techniques can be used to achieve this objective but will not be shown for the sake of brevity.

1.5.1.1 Routing on mobile networks

On mobile networks, due to their intrinsic characteristics, the concepts related to fixed networks are not always applicable. On mobile networks, there are mobile hosts that can be connected from any point on the network and that need to be connected from one or several fixed agents.

In such networks, the problem of routing of packets transmitted by a mobile host to a fixed agent of the network is particularly sensitive.

Hosts that never move from their position are called fixed hosts and are connected to the network through permanent cables that can be either copper or optical fibre. Fixed hosts that move every so

often and in any case use the fixed network for their communications are called migratory hosts. Hosts that process while they move and need to stay connected during their movements are called roaming hosts. Migratory and roaming hosts are named, in short, mobile hosts.

With regard to mobile hosts, it is assumed that the latter belong to a certain LAN, called domestic rental, which remains unchangeable over time and for this reason have a permanent domestic address that is used to find the relevant domestic location. To correctly route data from the mobile host to the domestic rental, the latter must firstly be located.

The network is divided into relatively reduced geographical extension areas, represented by LAN or wireless networks. Each of these areas is equipped with one or more external agents that are used to have a section of mobile hosts that use these areas. Each of these areas is also equipped with a fixed agent that keeps track of the mobile hosts that have a fixed location in such areas but that are not found at that moment in their areas.

If a new mobile host enters an area, it must register with the relevant external agent. The procedure is divided according to the following points:

1. Each external agent transmits, with periodicity and in broadcast mode, a packet that declares its existence and its relevant address. When a mobile host enters a new area, it can wait for one of these messages or transmit, after a certain period of time in which it has not received any message, a message to request the existence of a fixed agent in the same area.
2. The mobile host performs its own registration with the external agent, communicating the address of the domestic rental, the current data link address and the security information.
3. The external agent notifies the mobile host fixed agent of the existence of the latter in its remit area, its own network address and security information in order to authenticate the mobile host with the relevant fixed agent.
4. The fixed agent controls security information and, in the case of a positive outcome, authorises the external agent to be allowed to enter the mobile host network.
5. The external agent, having received confirmation from the fixed agent, communicates to the mobile host the positive outcome of the network registration.

When the mobile host leaves a certain area or is turned off, it should communicate this information to the relevant fixed agent. When an entity needs to transmit data to a mobile host, it sends it to the relative domestic rental that knows the location of the same mobile host. Its fixed agent performs the following steps:

1. Encapsulation of the necessary payload in an appropriate packet that is sent to the external agent where the mobile host is found via a mechanism called tunnelling. The external agent extracts the original packet from the sending capsule and forwards it to the mobile host.
2. Direct communication to the sending transmitter of subsequent data to the external agent of competence.

What has been addressed so far is valid when the hosts are mobile and the routers are fixed. There may however be some situations in which routers are also mobile as in the case of military vehicles stationed in battle areas without landlines, navigating ships, rescue teams and a group of portable computers located in an area that does not have fixed coverage. In this case, each node consists of a host and a router and networks composed of such nodes are called ad hoc networks or mobile ad hoc networks (MANET). Techniques used in fixed networks and illustrated previously, where paths always remain the same unless there is a router failure, are not, unfortunately, valid with this type of network. In ad hoc networks, given the mobility of nodes, the paths vary continuously and a path available at a given moment may not be available the next moment. One algorithm greatly used is ad hoc on-demand distance vector (AODV). This algorithm determines a path towards a certain destination only when a given subject requests it, taking into account the bandwidth and reduced life of the batteries that power the mobile nodes.

A typical operation that is performed when a source node wants to communicate with a destination node is the so-called route discovery. To do this, there are various types of algorithms that require the source node to send to the nodes under its coverage a specific message that these nodes in turn send to the nodes under their coverage. Once the propagation of such messages is completed, tables indicating the connections that each node has with the others are sent back, leaving the source node the possibility of obtaining the existence or not of a path that exists between it and the destination node. It is clear that this operation consumes a lot of bandwidth and many energy supply resources for which great attention is paid to this operation. Route discovery algorithms can be divided into proactive and reactive. Proactive algorithms perform with periodicity the operation of route discovery, keeping constantly updated internal routing tables while reactive algorithms perform the operation of route discovery any time that the need manifests itself. It is clear that if communication needs are high, it is preferable to use a proactive algorithm whereas if communication needs are reduced, it is preferable to use a reactive algorithm. There is also a class of algorithms, called hybrids, which use certain features of proactive algorithms and some characteristics of reactive algorithms.

Another type of network of relatively recent popularity is peer-to-peer. In these networks, some subjects are connected with each other through a network, typically the Internet, in order to share resources. Such systems are distributed, symmetrical and do not use any central control or hierarchy. In such networks, as there is no central repository, there is the problem of how to find out the information held by each node belonging to the network and there are various techniques that are not illustrated for reasons of space.

1.5.2 Congestion control algorithms

Networks are sized to carry a number of packets per second: if this number is exceeded, performance of the same inevitably degrades, becoming congested.

If the number of packets is less than the maximum allowed, all the packets are correctly forwarded to the destination. If this number increases greatly, routers are unable to perform their work properly and begin to lose some of those packets. If this number increases excessively, routers completely degrade their performance, and almost no packet arrives at destination.

Routers can become congested for various reasons. The main reason for this is due to the excessive increase of packets on input lines that require the same output line: in this case, packets are queued and temporarily deposited in the internal memory of the router. If the number of packets is excessive in relation to the actual capacity of the router's memory, some packets are lost. In most cases, it is not cost effective to increase the capacity of memory since, if packets are not forwarded with sufficient speed compared to those that are incoming, packets accumulate in the memory and expire, pushing the source to send other equal packets that continue to accumulate in the router's memory, obtaining a negative effect. In addition, all the packets are then forwarded to the destination, greatly increasing the load on the network.

Other reasons for congestion are represented by the low processor speed of routers or by the limited bandwidth available on lines. In practice, there may be congestion all the times that there is a less than optimal adaptation of the various devices or systems that make up a network.

It is important to stress the difference that exists between congestion control and flow control. Congestion control is responsible for ensuring that a given network is able to support a given traffic without clogging and is an issue for the entire network as a whole. Flow control, on the contrary, is responsible for ensuring normal point-to-point traffic between a transmitting unit and a receiver unit, often requiring a reaction response by the receiver that communicates to the transmitter correct performance of the actions of receipt.

Often the two concepts mentioned above are confused since the algorithms of congestion control, in the event of such an event, send to transmitters the reduction request messages of packet sending

speed. In this case, transmitters receive a reduction message that can originate either from the congestion control algorithms or from the receiver.

Solutions that can be adopted can be both open loop and closed loop. Open-loop solutions consist of a good network design, in such a way as to avoid the occurrence of congestion problems: once the network has been enabled, no checking is performed and it is hoped that the above problem does not occur. Closed cycle solutions use the concept of feedback that applies in three parts: network control for the detection of congestion, transmission of this information to the devices of a network able to carry out corrective manoeuvres and execution by the devices used for corrective actions.

Network congestion can be controlled using different metrics such as the percentage of packets discarded as a result of the exhaustion of the memory buffer, the average length of queues, the number of expired packets that is retransmitted and the average delay of packets. The increase of the values listed above can be a sign that congestion is occurring.

Once the information on congestion has been gathered, the latter must be transferred and this can be done by the congested individual routers vis-à-vis the sending sources of packets, even if this operation tends to aggravate the status of the network. To avoid this, a field reserved for routers can be added into each packet that is filled in all outgoing packets to warn neighbouring routers of the state of congestion. Another way of transferring congestion information consists of the transmission by hosts and routers of signalling packets intended to request information on the state of congestion and properly routing new packets by sending them on uncongested paths.

The mechanism of feedback must not be too fast, in order to avoid excessive oscillations of the entire network, yet not too slow to avoid excessive slowing down of the operation of the network as a whole. In an attempt to solve congestion, the capacity of the network can be increased, where possible, or load of the same can be reduced.

With regard to open systems, various criteria in the data link network and transport layers are used. In terms of the data link layer, the following are used: retransmission, caching out of sequence, acknowledgement and flow control. With regard to the network layer, the following are used: choice of virtual and datagrams circuits on the network, queuing, packets service, elimination of packets, routing and management of the useful life of the packet. In terms of the transport layer, the following are used: retransmission, caching out of sequence, acknowledgement, flow control and time-out determination.

In relation to closed systems, a technique that is widely used is called admission control that prevents the setting of any virtual circuit from the moment when congestion is declared until it is resolved. Another technique used instead admits the creation of new virtual circuits but only along non-congested pathways. Another technique consists of negotiation between the host and the network during the setting up of a new virtual circuit. When congestion is at very high levels, techniques of greater impact are used such as load shedding that consists of the brutal elimination of packets by routers. This technique, in order not to generate high levels of disruption, may require the cooperation of transmitters, requiring them to mark packets according to a priority level, allowing routers to discard only low-priority packets to maintain a minimum level of service even in the presence of significant congestion.

Since congestion is easier to manage when it first occurs with respect to when the entire network is jammed, there is a suitable technique called random early detection (RED) that allows the router to check the average length of the queue of their memory and begin to discard packets in the case when this average exceeds an alarm threshold, which prevents total congestion of the network.

In audio and video applications, delay is not as important in relation to retransmission (provided it does not exceed certain limits) as constancy of the latter once transmission is activated. Variation in the arrival time of the packet is called jitter. Upon arrival, a high jitter generates a sound or a video of variable quality while a low jitter generates quality sounds or video.

Jitter can be monitored by calculating the transit time expected with each hop along the source-destination path. Every time a packet reaches a router, the latter controls whether it is early or delayed

in relation to the anticipated time. If it is early, the router stores it and sends it at the right time and if it is late, the router sends it as soon as possible. In this way, jitter is reduced.

To reduce jitter in audio or video applications, a part of the data can be previously downloaded and playback started, in such a way that any delays in the subsequent packets do not affect the quality of data reproduced at destination as the data are retrieved from the receiver buffer.

1.5.3 Quality of service

In addition to avoiding congestion on the network, it is very important in the transmission of multimedia content to ensure a certain QoS.

It has already been seen that in connection-oriented networks, all packets that belong to a certain flow follow the same path, which does not happen in networks without a connection. There are essentially four main parameters that are used to characterise the quality of a stream of packets: reliability, delay, jitter and bandwidth. Each application is characterised by different values of the QoS (Table 1.7).

Since the first four applications (email, file transfer, web access and remote login) must be characterised by high reliability, no bits must be affected by errors. To accomplish this, checks are carried out at destination and, in the event of an error, confirmation of correct receipt is not sent, an event that causes the transmitter to resend the packet that incorrectly reached the destination.

On the contrary, the last four applications (audio on demand, video on demand, telephony and video conferencing) can tolerate errors and, in this sense, no specific control is carried out.

Email applications, file transfer, audio on demand and video on demand are not particularly sensitive to delay as if all the packets undergo the same temporal slowdown, there are no particular problems. Real-time applications, such as telephony and video conferencing, are instead particularly sensitive to delay.

Email, file transfer and web access are not particularly sensitive to jitter. But in general, remote login, and in particular audio and video on demand, telephony and video conferencing are, however, sensitive to jitter.

Electronic mail, remote login and telephony do not require much bandwidth; video in all its forms requires a lot of it while the remaining applications require it in medium quantities.

In order to meet the demands of QoS, ATM networks bring the streams together into four categories: constant speed (as, e.g., telephony), variable speed in real time (as, e.g., compressed video conferencing), variable speed not in real time (as, e.g., the display of a video stream on the Internet) and available speed (as, e.g., file transfer).

A possible solution to ensure high quality is oversizing, that is in the use of capacious routers with large buffers and high bandwidth. This solution is however expensive. The telephone system, for

Table 1.7 Main parameters of QoS for various applications.

Kind of application	Reliability	Delay	Jitter	Banda
Email	High	Low	Low	Reduced
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Reduced
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	Wide
Telephony	Low	High	High	Reduced
Videoconferencing	Low	High	High	Wide

example, is usually oversized in such a way as to be able to make telephone conversations at any time required.

Streams can be stored in buffers before being forwarded. This solution does not influence reliability and bandwidth, eliminates jitter but increases delay.

Since the majority of traffic is irregular, causing congestion and degrading the QoS, a technique called traffic shaping is often used that regulates traffic from the server rather than from the client. In this sense, at the time of transmission, the user and the network (i.e. the telecommunications operator) agree on the model of traffic sustainable by the network at that moment. This operation is called the service-level agreement. If the user complies with the agreement on the traffic model agreed, the operator undertakes to deliver according to this agreement. This technique is very useful in data transmissions in real time such as audio and video. Performance of transmission according to the model agreed is called traffic policing. This supervision is more manageable on virtual circuit networks with respect to datagrams networks.

The regulation of traffic represents a fundamental point in order to achieve a good QoS. An essential prerequisite for achieving this is that all the packets follow possibly the same path and for this reason a concept similar to virtual circuit must be used. This implies that, once the path is established, it must be possible to reserve the resources that are able to ensure the required capacity. The resources that are basically possible to reserve are bandwidth, buffer space and microprocessor cycles.

When a request is made for resource reservation to a router, the latter must decide on the basis of its possibilities and on already existing reservations whether to accept this or not. To allow routers to make a proper assessment, it is necessary to provide accurate information of the flow, which is called flow specifications. The flow specification is transmitted along the chosen path and all routers check whether they are able to satisfy it, possibly reducing the flow.

A simplified approach consists of defining the QoS through classes, offering differentiated services according to the class choice beforehand.

1.5.4 Connection between networks

In the real world, in most cases, different networks are interconnected (LAN, MAN and WAN) that use different protocols to form what is called an Internet.

When packets that are sent from a host have to pass through different networks in order to reach the final destination, problems can occur due to interfaces that connect the different networks. For example, if packets originate from a connection-oriented network and must go to a network without a connection, in most cases the data have to be put in order again, functionality that the transmitter does not intend to do and that the receiver may not be able to perform. In most cases, protocol conversions and address conversions are required. In addition, the packet size usually varies from one network to another, resulting in problems and additional processing charges. Similarly, the QoS suffers from the presence of different networks. This is the same situation for the control of errors, flow and congestion.

We have seen that the connection of multiple networks can take place at different levels of the ISO/OSI layer, using repeaters and hubs at the physical layer, bridges and switches at the data link layer, routers at the network layer and a gateway in the layers of transport and application.

In general, management of the connection between different networks is not an easy problem to solve. However, there is a case of easy solution, represented by two hosts connected on the same network type (such as Ethernet) connected through a network of a different type (e.g. a WAN). In this case, a technique called tunnelling can be used, wherein the network between the two networks is connected with the latter by multi-protocol routers. Multi-protocol routers package data from hosts originating from the related networks in a manageable format from the intermediate network, creating a true specific usage tunnel within the latter.

The routing that is carried out when connecting multiple networks is similar to the routing that is carried out within a single network, with the benefit of additional functionality with which to better manage the different technologies used. An interior gateway protocol is usually used within each network, while for connection between networks an exterior gateway protocol is used. Generally, a data packet that has its origin in a certain LAN is addressed to a local multi-protocol router where, within the network layer, the relevant routing tables decide to which multi-protocol routing it should be sent. If the packet remains within the same network, the protocols specific to the same network are used. If the packet has to travel on a network characterised by a different technology, then the same is suitably encapsulated for use of a tunnelling mechanism. The latter process is optionally repeated several times until the packet reaches the final destination.

Each type of network places limits on the maximum size of packets that can depend on various factors, including: the hardware, the operating system, the protocols, need for compliance with international standards, the reduction of retransmissions due to errors, reduction of time of use of a channel, etc. Problems are typically caused when a large packet has to be submitted on a network characterised by smaller size packets. In this case, if the router is unable to avoid, in forwarding, this network, the larger size packets must be divided into packets of smaller sizes, called fragments. This operation is carried out by multi-protocol routers that operate as gateways.

1.5.5 The layer network on the Internet

With regard to the network layer on the Internet, the ten basic principles that have directed the development of its architecture and that have allowed the success it has had should be remembered. These are: functionality, simplicity, clarity of choices, modularity, heterogeneity, minimising the number of options and static parameters, quality of the project, stringency of the transmission and tolerance in receipt, scalability and optimisation of the price/performance ratio.

In the network layer, the Internet network is composed of a set of subnets called autonomous systems (AS) connected to each other, as there is no reference structure but many major backbones composed of broadband lines and fast routers. To these backbones are connected intermediate-level networks (or regional networks) to which in turn are connected various LAN of different users subjects.

The main protocol is represented by the IP, which was specially designed to allow communication between interconnected networks of different technology, allowing various datagrams to be properly transported from a source to a destination without dealing with the positioning of the various hosts. To do this, the protocol takes data flows and divides them into datagrams, the size of which cannot exceed 64 kb but that, in practice, does not exceed 1.5 kb (which coincides with the Ethernet frame size). These datagrams are sent through the Internet, possibly breaking them down into smaller sized units if necessary. When these units arrive at their destination, they are reassembled from the original datagrams and passed to the upper transport layer for the required sending to the requesting application.

An IP datagram is composed of a header and a text part. The header is composed of a fixed part of 20 bytes and an optional part of variable length. The format of the header of the IP datagrams (in the IPv4 version) is shown in Figure 1.73.

The “version” field indicates the version of the protocol used for the datagram. A transition from IPv4 to IPv6 is currently taking place.

The “Internet Header Length (IHL)” field indicates the length of the header in words of 32 bits, which can vary from 5 to 15 for which the maximum size of the header cannot exceed 40 bytes.

The “type of service” field is used to catalogue the various classes of service and is composed of 6 bits.

The “total length” field contains the sizes of the entire datagram, corresponding to 65,535 bytes, considering both the header and the data.

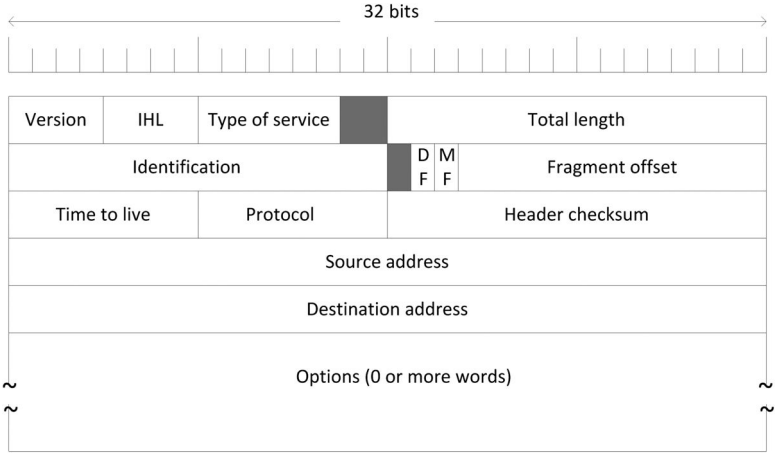


Figure 1.73 Format of the header of a datagram in IPv4 version.

The "identification" field is used by the recipient to understand to which datagram the incoming data fragment belongs.

The "DF" (don't fragment) field is used to request routers not to fragment the datagram because the destination is not able to recompose it.

The "MF" (more fragments) field is always set to 1 for all the fragments except for the last that is set to 0.

The "fragment offset" field is used to indicate the position of the fragment in the datagram. All the fragments of a datagram, except the last, must be multiples of 8 bytes that represent the size of the basic fragment: as 13 bits are available, the datagram can be broken into a maximum of 8,192 fragments, corresponding to 65,536 bytes (1 byte more than the total length).

The "time to live" field is used to limit the life of the packet. This is a counter that is decremented in seconds even if, in practice, it is decremented at every hope of a unit. The maximum value that it can reach is equal to 255.

The "protocol" field is used to indicate which process (typically TCP among others) in the transport layer is waiting for that given.

The "header checksum" field is used to check the header and is an aid for detecting errors generated by faulty locations of memory of the routers. It must be recalculated at every jump because the least that can happen every time is a change to the "time to live" field.

The "source address" field is used to indicate the network number.

The "destination address" field is used to indicate the host number.

The "options" field was intended as a possibility, in later versions, for the insertion of additional data and/or functionality. The following possible options were considered: "security" that specifies the level of secrecy of the datagram, "routing strictly defined by the origin" that defines the full path to be followed, "routing loosely defined by the origin" that lists the routers that should not be missed, "records the path" that requires each router to add its IP address and "time marking" that requires each router to add the address and time.

Each host and each router are defined by an IP address that identifies it to the network, since it is not possible to assign the same address to two different devices. An IP address is 32 bits long and is used both in the source address field and the destination address field. The IP address, rather than the device, is linked to the corresponding network card and as such if a host is equipped with two or more network cards, it will be characterised by the same number of IP addresses.

For a long time, IP addresses were divided into classes, as shown in Figure 1.74, even if this division is no longer in use.

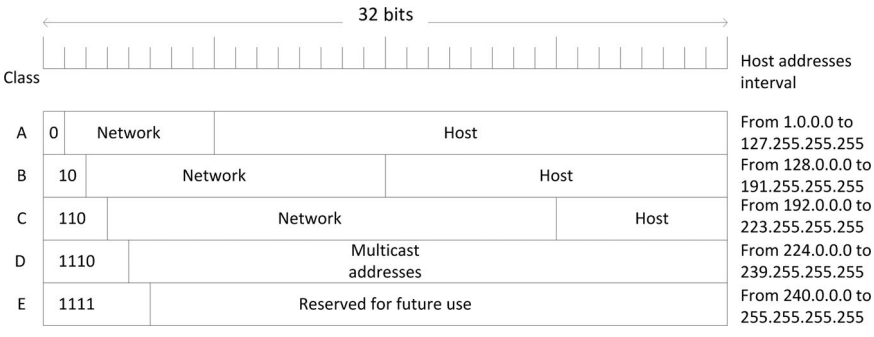


Figure 1.74 Division of IP addresses

Classes A, B, C and D support up to 128 networks, each composed of 16 million hosts, or 16,384 networks of 64,000 hosts or 2 million 256 host networks. The multicast service, that is the transmission of datagrams to multiple hosts, is additionally supported. Since the number of networks connected to the Internet is extremely high, in order to avoid conflicts, the addresses of networks are managed by a non-profit-making organisation called Internet Corporation for Assigned Names and Numbers (ICANN). ICANN, in turn, has given management of certain groups of IP addresses to recognised regional authorities.

The network addresses, as stated, are represented by 32-bit numbers, using commonly dotted decimal notation. In this sense, the 32-bit number is divided into 4 bytes, each of which represents a number ranging between 0 and 255. Using this notation, the lowest IP number is represented by 0.0.0.0 while the highest number is represented by 255.255.255.255.

The IP address 0.0.0.0 is used by hosts at the time of departure. IP addresses that have 0 as network number refer to the current network: using this mode, hosts can communicate on the current network without being aware of the number.

The address characterised by all 1s allows transmission broadcast on the local network, usually of type LAN. Addresses characterised by a certain number of network and by all 1s in the host field allow computers to send broadcast packets to remote LANs connected to the Internet. Addresses written in the form 127.aa.bb.cc are used for loopback tests and the relevant packets are not transmitted over a network, but used within the computer as if they were incoming packets (Figures 1.75 and 1.76).

All the hosts of a certain network must be characterised by one and the same network number. This mode can cause a problem if the number of hosts grows dramatically, exhausting all available IP addresses.

To avoid this, the network is divided internally into multiple subnets, in which a main router is connected to an ISP or to a regional network and the same router directs traffic to the specific subnets.

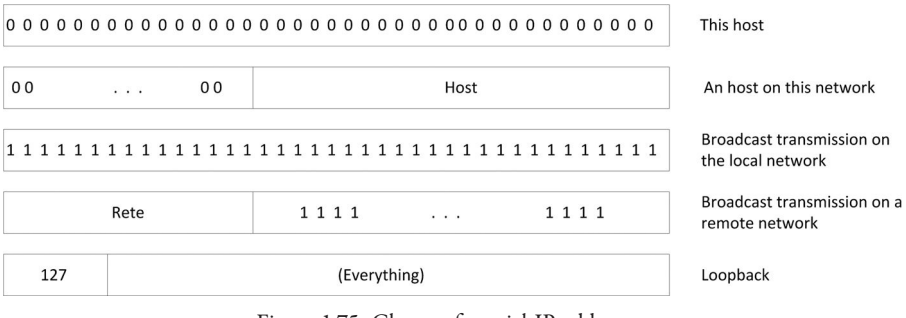


Figure 1.75 Classes of special IP addresses.

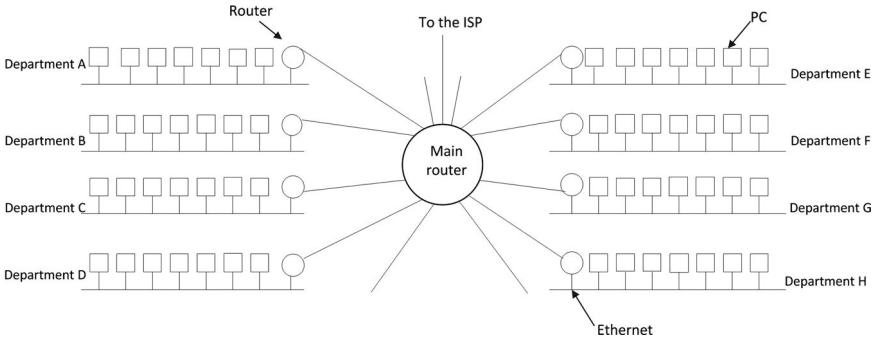


Figure 1.76 Example of division of a main network into subnets.

When a packet arrives at the router, the same could store within it all the addresses of the various hosts requesting a routing table that can reach considerable dimensions. To avoid this, instead of using a class B address with 14 bits for the network number and 16 bits for the host number, some bits of the host number are used to generate a subnet number. To enable this, the main router needs an appropriate subnet mask that states the dividing point between the network number and host number. This situation is shown in Figure 1.77.

The figure shows how the network mask is 255.255.252.0. From an external point of view, division into subnets is not visible and therefore the creation of a new subnet does not require prior authorisation by ICANN.

When the router has to find the requested address, it performs an AND logic Boolean operation between the arrival address and the subnet mask and goes to look for the result obtained within its routing tables.

The exhaustion of IP addresses is, currently, a significant problem. In fact, initially, there were more than 2 billion addresses, most of which were wasted with division into classes, especially by classes of type B. In fact, a class A network, which is characterised by 16 million addresses, is usually too large for most of the stakeholders while a class C network, characterised by 256 addresses, is too small: for this reason, class B networks, characterised by 65,536 addresses, represent the optimal solution. In most cases, not all the 65,536 addresses of class B networks are used and most are wasted. The problem could be solved if 10 bits instead of 8 were assigned to class C networks, for the identification of hosts connected on the same that could reach the maximum number of 1,022 addresses. In this way, the majority of stakeholders would have chosen class C networks, without problems of space as the addresses of this type would have been half a million instead of only 16,384 of class B networks.

To ensure the Internet as a more comprehensive approach in terms of addresses available, a solution called classless inter-domain routing (CIDR) was designed that involves assigning the remaining IP addresses to blocks of variable size, irrespective of the classes. But such a solution requires greater work by routers to operate correct routing.

To solve the problem of the exhaustion of IP addresses, different solutions have been devised. For example, an ISP, which has been awarded a certain number of IP addresses, assigns the latter

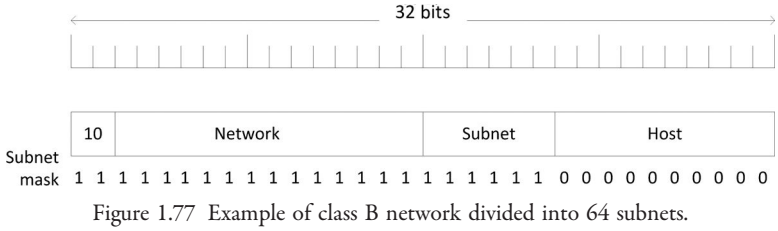


Figure 1.77 Example of class B network divided into 64 subnets.

dynamically, as users connect. When users have completed the connection, the relevant IP addresses are issued and are available for new users who connect.

However, this solution is not passable by large companies or by ISPs that provide an ADSL service and that have computers connected constantly to the Internet.

A solution is represented by IPv6 that guarantees 128-bit routing and that will be discussed later.

The practical solution to manage the address exhaustion of IPv4 is called network address translation (NAT). This involves assigning to each entity concerned an IP address for transmission over the Internet. Within the system, each computer is equipped with its own IP address required for traffic on the internal network. All the times that an internal packet must exit the internal network to enter the Internet, its address is translated. This operation is shown in Figure 1.78.

Within the system, each computer has an address of type 10.a.b.c that is translated into the address that is assigned to the system each time when the communication takes place outwards on the Internet. In most cases, the NAT device works in conjunction with a communications security device called firewall (which is further discussed later) that controls all inbound and outbound traffic.

The problem arises not so much in data transmission as in receipt, when a packet arrives that must be redirected to a computer within the system. To avoid this, since most IP packets transport useful loads of the TCP or UDP type (which will be discussed later) and since these loads contain an indication of the source port and destination port, represented by long 16-bit binary numbers, these parameters are used for NAT addressing. If a process must enable a TCP connection with a remote host, it takes a TCP port that is not used on its computer, called source port, and uses it to receive input data. This process also identifies a destination port for outgoing data. Ports that are typically used are characterised by variable numbering ranging between 0 and 1,023. For example, port 80 is used by Web servers. In this way, when a packet directed outwards reaches the NAT device, its 10.a.b.c. address is replaced by the IP address of the system and the source field is replaced by an index that points to a translation table consisting of 65,536 elements. This table contains the internal IP address and original source port. Similarly, when a packet originating from the Internet reaches the NAT device, the latter extracts the source port of the TCP header that is used to access the routing table. From the routing table, the internal IP address is recovered, in the format 10.a.b.c. and original TCP source port. The IP checksum is also recalculated, which is inserted in the packet directed at the internal computer.

This system is also used by ISPs that provide ADSL services, being able to use only one IP Internet address to serve all users connected to the ISP itself through the various ADSL lines.

It is clear that NAT is a buffer solution to the shortage of IP addresses for various reasons. The first reason is that it violates the hierarchical IP model, which provides that every device connected to the

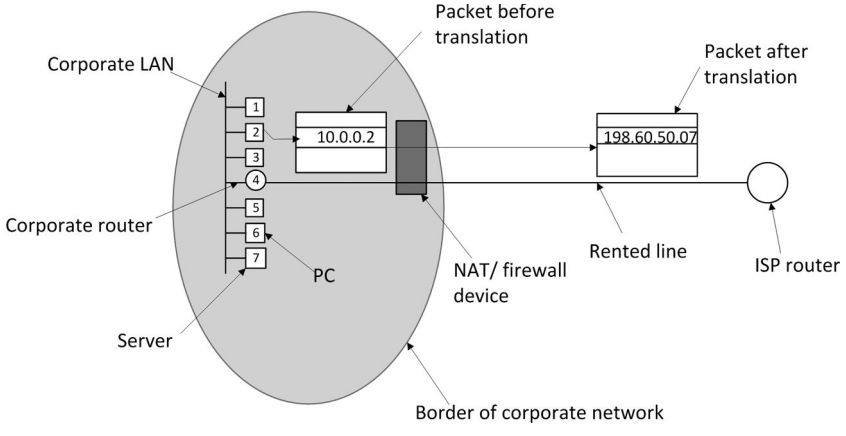


Figure 1.78 Example of operation of a system based on NAT.

Internet must be identified by a unique IP number while with NAT numerous devices can refer to the same IP address.

The second reason is that NAT transforms the Internet as a network without network connection into a network oriented to connections where the NAT device must store within it all the information pertaining to any connection that passes through it. If the device itself malfunctions, all mapping is lost and the same occurs for TCP connections. In the absence of NAT, any failure of a router has no effect on packets that, if after a certain period of timeout do not receive confirmation of receipt, are retransmitted. From this point of view, NAT makes the Internet vulnerable as a circuit-switched network.

The third reason is that NAT violates the stratification of the protocols of the ISO/OSI model, which requires that the generic layer n should not make any assumptions about what the layer n + 1 has entered in the necessary payload field, making the layers independent of each other.

The fourth reason is that on the Internet, TCP and UDP are not exclusively used, making the mechanism of redirection based on source and destination port futile.

The fifth reason is that certain applications embed IP addresses in the body of the text and the recipient extracts these addresses and uses them: as NAT cannot know anything about these addresses, the redirection system fails. A typical case is represented by File Transfer Protocol (FTP) applications that do not work properly in the presence of NAT devices. The same applies to the telephone protocol H. 323 (which will be illustrated later).

The sixth reason is represented by the limitation of the source port field (16 bits that allow representation to the maximum 65,536 addresses).

For all these reasons, it is preferable to use a system of extended addressing, such as IPv6, rather than resorting to temporary buffer solutions such as NAT.

In addition to IP, which is used strictly for data transfer, the layer network is equipped with other protocols useful for the Internet such as Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP), which are illustrated later.

ICMP is used for the Internet testing returning appropriate messages such as:

1. "destination unreachable" that is used when a subnet or a router is unable to find the final destination or when a packet marked with the DF bit cannot reach the destination as it has to go through a network characterised by smaller packets;
2. "time exceeded" that is used when a packet is discarded since its counter has reached zero;
3. "parameter problem" that is used to indicate an error in the header field;
4. "source quench" that is used to slow the hosts that transmit too many packets, avoiding congestion;
5. "redirect" that is used when a router realises that a packet has been routed incorrectly;
6. "echo" that is used to find out if the recipient is on the network and is active. Where positive, the latter responds with a message "echo reply";
7. "timestamp request" that works in a manner similar to "echo" with the difference that the date and time of the response are returned in the response message, type "timestamp reply".

In addition to the above messages, there are many others that are not reported for reasons of space.

ARP is used to identify an IP address on the network. In fact, IP addresses (32 bits) are a prerogative of the network level but this level is based on the data link layer, whose primary standard is Ethernet that uses 48-bit addresses and knows nothing of what happens in the upper layer. ARP works as follows: the host that must communicate with a certain recipient sends a broadcast packet where all the machines connected are asked who the owner of the IP address is with which the host intends to communicate. Transmission reaches all the machines connected and each of these controls its own IP address. Only the machine that owns the IP address responds via its Ethernet address allowing the sender to be able to associate with the requested IP address an Ethernet address with which communication can be commenced.

ARP is also used by computers upon ignition to transmit association of its IP address with its own Ethernet address to all the computers connected: in the event where the computer just connected receives a response, this means that it has an address that already exists on the network, highlighting it immediately to avoid subsequent collisions.

ARP is therefore very useful for finding the corresponding Ethernet address from a given IP address. However, there is also the inverse problem that is solved by RARP. In such a system, a machine that has just been switched on knows its own Ethernet address and sends a broadcast request stating that address and requesting a relevant IP address. The RARP server receives this request, checks its archive and sends the corresponding IP address. This mode is very useful on computers that do not have an operating system and need to download it from a central server: in this case, the computer is assigned a different IP address for each machine and then downloads the operating system that is the same for each machine.

Since the broadcast request message is composed of all 1s, this message is not forwarded by routers and as such, an RARP server for each network in which such a service is requested must therefore be present. To avoid this, an alternative BOOTP was devised that uses UDP datagrams that are, therefore, forwarded by routers. The problem of BOOTP is represented by the fact that manual configuration of the tables that associate IP addresses with Ethernet addresses is required and if a new computer is added to the network, this computer cannot communicate until its data are inserted within the table.

To avoid this problem, the DHCP was introduced that allows manual or automatic assignment of IP addresses. The DHCP server does not necessarily have to reside on the same network as the requesting host. Because this server may not be accessible from broadcast transmissions, a DHCP relay agent must be provided on each network. In this case, as soon as the computer is switched on, it sends a DHCP DISCOVER packet that is intercepted by the DHCP agent and sent in unicast mode to the DHCP server whose address is well known by the DHCP agent. A typical problem is represented by the allocation of IP address, since the machine may not communicate its output from the network with consequent loss of its IP address. To avoid this, a technique called leasing is used in which the IP address is assigned for a certain period of time, after which the computer is asked for confirmation: if a response is received, the IP address is left otherwise this address is recovered and reassigned to another computer.

In most cases, IP transmission takes place between a transmitter and a receiver but in some situations, it may occur between only one transmitter and several receivers as in the distribution of distributed databases or in digital audio conferencing. The IP protocol is able to support such transmission using class D addresses as such addresses identify a group of hosts and 28 bits identify the groups, and as such about 250 million groups can be reached. In this case, multicast transmission is possible because of the special multicast routers that may be assisted by standard router.

IP may also be used in mobility by resorting to appropriate mechanisms as displacement of the IP address from one network to another can cause management problems, since the system and all other devices have to keep track of the location of the user. Practically, every website that wants to allow its users to move must create a fixed agent (home agent) and every website that wants to allow access to mobile users must create an external agent (foreign agent). When the user wants to use an external website, they must present themselves and register and the related local foreign agent contacts the home agent providing them with a temporary IP address, represented, usually, by its own IP address that is used by the home agent to redirect data packets.

Even though protocols such as NAT can momentarily buffer, exhaustion of addresses is close to manifesting itself. For this reason, already by 1990, work had started on a new version of IP capable of never exhausting addresses and characterised by the following properties:

1. Ensuring the addressing of billions of hosts even in the presence of an inefficient allocation of addresses.
2. Reduction in the size of routing tables.

3. Simplification as far as possible of the protocol in order to allow fast processing of packets by routers.
4. Ensuring a minimum level of security, through authentication and confidentiality, with respect to the current version.
5. Paying more attention to the type of service with particular reference to applications in real time.
6. Improvement of the multicast transmission.
7. Allowing displacement of the host without having to reassign addresses.
8. Allowing evolution of the protocol.
9. Allowing coexistence between old and new protocol.

After many discussions, a protocol called Simple Internet Protocol Plus (SIPP) was selected that took the name of IPv6.

This protocol has a greater number of addresses with respect to IPv4 and has a more simplified header of only seven fields against the 13 of IPv4.

The heading of IPv6 is shown in Figure 1.79.

The "version" field is characterised by a value equal to 6 for IPv6 and 4 for IPv4. This field enables routers to identify the type of packet in transit and to behave accordingly.

The "traffic class" field is used to classify packets based on the specific needs of transmission (real-time data).

The "flow label" field is used to allow a source and a destination to set a pseudo-connection with certain properties and characteristics.

The "payload length" field is used to indicate the number of bytes that follow the 40-byte header shown in Figure 1.79.

The "next header" is used to indicate which of the six extended headers follows the current header.

The "hop limit" field is used to limit the useful life of the packets.

The "source address" and "destination address" fields are immediately understandable. These are of fixed length of 16 bytes, which are written as eight groups of four hexadecimal digits separated by two points. The addresses of the type IPv4 may still be used always expressing them in decimal notation,

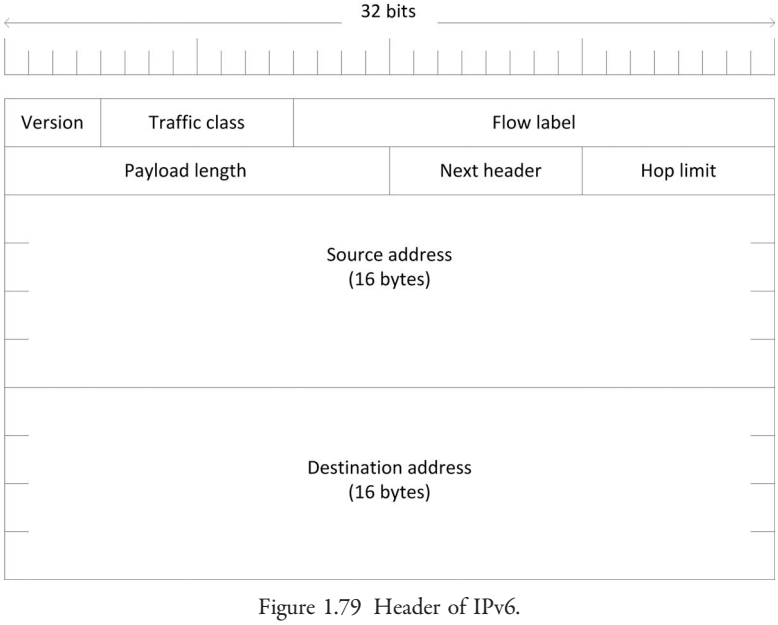


Figure 1.79 Header of IPv6.

using four numbers ranging from 0 to 255, separated by dots, with the precaution of beginning this notation with two pairs of double points.

Using the 16-byte notation, there are $2^{16 \times 8} = 2^{12}$ addresses, corresponding to 3×10^{38} addresses.

To get an idea of the density of addresses per square metre of the earth's surface, including the areas covered by water, we need only to divide the number of addresses by the surface of our planet to obtain 7×10^{23} addresses per square metre.

Since some missing fields of IPv4 are sometimes still used, IPv6 is characterised by an "extension header" that is used to provide additional information that is efficiently encoded.

1.6 The transport layer

The transport layer represents the central nucleus of the protocol stack. It deals with the transport of data reliably and efficiently from the transmitting host to the receiving host regardless of the type of network used. The main purpose is to provide an effective service to the overlying application layer using the underlying network layer.

In this sense, all the hardware and software that perform this service work within a virtual drive called transport entity, located in the operating system kernel, in a separate user process, in a packet of libraries relating to network applications or in a network interface card. The logical relation existing between the layers of application, transport and network is illustrated in Figure 1.80, being the message exchanged between a transport entity and the other named transport protocol data units (TPDUs).

The transport services, similarly to network services, can be connection oriented or non-connection oriented.

The connection-oriented service, similarly to the relative network service, provides three operational steps: constitution, data transfer and release.

Non-connection-oriented service is also very similar to the relative network service.

The existence of the transport layer allows the relevant service to be more reliable than the underlying network layer, as lost packets and damaged data can be identified and corrected by the same transport layer.

Because of this layer, applications using a standard set of primitives can be created to have programs running on different networks, without having to worry about the management of interfaces of subnets and unreliable transmissions.

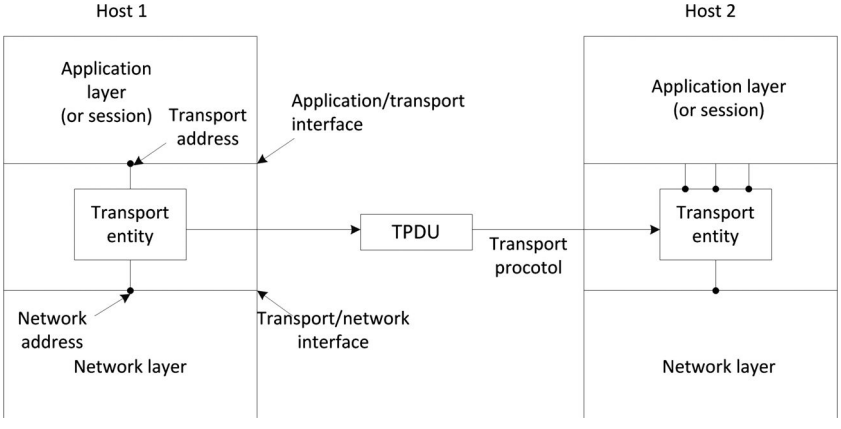


Figure 1.80 Logical relation between the layers of application, transport and network.

It performs the key function of isolating the upper layers from the technology, from the structure and the imperfections of the underlying subnetwork.

For this reason, a division between the four lower layers is usually created, which may be regarded as the suppliers of transportation service, from the upper layers, which can be considered as users of the transportation service, being the transport layer the boundary between the two groups.

The transportation service is very similar to the service network with the difference being that the second needs to model what is being offered by real networks that can lose packets thus making this service unreliable. The transportation service, on the contrary, must ensure a reliable service to an unreliable network.

A further difference between the transportation and network service relates to the addressees of the same: few people access the network service while there are many people who have access to the transportation primitives (seen as consumers and as programmers).

A basic example of service primitives can be represented by the following:

1. LISTEN that does not send any packet but which remains pending until a process tries to connect.
2. CONNECT that sends a request connection packet and actively tries to establish a connection.
3. SEND that sends data packets.
4. RECEIVE that does not send any packet and waits for receipt of the data.
5. DISCONNECT that sends a connection closure request packet and alerts that it intends to release the same.

Disconnection may be asymmetric or symmetric. In asymmetric disconnection, each user can perform the DISCONNECT primitive and arbitrarily release connection. In symmetric disconnection, when a host performs primitive DISCONNECT, this means that it no longer has data to send but can still receive data: it is waiting for implementation of a similar primitive on the other side before releasing the connection.

The transportation service is implemented by a transport protocol used between two transport entities. It is very similar to the data link protocols with the difference being that while the latter communicate directly through a physical channel in the transportation layer, the physical channel is replaced by the subnet.

If an application process needs a connection with another remote application process, it must specify to which process it intends to connect. The method that is used consists of the definition of transportation addresses through which the processes can receive connection requests. If the Internet is being used, these end points are named ports while, in general, they are called transport service access points (TSAPs). The corresponding end points in the network layer are known as network service access points (NSAPs). A typical example of NSAP is represented by an IP address. Figure 1.81 shows a diagram that illustrates the relation between transport connection, TSAP and NSAP.

The application processes, both of the client and of the server, connect to a TSAP to establish a connection with a remote TSAP. These connections pass through the NSAP on each host. While there can be several TSAPs on a single host, there can be only one NSAP shared on the same host and therefore this NSAP must be able to identify in various TSAPs of the upper level it is serving.

A big problem is represented by the proper management of duplicates that can be generated when there is congestion on the network and a transmitting host resends new packets that have already been sent thinking that previous packets have been lost without knowing that they will re-emerge from the congestion area after some time, as long as the scope of their useful life is still valid.

For this reason, it is necessary to limit the life of a packet up to a maximum value using the following techniques: design of limited subnets, insertion of a hops counter in each packet and application of a timestamp to each packet.

The basic principle is to enumerate the various TPDUs and avoid the existence on the same network of two TPDUs numbered in the same manner. In this sense, each host may be equipped with an internal clock, not necessarily synchronised with the others. When a connection is set, the lowest

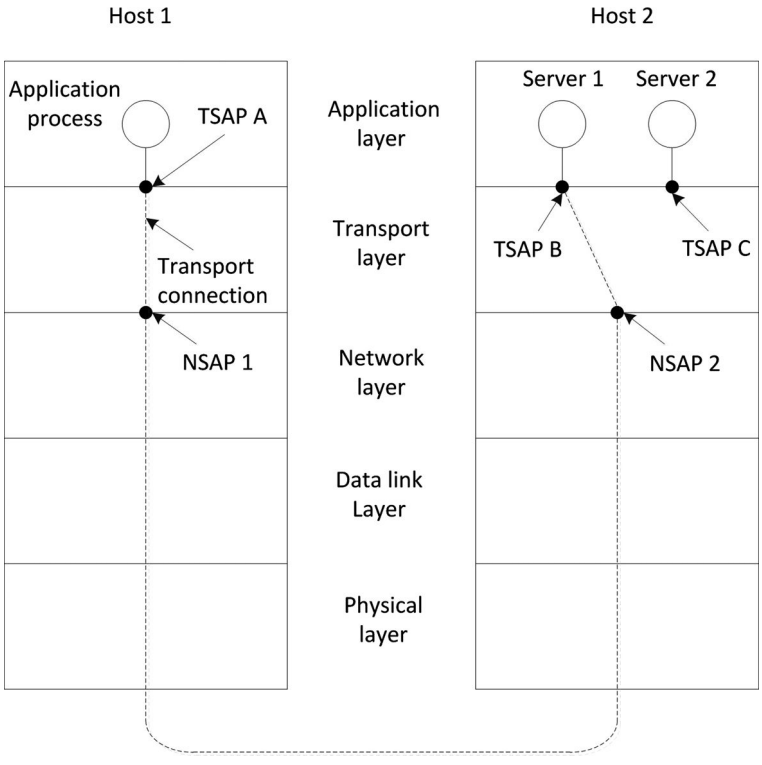


Figure 1.81 Relation between transport connection, TSAP and NSAP.

order k bits of the clock are used as the number of initial sequence. In this way, each connection begins to enumerate its TPDU with a different sequence number. It must of course be guaranteed that the sequence space is sufficiently large in such a manner that, when the sequence numbers resume from the beginning, the older TPDUs are no longer in circulation on the network. The linear relationship between numbers of initial sequence and time is shown in Figure 1.82.

If a host is subject to a block when it restarts, it cannot know at what point of the sequence space it was. A possible solution is to wait for a T parts time to allow the removal of all the old TPDUs,

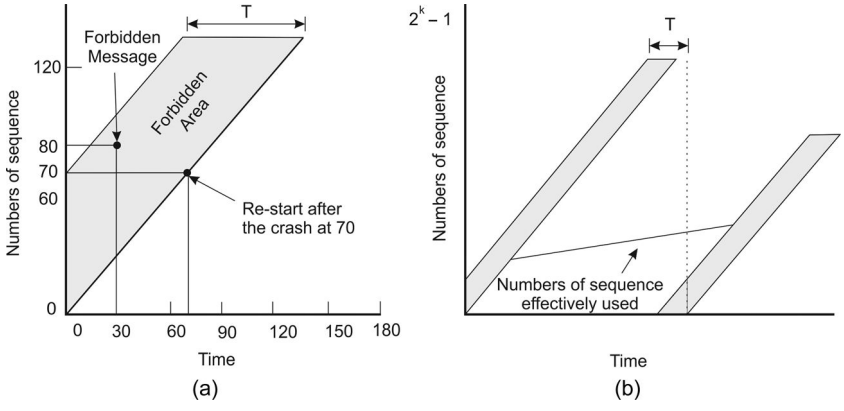


Figure 1.82 Relation between sequence numbers and time. (a) The TPDUs may not enter the prohibited area. (b) The problem of re-synchronisation.

avoiding having to enter the prohibited area indicated in Figure 1.82. Before sending a TPDU, the transportation entity must read the clock and verify that it is not in a prohibited area. If a host sends data too quickly on a recently opened connection, the actual sequence number could grow faster than the initial sequence number on the time chart: this means that the data rate may not exceed a TDPU clock tick and that the transport entity, before reopening a connection following a block, must wait up to a clock tick to avoid using the same sequence number twice. In any case, the transport entity always has to control the internal clock to avoid entering into prohibited area. In this case, it either delays by a T time or re-synchronises the sequence numbers.

1.6.1 The UDP transport protocol on the Internet

With regard to the Internet, there are two protocols in the transport layer: one without a connection, called UDP, and one connection oriented, called TCP.

The UDP allows applications to send encapsulated IP datagrams without having to establish a connection. This transmits segments consisting of an 8-byte header followed by the payload. This header is shown in Figure 1.83.

The source port is used when a response has to be sent to the source: by copying the relevant field from input segment in the destination port field of the output segment, the process that must send the response can indicate which process on the source host must receive it.

The "UDP length" field (UDP length) includes the 8-byte header and data.

The "UDP checksum" field is optional and contains 0 if it is not processed.

UDP does not address the flow control, error checking or retransmission after receiving an incorrect segment, all tasks that are carried out by the user process.

UDP is useful for brief communications between client and server, where a short answer is required and if the latter is lost, the client can go into time out and request it again.

UDP is used by an application called DNS that will be illustrated later, where the application layer will be illustrated. It is substantially used when a host needs the IP address of a server on the network knowing only its full name (e.g. www.servercercato.edu). In this case, the host may send a UDP packet to a DNS server and wait for a response indicating the IP address of the server required and contact it directly. Preparation and release of the connection is not therefore necessary, merely two messages (one of request and one of response).

UDP is therefore a simple protocol but with niche uses because most Internet applications require a reliable delivery in sequence that UDP is not able to provide.

1.6.2 The TCP transport protocol on the Internet

In order to guarantee the transmission of a stream of bytes in a reliable manner between two hosts using an unreliable network, the protocol TCP was created. TCP was designed to adapt to the properties of the network such as the topology, bandwidth and packet size, ensuring reliable performance in the presence of various types of error.

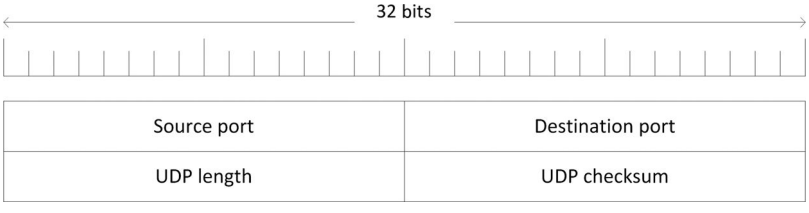


Figure 1.83 The UDP header.

Any computer capable of supporting TCP has a TCP transport entity, represented by a library, a user process or a part of the kernel. These objects handle the TCP streams and interfaces to the IP layer. The TCP entity receives data flows of the users from local processes and divides them into parts that are not larger than 64 kB (usually equal to 1,460 bytes of data such that they fall within a single Ethernet frame including the IP headers and TCP). Each part is then sent in an autonomous IP datagram. These datagrams reach the remote host and are received by the corresponding TCP entity that returns the original byte streams.

Since the IP layer does not guarantee delivery without datagram errors, TCP is responsible for performing timeout and possible retransmission as required. Datagrams may also arrive with a different sequence compared to that of sending: in this case, TCP deals with restoring the original sequence. In practice, TCP ensures the necessary reliability that IP is not able to ensure.

TCP, in order to operate correctly, creates end points, called socket, both on the side of the sender and of the receiver. Each socket has an address (socket number) comprising the IP address of the host and a number local to the host called port. The TSAP in TCP is in fact the port. To have a TCP service, a connection between a socket of the transmitting host and a socket of the receiving host must be established. The most popular protocols are shown in Table 1.8.

Each socket is able to support multiple connections simultaneously. All the TCP connections are full-duplex point-to-point, that is traffic can flow simultaneously in both directions only two between two users at a time (multicast or broadcast is not supported).

A TCP connection is represented by a stream of bytes and not by a stream of messages.

Each byte of a TCP connection is characterised by its own 32-bit sequence number. This was not an issue at the dawn of the Internet, when 56 kbps connections were being used, as a host that operated at maximum speed, and a continuous cycle needed a week to exhaust all the sequence numbers. Currently, given the higher baud rates available, the problem of exhaustion of sequence numbers has become a concern.

TCP entities exchange data in the form of segments. A TCP segment is composed of a 20-byte header followed by zero or several data bytes. The TCP software decides on the size of the segments, and can insert in the same segment the data from multiple scripts or divide the data of one script into multiple segments. There are two segment size limits. Each segment, including the TCP header, must be able to be contained in the payload of 65,515 IP bytes. Each network has its own maximum transfer unit (MTU) and it must be possible to contain each segment in the MTU. Since the MTU is usually equal to the size of the Ethernet payload (1,500 bytes), the limit is the maximum segment size.

When a sender transmits a segment, it also starts a timer. When this segment reaches the TCP entity of the recipient, the latter sends a segment marked by an acknowledgement number equal to the next sequence number that it expects to receive. If the sender's timer expires before receipt of that confirmation segment, the sender retransmits the segment.

Table 1.8 Main ports used by TCP.

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	Email
69	TFTP	Trivial file transfer protocol
79	Finger	User information research
80	http	World Wide Web
110	POP3	Email remote access

This can cause problems since the segments may arrive at their destination in a different order from that of departure, or may be delayed in such a way as to send the timer of the sender in timeout, making it to repeat the sending operation.

The structure of a TCP segment is shown in Figure 1.84.

Each segment begins with a header of 20 bytes with fixed format that can be followed by a number of header options. After the options, follow up to 65,535 – 20 – 20 = 65,495 data bytes, the first 20 bytes referring to the IP header and the second 20 bytes to the TCP header.

There may also be segments without data that are used for control and acknowledgement messages.

The "source port" and "destination port" fields serve to identify the local connection details.

The "sequence number" and "acknowledgement number" fields perform the functions already illustrated.

The "TCP header length" is used to indicate how many 32-bit words are contained in the TCP header.

The 1-bit flags that follow are:

1. URG (urgent pointer) that, when set, indicates the offset in bytes in which the urgent data are found;
2. ACK (acknowledgement) that, when set to 1, indicates the "acknowledgement number" is valid;
3. PSH (push) that indicates the presence of push data, namely data that must be delivered to the application upon arrival without being stored in the memory buffer;
4. RTD is used to re-set a connection that has become inconsistent due to a crash of the host or for other reasons;
5. SYN that is used to establish connections;
6. FIN that is used to terminate a connection.

The "window size" field is used to indicate how many bytes can be sent from the one that has received acknowledgement.

The "checksum" field performs verification of the header, data and conceptual pseudo-header.

The "options" field is used to add extra features not available in the normal header.

It has already been said that when the load applied to a network is greater than its managerial capacity, the result is congestion. It is also already been shown how the network layer attempts to manage congestion. In reality, most of the work in this sense is carried out by TCP as the most effective solution to avoid the phenomenon resulting in data input speed reduction. In this sense, the avoidance

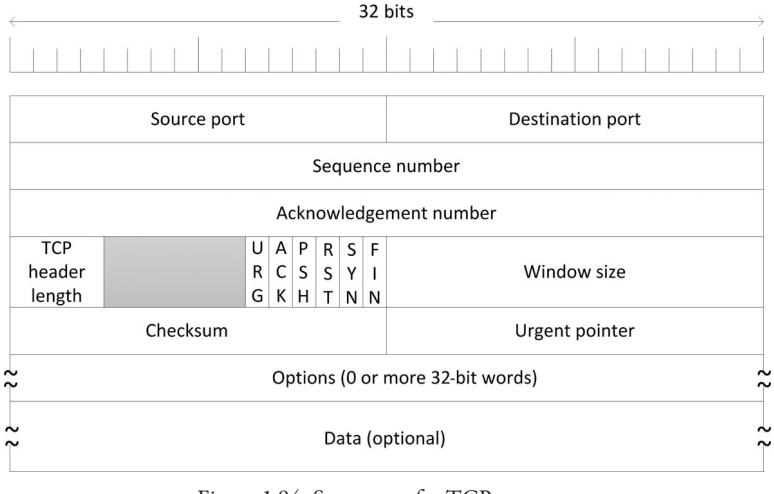


Figure 1.84 Structure of a TCP segment.

of inclusion of a new packet on the network is sought until the old packet leaves, that is until the same is delivered.

To manage congestion, they must first be identified. At the dawn of the network, timeout of the packets was used that could be caused by interference on the transmissions line rather than by real congestion: at the time, it was very difficult to understand the difference. Currently, packet loss due to transmission errors is a rare event (case aside is represented by wireless networks) and the majority of cases of timeout is represented by congestion.

In the case of the Internet, both the capacity of the network and the capacity of the receiver represent two critical elements that must be managed appropriately, failing which results in congestion. In this sense, every sender uses two windows: the window that the recipient has guaranteed and a congestion window; each of them represents the number of bytes that the sender can transmit. The actual number of bytes consists of the smaller of the two windows in question. In practice, the actual window is represented by the smallest of the values of what the sender thinks is right and what the recipient thinks is correct.

When a connection initialises, the sender initialises the congestion window to the size of the segment most used on the connection and sends a segment. If this segment receives an acknowledgement before the timer expires, the sender adds to the value of the congestion window a number equal to the size of the segment in such a manner that its size is twice the maximum segment size and then sends two segments, repeating the same process until the window reaches *N* segments. Essentially, each group that receives the acknowledgement doubles the size of the congestion window. This window grows exponentially until timeout or until the window of the recipient is reached. This algorithm is also called slow start even if, in fact, it is exponential and is supported by all TCP implementations.

In order to control congestion on the Internet, a third parameter, called threshold, is used whose initial value is set at 64 kB. When timeout occurs, the threshold is set to half of the congestion window and the congestion window is set to the maximum size of a segment. A slow start is then used to determine the network management capacities and exponential growth ends when the threshold is reached. From that moment onwards, the transmissions that have been successful increase the congestion window in a linear fashion until the next timeout or until reaching the receipt window. This operational principle is illustrated in Figure 1.85.

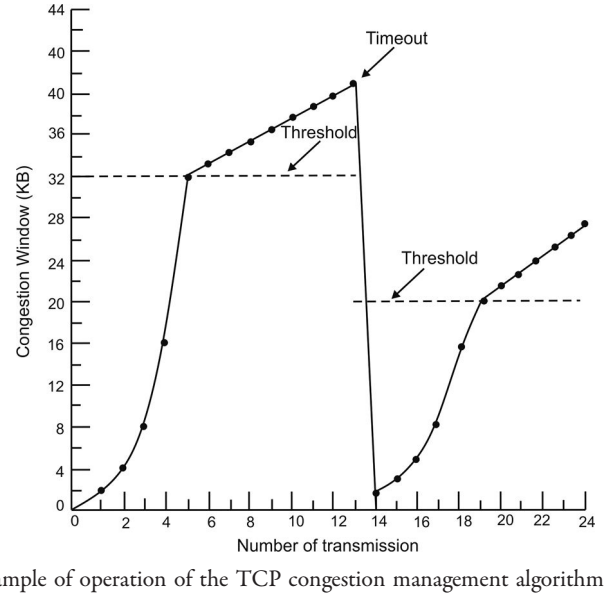


Figure 1.85 Example of operation of the TCP congestion management algorithm on the Internet.

TCP uses different timers to do its job properly. The most important timer is represented by the retransmission one that is activated whenever a segment is sent. If the timer receives acknowledgement before the deadline, the timer is stopped; if the timer expires before receipt of the acknowledgement, the segment is transmitted again and the timer is restarted. The choice of the count interval of the timer is a problem that is not easy to solve.

This problem has a fairly straightforward solution on the data link layer because the estimation of the delay is largely predictable, that is its variance is low. In this case, the timer can be set to expire immediately after the estimated time of receipt of the acknowledgement. Since in the data link layer the acknowledgements are very rarely delayed, the absence of one of the latter, when expected, is used to indicate that the frame or the acknowledgement itself is lost. A typical trend of the probability density relative to the time required by an acknowledgement to return is shown in Figure 1.86(a).

TCP must instead operate in a completely different environment in which the probability density function relative to the time of return is characterised by a variance greater than in the previous case, as shown in Figure 1.86(b).

In this case, it can be very difficult to determine the round-trip time and if it were possible, it is not easy to determine the timeout interval: if it is too short, there would be a risk of continuous retransmissions that would clutter up the Internet with useless packets; if it were too long, the performance would suffer due to the long delay in retransmission required every time a packet is lost.

To avoid these problems, highly dynamic algorithms were developed that determine the timeout interval according to the values returned by continuous measurements of network parameters.

However, there are other types of timers, such as that of persistence, keepalive and time wait, which will not be addressed for reasons of space.

In general, transport protocols should be independent of the technology of the network layer that operates below them, not having to differentiate if the technology is, for example, fibre-optic or radio wave (wireless). In reality, this becomes important as TCP optimisation is performed assuming operation with the most reliable cable networks rather than with the less-reliable wireless networks. If performance of the transmissive medium were neglected, there would be a risk of extremely poor TCP connections on wireless. The main obstacle is represented by the TCP algorithm for congestion control, which has already been addressed, which presupposes, mainly, that timeout is generated from congestion and not from lost packets (as frequently happens on wireless networks), slowing down transmission where necessary and sharing progressively. As such, if on wired networks, when a packet is lost, slowing down is necessary, then on wireless networks, when a packet is lost, more forceful reattempts are required. In most cases, the path of a network can be heterogeneous, since most of the

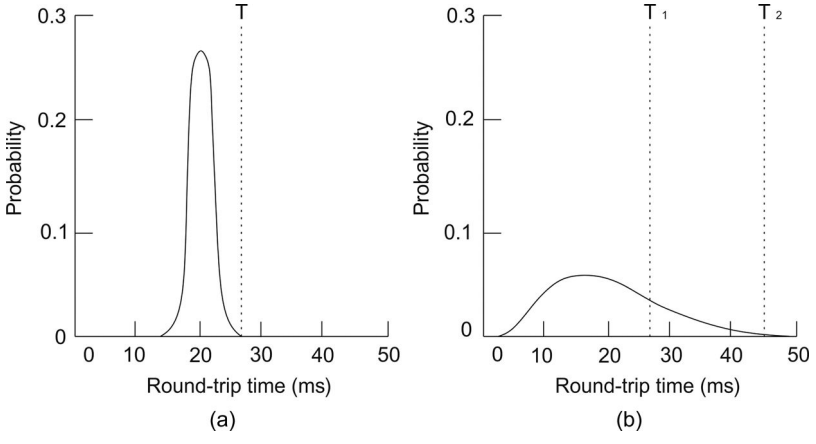


Figure 1.86 Probability density of the arrival times of acknowledgement. (a) Data link layer, (b) TCP.

section can take place on a wired network while the last few metres take place over a wireless network. A possible solution is to split the TCP connection into two separate connections, generating what is called TCP indirect, as shown in Figure 1.87.

In this case, there are two homogeneous connections: while timeout of the first connection can slow down the sender, those of the second can accelerate it, but it may also adjust other transmission parameters. As each part of the connection represents a complete TCP connection, when the radio base station sends the acknowledgement, this does not mean that the end user has received it, violating the TCP semantics. There are however other techniques that can be configured to operate correctly, after appropriate modifications, without violating the TCP semantics. These techniques will not be illustrated for reasons of space.

1.6.3 Performance on networks

The performance of networks is very important when the same are connected to thousands of hosts that must communicate with each other and unexpected interactions may occur.

The performance of complex networks, in most cases, is very difficult to define scientifically because the basic rules are almost ignored when the entire network system interacts in a complicated manner.

Problems arise not only in the transport layer but also in the lower network layer, even if the latter is more directed to the control of routing and congestion.

In the following, we will discuss some of the fundamental aspects of the performance of networks that are performance problems, measuring of network performance, system design for better performance, fast processing of TPDU and protocols for future high-performance networks.

1.6.3.1 Performance problems

Typical problems, such as congestion, are generated by momentary overloads of network resources, such as a router that reaches and exceeds, at a certain moment, its capacity to manage traffic. Performance can also degrade in the presence of an imbalance of resources, for example a fast network connected with a slow receiving host.

Overloads may also arise in a synchronous manner. If, for example, a TDPU contains an incorrect parameter, the receiver will send an error notification. If this TDPU was sent in multicast or broadcast mode to a large number of hosts, the latter, in an almost synchronous manner, would send back the same error message generating what is called broadcast storm, which could overwhelm the network itself. Another example of synchronous overload can occur when there is a power cut. When power is restored, all computers and devices begin boot sequence, eventually requiring information from their DHCP reference server: if the number of machines and devices is very high, there may be network congestion.

Another problem to be optimised is represented by the correct choice of timeout. This problem has already been addressed and as such will not be explored further.

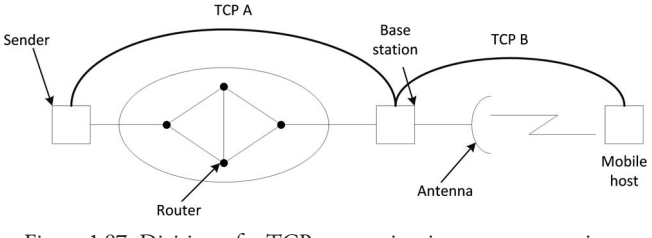


Figure 1.87 Division of a TCP connection into two connections.

Another problem is represented by the protocols used on gigabit networks over long distances. In fact, when transmitting a packet, these protocols await the return of the acknowledgement which, over long-distance networks, can return after a few tens of milliseconds, wasting the time necessary for transmission and reducing the efficiency of the network itself. In this sense, a very useful parameter is represented by the bandwidth-delay product, which is obtained by multiplying the bandwidth, expressed in bits/s, for the round-trip delay time, expressed in seconds. In this way, an impression can be formed of the maximum amount of data that can be transmitted before receiving the first acknowledgement. For this reason, in order to have high performance, the window of the recipient must be equal to or greater than the bandwidth-delay product. For a transcontinental gigabit line, at least 5 Mb is required.

Another problem is represented by the jitter that affects the applications in which time is a fundamental requirement. In these applications, not only the average transmission time must be relatively small, but the relative standard deviation must also be as small as possible.

1.6.3.2 Measuring network performance

To increase the performance of a network, a precise understanding of exactly what happens on the network is necessary and to do this, appropriate measurements must be performed.

To improve the performance of a network, the following steps must therefore be performed: measurement of performance and relevant network parameters, an understanding of the operating mechanisms and parameter change. These steps must be performed several times to optimise the network itself.

Measurements can be performed in different ways and places. The simplest measurement is to activate a timer at the beginning of an activity to check the time required to perform the same activity.

The measurement of network parameters may present a series of difficulties such as: insufficient number of samples, use of non-representative samples, use of an inaccurate clock, unexpected events during testing, the influence of caches, incomprehension of the measurement and extrapolation of the results.

With regard to the insufficiency of the number of samples, it must be emphasised that it is not only the sending time of just one TPDU that needs to be measured but this operation needs to be performed several times before calculating the average and failure to do so would produce an insignificant result.

With regard to the use of non-representative samples, the same measurements referred to above should not be performed during the same period of the day but at different times and on different days in order to take into account the variability of the network load.

With regard to the use of an inaccurate clock, it must be considered that such a situation can cause great errors in measurement results.

With regard to the unexpected events during the test, it should be remembered that when running the tests, the networks may be loaded with unwanted applications. For this reason, measurements must be taken when it is certain that the networks are empty, artificially generating the desired load to take measurements.

With regard to the influence of caches, it is important to remember that in order to measure the time taken to transfer a file characterised by considerable dimensions, it must be opened and read in its entirety, closed and the sending time must be recorded. This measurement must be performed several times. If the computer has a memory cache, only the first reading represents a significant value because the following readings represent simple readings from the cache memory.

With regard to the misunderstanding of the measurements, it must be remembered that when, for example, measurement of the reading of a remote file is performed, the results depend on the network, on the operating systems of both the client and the server, on the network interface cards, drivers and

many other factors. The measurements should therefore be carried out taking into account all these factors. Relevant measurements can therefore vary considerably from system to system.

With regard to the extrapolation of the results, it should be borne in mind that an initial linear trend is not said to continue as such for high-load conditions, where response time tends to assume a behaviour of the type 1/ (1-load), as shown in Figure 1.88. For this reason, great attention should be paid when performing extrapolation.

1.6.3.3 System design for better performance

An optimised initial design cannot be replaced by any compensatory measure carried out by following accurate measurements. A network of poor quality remains the same even after following corrective actions.

There are a series of practice rules, represented by the following:

1. The microprocessor speed is more important than the network speed.
2. Reduction in the number of packets to reduce the software overhead.
3. Minimisation of context changes.
4. Minimisation of copies.
5. Possibility of extension of the band but inability to decrease the delay.
6. Minimisation of congestion.
7. Minimisation of timeouts.

With regard to point 1, it has been seen that, in most networks, overhead of the operating system and protocols becomes dominant on the actual transmission time on the medium. For this reason, it may happen that, in order to speed up the baud rate, capacity of the network is increased without obtaining significant results since the criticalities, in terms of delay time, are represented by computers and their software.

With regard to point 2, it should be remembered that the processing of a TDPU causes a certain overhead and a certain amount of processing per byte. When the amount of data to be sent is high, the overhead per byte remains more or less the same and can reach considerable values. In addition, the overhead can occur at lower levels as the arrival of a packet causes an interrupt at processor level, which must perform a series of buffer operations such as saving of registers, the saving of operations in progress and so on, interrupting the normal operational flow and slowing down the computer itself. For this reason, if the TDPU is reduced, the overhead of the packets and microprocessor interrupts will be consequently reduced.

With regard to point 3, it must be remembered that the context changes, for example passage of the operating system from kernel mode to user mode, greatly slow the computer on which this operating

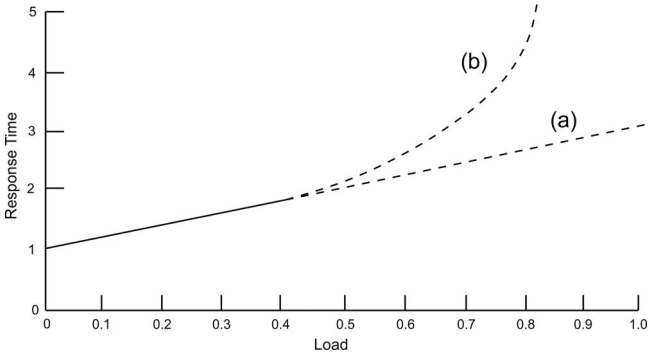


Figure 1.88 Response time trend according to the load in (a) the ideal case and (b) the real case.

system is running. The context changes can be reduced if the application that sends the data accumulates the latter in a buffer to send them all together. In the same way, the small input TDPU should be collected and sent in a block when they reach a certain size, in order to avoid context changes. Context changes can also be multiple upon the arrival of a single packet, causing a significant slowdown whenever that event occurs.

With regard to point 4, it is recommended that the multiple copies that slow down the system even further with respect to context changes are minimised. It often happens that a packet is copied many times before the TDPU that encloses it is delivered, greatly slowing down the computer when there is a high stream of input data.

With regard to point 5, it is evident that if larger bandwidth is required, it need only be purchased from the service provider. If the network is in optical fibre, it may be an idea to use a second fibre in such a way to double the bandwidth without increasing delay. Delay due to the computer can be reduced by improving the software that handles protocol, operating system or the network interface while that due to the transmission time remains as such.

With regard to point 6, it is worth remembering that it is very important to avoid congestion anticipating it in good time in order to avoid packet loss, waste of bandwidth, delays and so on, without neglecting the enormous commitment of resources and time necessary to recover congestion.

With regard to point 7, it should be remembered that timeouts should be reduced as much as possible to avoid repeating necessary, but at times dangerous, actions due to the generation of congestion, such as the resending of packets. For this reason, adjustment of timers in a conservative way is necessary as a timer that expires late introduces a further delay in connection while a timer that expires too soon uses up the microprocessor, wastes bandwidth and further loads many routers without a valid reason.

1.6.3.4 Fast processing of TPDU

It has been seen that one of the fundamental factors that slows down the speed of the networks is represented by the software that manages protocols. There are many paths that can be taken to optimise software. Some of them are already known such as buffer and timer management. This section will not go into specific detail for reasons of space.

1.6.3.5 Protocols for future high-performance networks

Around the beginning of the 1990s, the first high-performance networks, called gigabit networks, began to appear. The old standardised protocols that soon showed all their limits on those networks were used on such networks.

An initial problem is represented by the already mentioned use of 32-bit sequence numbers. Initially, 56 kbps leased lines were used, which means that a host that sent data at the maximum speed for 24 h a day exhausted the sequence numbers in a week before recommencing the cycle. With 10 Mbps Ethernet, the sequence number cycle exhausted in about 1 h while with 1 Gbps Ethernet, the cycle time lasts a little over 30 s. Since packets on the Internet have an average lifespan of around 120 s, there is a risk of introducing onto the network packets with the same sequence number.

Another problem is represented by the greater increase in baud rate over time with respect to the increase in processing speed. This means that the time available for processing by protocols must always be shorter and the latter must always be more simplified.

Another problem is the fact that gigabit lines are limited by the delay while megabits lines are limited by the bandwidth. This means that further increasing the bandwidth does not result in any benefit because transmission is dominated by delay. For example, Figure 1.89 shows, as a practical case, the time required to transfer a file of 1 Mb along a 4,000 km line at different baud rates

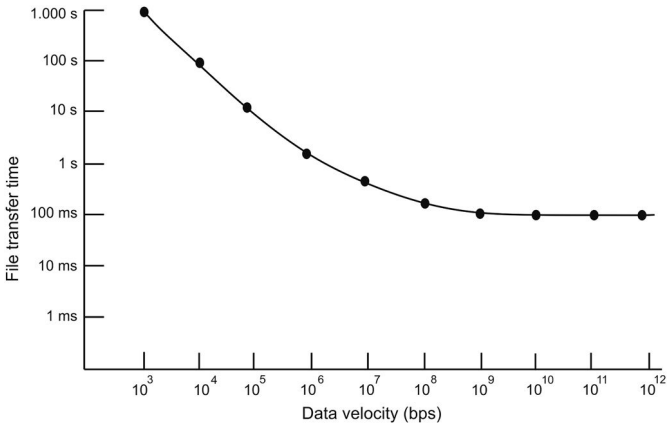


Figure 1.89 Times for transfer and receipt of the acknowledgement of a 1 Mb file for different baud rates.

It can be seen how, at a rate of 1 Mbps, the transmission time (1 s) is dominated by the speed of sending bits, while at 1 Gbps the round trip delay of 40 ms dominates the sending time of the file equal to 1 ms. Further increasing the band does not result in any benefit.

Another problem is the variance of delay times which, for multimedia applications, is as important as the average delay itself. In such applications, a relatively reduced speed, but with characteristics of uniform delivery, is preferable to a high speed but with characteristics of variable delivery.

For gigabit networks, it is therefore important to design with speed in mind and not optimisation of the bandwidth that is in any case very wide in such networks.

1.7 The session layer

The main function of the session layer is to provide a way for session users to establish communicative sessions between remote processes and can transfer the data on them in an orderly fashion.

A session is a connection between active processes on remote machines, already linked at the transport level by a connection. A session has a close resemblance to a transport connection, except that the session is between processes and the connection between hosts.

It is possible that when the session layer receives a request to establish a session, a transport connection to enable the session should also be established. When the session is completed, the transport connection can be released or not.

There are various operating modes, which are represented by:

1. multiple sessions per connection: for example a company with distributed sales offices. When the first transaction request is received, a session with the central management system is activated. At the end of the transaction, the session is terminated but does not release the connection because it will be needed again within a short space of time. The setup allows making multiple sessions to use the same connection;
2. multiple connections per session: for example, if a connection fails, the session layer may establish a new connection on which to continue the session.

A session, the equal of a connection, passes through three stages: activation, use and release.

The main difference with the connection is the way in which the sessions and connections are released:

1. connections are terminated abruptly and may cause irreversible loss of data in transit at the time of release;
2. sessions are released gradually, not leading to the loss of data.

Dialogue management (full-duplex or half-duplex) establishes rotation shifts in communication and is one of the services that can be provided by the session layer upon request.

Dialogue management is implemented through the use of a data token.

If a half-duplex mode session is established, initial negotiation also establishes the identity of the first token holder. Only the user who holds the token can transmit data while the other must remain silent. When the token holder has finished transmitting, the token is passed on.

Another service of the session layer is represented by synchronisation that serves to report the session entities in a known state in the event of an error or in any case of an inconsistency. Implementation of this service undergoes maintenance by level 5 software of a state associated with the session, which is restored in the event of detection of inconsistencies.

A characteristic closely linked to synchronisation is the management of activities. The idea is to offer the user the opportunity to split the message into logical units called activities. Activities are independent of each other and are therefore natural candidates to identify the points on which to define the states to be maintained. It is the user who has the task of determining what an activity is.

1.8 The presentation layer

The presentation layer was initially designed to allow communication between machines with different internal representations (e.g. EBCDIC – ASCII), not only from one code to another, but also for the numeric format representation (one or two complement); for small or large final representation and for memory management (4-byte or 2-byte words, etc.).

The needs arising from the exchange of multimedia data or in any case structurally different data have led to the need to provide for the possibility of using a format in transmission different from the format used in programming.

Other features of this layer are represented by:

1. data compression;
2. encryption;
3. greater transmissive reliability.

With regard to data compression, this can dramatically speed up the transmission of data. There is a compromise between baud rate and reliability. Encodings that allow speeding up of transmission can be identified, with the same reliability, with respect to that obtainable with “natural” encodings of data to be transmitted.

With regard to cryptography, this is linked to transmission security and is illustrated in detail in the following chapter. In addition to the need for efficient and fast transmission, confidentiality of the data being transmitted is also very often required. Is it possible to use encryption algorithms that re-encode data in a functional format depending upon the needs of the specific session.

With regard to transmissive reliability, the presentation layer may be asked for more detailed error checking than that guaranteed by checksum or Cyclic Redundancy Check (CRC). This translates into the use of codes with given corrective properties to be used in the stages of transmission.

1.9 The application layer

The application layer is directed primarily at the end user, contrary to the underlying layers that are aimed at ensuring as reliable a service as possible.

There are also protocols on the application layer that allow applications to work properly such as the DNS, which will be illustrated later on together with widely used applications such as email, the WWW and multimedia.

1.9.1 The domain name system

The DNS was developed to address the difficulty that users would have remembering network addresses off by heart (it should be remembered that current network addresses are composed of four variable numbers between 0 and 255, separated by dots).

To avoid this, it was decided to use addresses composed of ASCII characters, easier to remember and use, minimising the risk of errors. It was therefore necessary to devise a system that would convert addresses composed of strings of characters into valid IP addresses.

DNS is a hierarchical naming scheme based on domain combined with a distributed database system for the implementation of this scheme. It is used to associate host names and email addresses, expressed in strings of characters, with IP addresses. It can also be used for other purposes. To associate a name with an IP address, an application program activates a procedure named resolver, providing it with the desired name as a parameter. The resolver sends a UDP packet to a DNS server, which seeks within it the IP address associated with that name and returns it to the resolver. The resolver passes the received IP address to the application program that requested it, allowing it to establish a TCP connection and to send UDP packets.

The problem of name management within a large space such as the Internet is not easy to solve. For this reason, the Internet was divided into 200 top-level domains that can cover many hosts. Each domain is in turn divided into subdomains that are in turn further divided.

The top-level domains can be generic and by nation. The original generic domains were “com” (commercial), “edu” (educational institutions), “net” (network providers), “org” (non-profit-making organisations), “.gov” (US federal government), “int” (selected international organisations) and “mil” (US armed forces).

Other top-level domains such as “biz” (business), “info” (information), “name” (names) and “tv” (television) were then introduced.

It is usually relatively simple to have a second-level domain of the type “namesubject.it” as a “registrar” (domain registration authority) must be contacted to check if the second-level domain, content of the top-level domain, is available, or if it has already been registered by others. In the event the domain is free, the registration process can be followed, obtaining, in a short space of time, assignment of the desired second-level domain, at an annual fee.

Each domain is named by the path between it and the root. The components are separated by dots (e.g. ing.uniroma1.it to indicate the engineering faculty of SAPIENZA – University of Rome (uniroma1 to distinguish it from other universities of Rome called uniroma2 and uniroma3 in Italy)).

Domain names are “case insensitive”, that is they are insensitive to the fact that the characters may be written in upper case or lower case, providing in any case the same result.

The components of the name can be up to 63 characters long while the full path must not exceed 255 characters. Most organisations of the United States use a generic name while most of those external to the United States use the domain of their own nation.

Each domain controls how to allocate the subdomains. To create a new domain, authorisation of the domain in which the same will be included is required. Once a new domain has been created and registered, the latter can create subdomains without requiring prior authorisation of the structure parts described above.

The name follows the organisation that manages the domain and not the physical networks: entities belonging to the same organisation and physically contiguous can refer to different domains.

A single name server would be able to store the entire DNS database and respond to all requests but this server would be practically overloaded and would not be able to function properly. In addition, a single server would be far too vulnerable as should it malfunction, it would block operation of the entire network. For this reason, the DNS name space is divided into zones that do not overlap where a server dedicated to names belonging to the same areas is used. Usually, each area is equipped with a primary server and one or several secondary servers that receive information from the primary name server. To increase reliability, servers may be located outside of the areas served. The network administrator is the subject assigned to division of the network into areas that also depends on the number of servers in each area.

When a resolver receives a query on a domain name, it questions one of the local servers. If the domain is located within the area of competence of the name servers, it returns the authoritative records of the resources, where an authoritative record is an information file provided by the authority that manages the records, which is therefore always correct contrary to records stored in cache memories that may not be updated. If the domain is remote and the information requested is not available locally, then the name server sends a request message to the first-level name server for the sought after domain. This method of operation is called recursive questioning because each server that does not have the requested information has to look at other servers and then return the information.

1.9.2 Email

Before 1990, electronic mail (email) was mainly used in universities while in subsequent years it experienced a boom in use by the general public. Currently, the number of emails each day is far higher than the number of hard copy letters.

The first email systems comprised simple FTPs, wherein, conventionally, the first line of each message contained the address of the recipient. This system was very primitive and full of limitations and over time more advanced and functional systems have been developed.

Email is composed of two entities: user agents, which allow people to read and send email, and message transfer agents, which move messages from source to destination.

User agents are represented by local programs that provide a service based on messages, menus and graphical interfaces to interact properly with the system.

Message transfer agents are represented by system “daemon”, processes run in the background that move email messages in the system.

Email systems support five basic functions that are as follows:

1. composition;
2. transfer;
3. reporting;
4. display;
5. placement.

Composition represents the process of creating messages and replies.

Transfer performs the moving of messages from the sender to the recipient.

Reporting provides the user with the informational messages on the actual status of delivery, receipt, reading, etc., of email.

Display deals with correctly showing the user the contents of the email, converting appropriately the various formats it contains.

Placement represents the final operation performed by the user following receipt of the message, which may be cancellation, storage, etc.

In addition to these basic services, email systems can also provide other services such as mailing lists that consist of lists of addresses to which is simultaneously sent one and the same message, copies for

information, copies for hidden information, confidential or encrypted mail and alternative recipients if the first is not available.

1.9.2.1 User agents

User agents are programs that use various commands to compose, receive, reply to and organise emails.

To send an email message, after having composed the message itself and having added any attachments, the address in the form "utente@indirizzo.dns" must be added. Most email systems support mailing lists.

There is however a series of problems that can occur in the sending and receiving of messages written in languages with accents, messages in non-Latin alphabets, messages written in languages with ideographic alphabets and messages that do not contain a header but, for example, audio and images. To solve this problem, a solution called multipurpose Internet mail extensions (MIMEs) was devised that is widely used.

1.9.2.2 Message transfer agents

The message transfer agents deal with transmission of emails from the sender to the recipient. To do this, they create a transport connection from the source host to the destination host that they use to transfer the message.

On the Internet, email is delivered by creating a connection between the source host and port 25 of the destination host. A *daemon* is always listening on this port that uses the SMTP system. This *daemon* accepts incoming connections and copies in mail boxes the emails received from these connections. If an email cannot be delivered to the recipient, the sender is sent an error message.

SMTP consists of an ASCII protocol. After a TCP connection on port 25 has been established, the transmitter host, which acts as a client, awaits communication from the receiving host, which operates as a server. The server begins by sending a text string that is used to communicate its identity and the possibility of sending mail: if this does not occur, the client drops the connection and tries again later. If the server is ready to receive the email, the client declares who is trying to email and for whom it is intended: if the indicated recipient exists at destination, the server instructs the client to send the email. Once the client has sent the email, the server sends the acknowledgement. As TCP offers a reliable connection, it is not necessary to inspect the data sent. If there are several email messages, these are sent afterwards, one after another. After the email has been exchanged in both directions, the connection is released.

To solve the problem of communication between users who are not connected, a message transfer agent was then installed on an ISP server, which accepts emails for its customers and stores these emails in mailboxes on the same server. Since the above agent is always connected, the email may be sent and received at any time.

For a user to receive email from the ISP transfer agent, it uses a protocol called Post Office Protocol 3 (POP3), which is activated when the user opens the mail-reading application. The mail-reading application contacts the ISP and establishes a TCP connection with the message transfer agent on port 110. Once the connection is established, the POP3 protocol proceeds with authorisation, which involves login of the user; transaction, which is responsible for receiving mail and deleting it from the ISP and the update, which generates the actual deletion of email. The situations of user always connected and user not always connected are shown in Figure 1.90.

POP3 is suitable for those users who always download their email from the same computer. If several different computers are used, as POP3 causes the deletion of messages received from the ISP, there is the risk of someone's emails being spread over several machines. In this sense, a new protocol called Internet Message Access Protocol (IMAP) was developed that allows the downloading of email without having to delete the ISP folders.

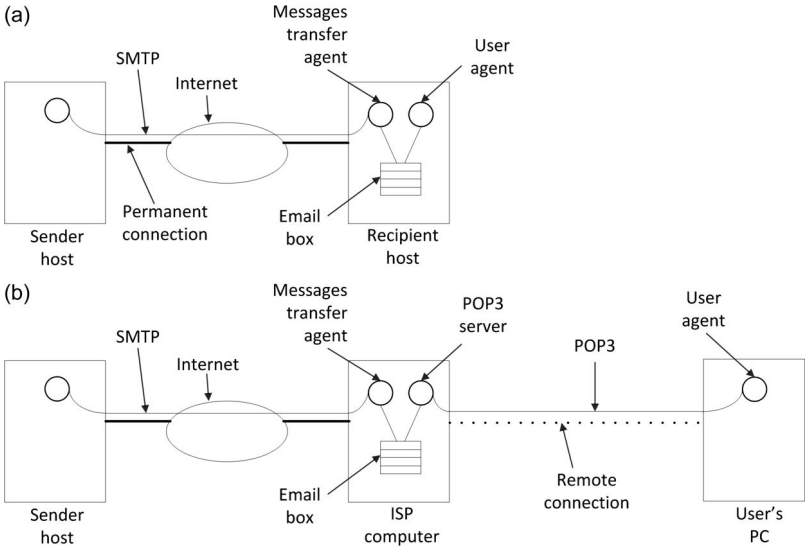


Figure 1.90 Use of email when (a) the user is always connected; (b) when the user is not always connected.

A further email functionality is represented by webmail that allows access to our account via the normal Web pages on the navigator. Once we have entered our account, we can read email received, send new email and delete email in the account.

1.9.3 The World Wide Web

The World Wide Web (WWW) is a system to facilitate access to linked documents that are available on millions of computers connected to the Internet.

Its rapid spread is due to the ease of access and presentation of information on hypertext documents.

WWW was developed in 1989 at CERN (French acronym for European Organisation for Nuclear Research), with headquarters in Geneva, Switzerland. Several groups of scientists from different nations operate at CERN on different experiments regarding particle physics. The various experiments are usually very complex and require years of study, design and construction before becoming operational. To do this, an intense exchange of information, such as documents, photographs and video, even at a great distance, is therefore necessary to reach all members of the various working groups scattered throughout the world.

In March 1989, the physicist Tim Berners-Lee put forward an initial proposal that became operational 18 months later. This proposal allowed the creation of a network of linked documents. In December 1991, this proposal was submitted to the Hypertext 1991 conference that held in San Antonio, Texas. The presentation immediately aroused great interest including that of Marc Andreessen of the University of Illinois that development the first graphic navigator (browser), called Mosaic, which was released in 1993. This browser immediately became very popular and the following year Andreessen created a company called Netscape Communications Corp. whose objective was to develop clients, servers and programs for the Web. This company was subsequently listed on the stock exchange and attracted a flood of capital, arousing great interest in the field from the other big software production companies such as Microsoft that immediately developed an antagonist browser called Internet Explorer characterised by functionality improvements with respect to its direct competitor.

In 1994, CERN and Massachusetts Institute of Technology (MIT) signed an agreement to establish the World Wide Web Consortium (synthesised in W3C) that represents an organisation

focussing on the further development of the Web, the standardisation of protocols and the encouragement of interoperability between sites. The director of this consortium was Berners-Lee. Subsequently, several universities and companies participated, and continue to enter this venture, to be part of this consortium. The consortium, whose web address is www.w3.org, is the place where, par excellence, all the information and latest news concerning the Web can be found.

1.9.3.1 The architecture

The Web appears to end users as a set of Web pages present on all computers connected to the Internet. Each Web page, or in short page, can be connected to other pages contained within other computers.

Users, to follow a link, need simply to "click" above, reaching the relevant page pointed to by the same link. This operating mode refers to the concept of hypertext, already designed in 1945, before the creation of Internet by Vannevar Bush, a lecturer at MIT.

The pages are displayed by the browser (user interface program for navigation on the Internet) that picks them up, on request, by computers or remote servers, interprets the text and formatting commands and displays them on the screen. Pages, for the most part, contain a title, information and usually end with the email address of who created and manages them. The pieces of text that refer to other pages (hyperlinks) are highlighted in another colour, underlined or both. When we hover over a hyperlink, the graphical browsers change the shape of the mouse pointer allowing it to be selected and moving on to display of the "linked" page.

Usually, any links contained in a page and already visited are highlighted with a different colour from the unvisited links, in order to facilitate navigation for the user.

The simplified operational model of the Web is shown in Figure 1.91.

When the user displays a Web page via the browser, if the same "clicks" on a link that refers to a page that is contained on the AAA.com server, the browser sends a message to request that page from the AAA.com server. When that page is received, it appears on the screen. If this page contains a hyperlink to another page on the BBB.com server and this connection is "clicked", the browser performs a similar procedure in respect of the BBB.com server and so on for all the links between linked pages.

1.9.3.2 The client side

If a user selects a hyperlink or link, the browser follows the link and downloads the relevant page. For this reason, the link contained on the page must be able to rely on a method to refer to the others on the Web. The pages are found by using the Uniform Resource Locator (URL). An example of a URL is http://www.aaa.com/index.html. URLs will be explained in more detail below. Each URL is in any case composed of three parts: the name of the protocol (e.g. http), the DNS name of the computer on which the page sought is found (e.g. www.aaa.com) and possibly the name of the file containing the page (e.g. index.html).

When the link above is selected, the following actions take place:

1. The browser determines the URL.
2. The browser questions DNS to obtain the IP address of www.aaa.com.
3. DNS responds with the relevant IP address requested.
4. The browser performs a TCP connection to port 80 on the IP address provided by DNS.
5. The browser sends a request for the file/index.html.
6. The server www.aaa.com sends the file/index.html.
7. The TCP connection is issued.

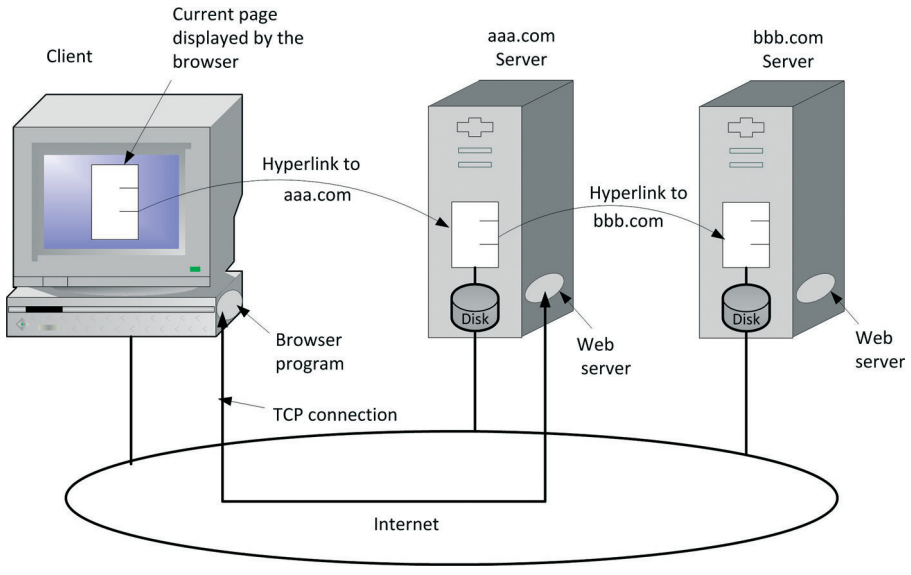


Figure 1.91 Basic model of the Web.

8. The browser displays the text in/index.html.
9. The browser extracts and displays all the images of this file.

Browsers usually display the operation being performed on a status line that is located at the bottom of the screen. In this way, the user can see the ongoing operations and stay informed about any low performance of the network.

It is obvious that, in order to view the pages correctly, a standard language called Hypertext Markup Language (HTML) must be used. A browser is substantially an HTML interpreter, which also offers additional features that make it easier to navigate the Internet.

Web pages, in addition to text, can also contain other types of data such as images, icons, audio, video and photos. Not all pages contain HTML, but the same can be represented by a text in word format, a PDF document, an image in Joint Picture Expert Group (JPEG) format, an audio file in MP3 format, a video file in Motion Picture Expert Group (MPEG) format, etc.: the browser only identifies problems when it is not able to interpret the file format to ensure proper display. If the MIME type does not correspond to one of those built-in, the browser checks its table of MIME types to understand how the page works. This table performs the association between MIME types and their viewers.

The browser can use two methods to be assisted in display: plug-ins and helper applications. A plug-in is basically a form of code that the browser retrieves from a special directory on the hard drive and installs as an extension to itself. This mode is shown schematically in Figure 1.92(a). After the plug-in has done its job properly, it is removed from the browser memory.

The helper application is represented by a full program that is run as a separate process as shown in Figure 1.92(b). As this is a separate program, it does not provide an interface to the browser and does not use the services of the browser but accepts the name of the temporary file, opening it and displaying its content.

The ability to extend the browser with any number of types is very useful but it can have drawbacks in terms of security. In fact, when the browser extracts a file with the extension.exe (as in the case of helpers), it does this with the purpose of installing the alleged helper: this is extremely dangerous as the file that is executed can contain a virus or a malicious code can harm the computer or the data it contains (this mode will be illustrated in detail in the chapter 5 relating to networks security). In any

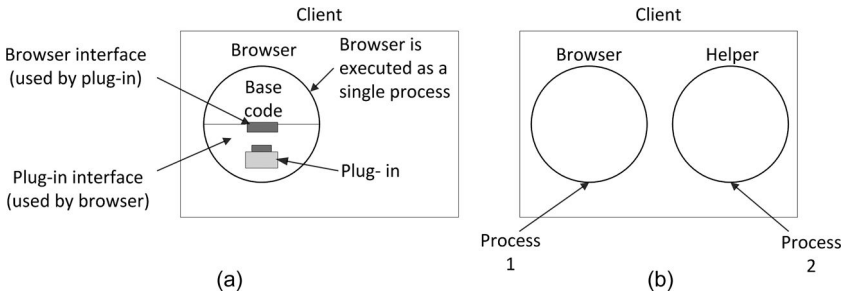


Figure 1.92 Schematisation of (a) plug-in, (b) helper application.

case, the browser can be configured to report an attempt to run unknown programs, ensuring a certain degree of security.

1.9.3.3 The server side

With regard to the server side, this operates according to the following points:

1. Acceptance of a TCP connection from the client's browser.
2. Receipt of the name of the requested file.
3. Removal of the file from our hard drive.
4. Return of the file to the client's browser.
5. Release of the TCP connection.

It is clear that for each request, access to the disc must be performed for recovery of the file and as such the server is not able to meet a number of requests per second more than the number of accesses per second permitted by the hard drive. A standard hard drive is characterised by an average access time equal to 5 ms that reduces to a maximum of 200 the number of requests that can be performed per second, a number too small for a server that is used by an important website.

This can be improved if the server keeps in its cache memory (characterised by a reduced access time with respect to the hard drive) a certain number of files of most recent or more frequent use: before accessing the hard disk, the server checks if the requested file is present in the cache, returning it quickly where this is the case. If the requested file is not present in the cache, the server retrieves it normally from the hard drive.

A faster server can be of multi-thread type. In this type of server, there is a front end that accepts all requests for processing and a certain number of processing modules, as shown in Figure 1.93.

The threads all belong to the same process and, therefore, all the processing modules have access to the cache memory in the process addressing space. When a request arrives, the front end accepts it and prepares a descriptive record that is sent to one of the free processing modules. The processing module responsible controls the cache to see if there is a requested file otherwise it retrieves it from the hard disk, loads it into the cache and returns it to the client. The benefit of this diagram is represented by the fact that while the modules are pending for completion of the operations on the hard disk, other modules can be used to receive other requests. It is clear that, in order to improve performance, the number of hard drives must be greater than one to ensure a high access rate.

Modern Web servers perform a series of other operations. In these servers, the front end sends each incoming request to the first module available, which transports it by executing a portion of the following steps: resolution of the name of the requested Web page, authentication of the client, implementation of access control on the client, implementation of the access control on the Web page, cache control, extraction of the requested page from the hard drive, MIME type determination to be

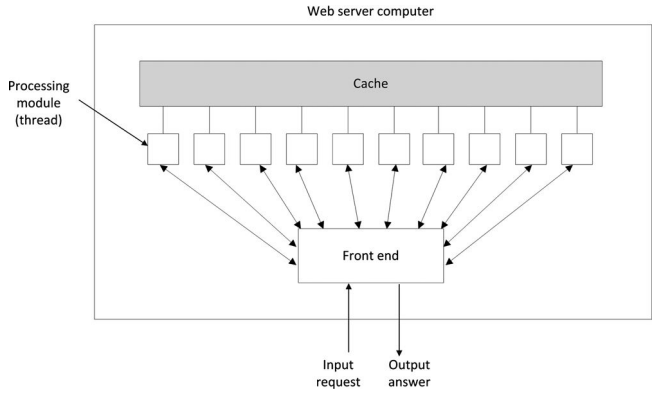


Figure 1.93 Diagram of a multi-thread-type Web server that uses a front end and various processing modules.

included in the response, addressing of various issues, return of the response to the client and creation of a log entry of the server.

If in the unit time too many requests arrive, the microprocessor is not able to meet them all, regardless of the number of parallel disks. To avoid this, other servers can be added with discs replicated in such a manner that the discs themselves do not become an operating limit. This model is called server farm. In this case, the front end receives incoming requests and divides them between several microprocessors, rather than on multiple threads, in order to reduce the load on each computer that can be multi-thread. The operational layout is shown in Figure 1.94.

Server farms are characterised by the lack of a shared cache because each server is equipped with its own memory. To avoid a loss of performance, it is very important that the front-end stores where each request is sent and then always sends requests for that page to the same server. In this way, each server becomes a specialist of certain groups of pages and the cache memory is not misused by dispersion of the pages on the various server caches.

In server farms, the TCP connection ends at the front-end and for this reason, the response must start again from the same front-end, causing a possible bottleneck. To avoid this, the so-called TCP handoff is used that consists of answering directly the processing node concerned. The two situations are shown in Figure 1.95.

It has already been explained that to find every page on the Web, each one is associated with a URL. A URL is composed of three parts: the protocol (also called diagram), the server DNS on which this page is present and a local name that clearly indicates the desired page.

There are obviously various protocols to reach different types of resources and the most common are shown in Table 1.9.

The http protocol is the native language of the Web, that is that most widely used by Web servers.

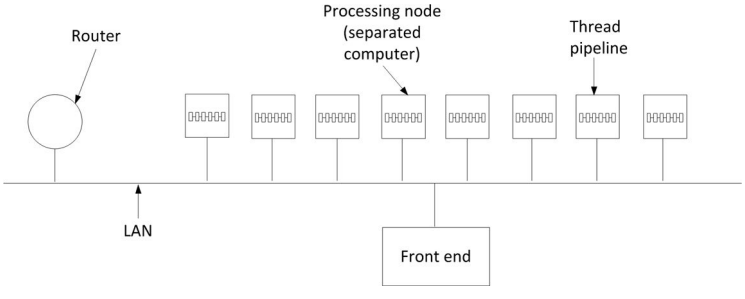


Figure 1.94 Operating layout of a server farm.

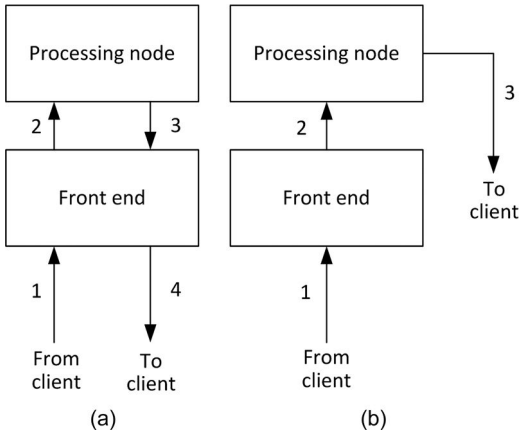


Figure 1.95 (a) Normal response schema. (b) Response schema of the TCP handoff.

The ftp protocol is used to access files with FTP, Internet file transfer protocol. It has been used for a very long time and as such represents a well-consolidated protocol.

The file protocol allows access to a local file on a computer as if it were a Web page.

The news protocol is used to retrieve an article of news as if it were a Web page. It exists in two formats: the first specifies a newsgroup and can be used to receive a list of articles from a predefined site while the second requires the identifier of a specific news article.

The gopher protocol is used by the Gopher system, which takes its name from an athletics team at the University of Minnesota. It was used to retrieve information, in a similar manner to the Web, but limited to text. It is no longer being used.

The mailto protocol allows users to send email from a Web browser.

The telnet protocol is used to establish a connection in line with a remote computer.

The weakness of URL lies in the fact that in any case, it needs to be specified where the desired page is found and not simply to describe the type of page being sought. To avoid this, a new system called Universal Resource Name (URN) is being studied that represents a generalised URL.

In some cases, for websites, the user must be stored, in a transparent manner, in order to present a personalised access page every time that a user visits the same sites, a very useful feature, for example, for e-commerce sites.

There are numerous solutions for resolution of this problem but the most effective and widely used is represented by cookies, a much criticised technique developed by Netscape.

Table 1.9 The more popular and common URLs.

Name	Use	Example of use
http	Hypertext (HTML)	http://. . .. .
ftp	FTP	ftp://. . .. .
File	Local file	file://. . .. .
News	Newsgroup	news:. . .. .
News	News	news:. . .. .
Gopher	Gopher	Gopher://. . ..
mailto	Email sending	mailto:. . .. .
telnet	Remote login	telnet://. . .. . ..

When a client requests a Web page, the server may provide additional information together with the requested page, including a cookie that is a small file or a string with a maximum length of 4 kb. Cookies are stored by browsers in a special directory located on the hard drive of the receiving computer unless the user has disabled this feature.

Cookies are therefore text files but sometimes may also contain viruses and it is not easy afterwards to remove the virus itself. However, if the browser contains a bug (error), this error can be used to remove the virus.

A cookie can contain up to five fields that are domain, path, content, expiry and security. The domain name indicates the origin of the cookie. Browsers should check the veracity of statement of origin. In addition, each client may not store more than 20 cookies coming from the same domain. The path field actually represents a path in the directory structure of the server, which identifies which part of the file structure of the server the cookie can use. The content field takes the form name=value that can be of any value chosen by the server. The expires field in fact indicates the cookie expiration. If this field is absent, the cookie is deleted from the browser upon closure of the connection: in this case, we are talking about non-persistent cookies. If the date and time are designated, the cookie is called persistent and is kept in memory until expiration, referring to Greenwich Mean Time. The secure field can be set to indicate that the browser can only return the cookies to a secure server. This feature is used for electronic commerce and online banking transactions.

Browsers, before sending the request to the server of a certain domain, check if they have in their memory cookies for this domain: if present, they are sent together with the connection request so that the server can use them to customise the answer.

Cookies can also be used by the server to know the number of unique visitors and the actual pages visited. When the first request arrives, it is not accompanied by cookies, and the server responds by also sending a cookie with a counter field set to 1. As the site is gradually visited, the server increments the counter field and sends a new cookie. At the end of the visit, by controlling the counter field of each visitor, it is possible to know the number of pages visited.

Cookies can also be used abusively. In fact, they should be returned only to the site that generated them but hackers (which will be discussed more extensively in the chapter 5 relating to the security of networks) seek to exploit the bugs present in browsers to read cookies not intended for them. In this way, sensitive data such as credit card numbers that are used in e-commerce sites can be read.

Cookies can, unfortunately, be used for the unauthorised gathering of surfing habits of the Internet users, operating in the manner shown below. An advertising agency contacts the main websites to insert ads on their sites, of course paying an appropriate fee. Rather than providing the sites with a file in graphical format, the advertising agency communicates a URL to be inserted on the various pages. When a user visits the pages containing the URL, their browser attempts to reach the URL to download its contents and show the full page on the browser. An advertising image is downloaded from the server of the agency, for which the agency itself receives a fee from the relevant customer, and a cookie containing a user ID with the name of the page visited. When the user visits a second page, perhaps on another site, its browser sends the relevant cookie to the agency, which learns that the same user has visited a new page, and so on during navigation. In the end, the agency is able to reconstruct, through cookies, the profile of navigation for the user and perhaps sell it, unknown to the user, to third parties for the most disparate uses. Because of IP address, it is possible to trace back to the name of the user, thus violating heavily the privacy of the latter.

To avoid this, the receipt of cookies can of course be disabled but this may block some of the features offered by correct and safe sites.

1.9.3.4 Static Web documents

Most of the activity on the Web consists of the transfer of pages from the server to clients. The pages, in their basic form, are static and are located on the servers waiting to be downloaded. From this point of view, images, audio files and videos also represent static pages because they are substantially files.

It has already been said that Web pages are written in a language called HTML. HTML allows the writing of Web pages that contain texts, media files and pointers to other pages. It represents a "markup" language that is a language that communicates how documents must be formatted. If markup commands within the HTML file are incorporated, any Web browser can read them and convert them into the original graphic form, adapting them appropriately to the graphical display format available. In this case, the page can always be displayed on each type of monitor, regardless of the size and resolution.

HTML has been produced in different versions. Version 1.0 was substantially unidirectional, allowing the users to view pages downloaded without, however, being able to send information. Version 2.0 supports the so-called modules that contain boxes and buttons that allow users to enter information or make choices that are subsequently sent to the owner of the pages.

HTML, also in the version that supports modules, provides no structure to Web pages, mixing their content and formatting. To implement this separation, W3C has developed an extension to HTML that allows the structure of Web pages for automatic processing. The first extension, called extensible Markup Language (XML), describes the Web content in a structured manner while the second extension, called extensible Style Language (XSL), describes the formatting independently from the content.

XML can be used for different purposes from the description of Web pages. In this sense, there is a considerable growth of its use as a language of communication between application programs. In particular, Simple Object Access Protocol (SOAP) is very widespread and allows applications to be run apart from languages and systems used. It operates according to the following modes: the client creates the request as an XML message and sends it on to the server, using the HTTP protocol, the server responds with a message formatted in XML, allowing different platforms to communicate with each other.

The most modern version of HTML is extended Hypertext Markup Language (XHTML), which consists of an HTML language reformulated into XML.

1.9.3.5 Dynamic Web documents

The basic model of use of the Web provides that a client sends a file request to a server and the latter recover it from its memory and sends it. At the dawn of the Web, all documents were static, were namely prepared once and for all and stored on hard drives on the server for subsequent retrieval. Over time, the creation of pages on request of users was commenced and there was talk on dynamic Web documents. The generation of these documents may take place both on the server and client side.

With regard to generation on the server side, when, for example, we fill in a form, certain processing will be prompted in order to retrieve the information in the database and to generate a custom page for the same user. To do this, a program or script must be called on the server to process the information provided and to generate an HTML response page. The method that is used to manage the modules and the interactive pages is called common gateway interface (CGI) that consists of a standardised interface that allows Web servers to communicate with final scripts and programs that can accept input data (e.g. those of modules) and generate response HTML pages. Usually, the scripts are written in Perl that represents a programming language that is both easy and fast. By convention, the scripts are located in a directory called CGI-BIN: Common Gateway Interface Binary visible in the URL. A diagram of the operation is shown in Figure 1.96.

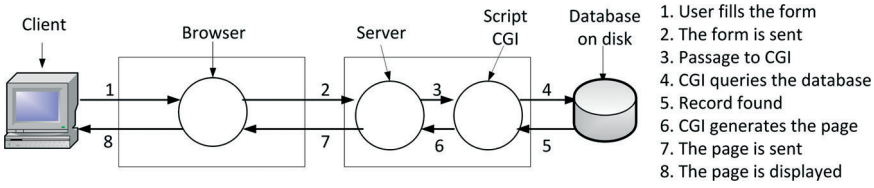


Figure 1.96 Processing sequence of HTML pages

CGI scripts are not the only method for the generation of dynamic content on the server side. Another widely used method consists of the incorporation of a small size script in HTML pages and getting them the server itself to perform them to generate the pages. A language widely used in this sense is Hypertext Preprocessor (PHP). To use this language, the server needs to understand it, as does the browser with pages written in XML. Browsers usually expect that the Web pages containing PHP are characterised by a php extension rather than html or htm. PHP, although easy to use, represents a language that is very powerful and oriented to the interfacing between the Web and a database on server. It represents an open-source code and is therefore available free of charge.

In addition to CGI scripts and PHP embedded code, in order to generate dynamic HTML pages, there are two other techniques called JavaServer Pages (JSP) and Active Server Page (ASP). JSP is similar to PHP with the difference that the dynamic part is written in Java language rather than PHP. ASP is Microsoft's version of PHP and JSP that due to the generation of dynamic content uses a proprietary language called Visual Basic Script. Pages that use this technique are characterised by an asp extension. The set of techniques for generating dynamic content is also called dynamic HTML.

With regard to generation on the client side, it can be said that CGI, PHP, JSP and ASP scripts are able to manage the forms and interact with the databases on servers but are not able to interact directly with the user. To do this, scripts embedded in HTML pages must be used that are performed on the client rather than on the server. The scripting language most widely used on the client side is represented by Javascript that is freely inspired by the Java programming language.

The difference between client-side scripting and server-side scripting is shown in Figure 1.97.

Javascript is not the only way to make the pages interactive. There is also another method that provides for the use of applets, which are short Java programs compiled in machine code that are executed by a virtual computer called Java virtual machine (JVM). Applets can be embedded in HTML pages and interpreted by browsers that support JVM. Since Java applets are interpreted and not carried out directly, the Java interpreter may prevent them from performing operations risky for the security of the computer on which they run. This is true in theory since the authors of applets have discovered a significant number of bugs in Java I/O libraries that can in theory be exploited by potential external attackers.

Microsoft's answer to Java applets is represented by ActiveX controls that are programs compiled in machine language of Pentium microprocessors and executed directly from hardware. This property makes them very fast and flexible because they are capable of carrying out any operation performed by a program. When Internet Explorer finds an ActiveX control, it downloads it, verifies its identity and

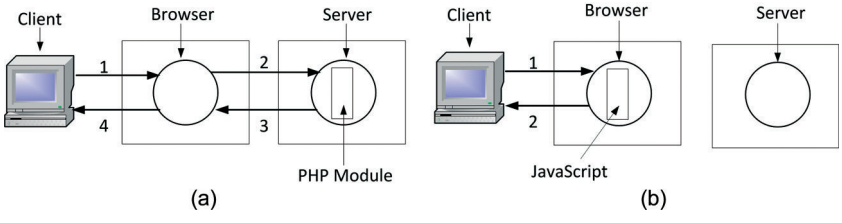


Figure 1.97 Scripting on the side: (a) server with PHP; (b) client with Javascript.

executes it even if this mode of operation can be vulnerability in the security of the computer that performs it.

Javascript programs are easy to write, Java applets are performed quickly and ActiveX controls are carried out even faster. Since all browsers implement the same JVM but not the same version of Javascript, Java applets are characterised by a greater portability than Javascript programs. A summary of what has been seen so far, both on the server side and on the client side, is shown in Figure 1.98.

1.9.3.6 Hypertext Transfer Protocol

The protocol used on the Web is HTTP that indicates which messages the servers can send to the server and vice versa. Every request or response consists of an ASCII file followed by an MIME.

With regard to the connections, the latter are established by the browser using TCP connections to port 80 on the server. Using TCP, neither the client nor the server needs to worry about messages possibly lost, duplicates, long messages or acknowledgement messages.

In version 1.0 of HTTP, after establishment of the connection, a single request is sent and a single response received after which the connection is released. If the Web page is composed of text only, this method is optimal. On the contrary, if the Web page contains icons, images and so on, this method is not efficient and for this reason http 1.1 was introduced that supports persistent connections. In this way, once TCP connection has been established, various requests can be sent one after the other as the relevant responses are received, distributing the setting and the release of the connection itself on many requests, optimising the network traffic.

HTTP was designed for use on the Web and has been made more generic than necessary with a view to future object-oriented applications. Every request sent by http consists of one or more lines of ASCII text in which the first word of the first line represents the name of the requested command. The commands are case sensitive and must be written in upper case. The list of main commands is shown in Table 1.10.

1.9.3.7 Performance improvement

The Web has become widely used and for this reason networks are often overloaded. To minimise delays, several improving techniques have been developed such as caching, replication of the server and content delivery networks.

With regard to caching, the ability to download and retain in memory the most visited pages is exploited, in order to be able to recover them quickly when required. To do this, a process called a proxy is used that stores the pages in a cache memory, ready to be sent on request. In this sense, the browser need not direct its requests to the remote server but directly to the proxy. If the page is already available, it is sent to the browser, otherwise, it is removed from the requested server, stored in the cache of the proxy and subsequently sent to the browser for display. The proxy, on LAN networks, is

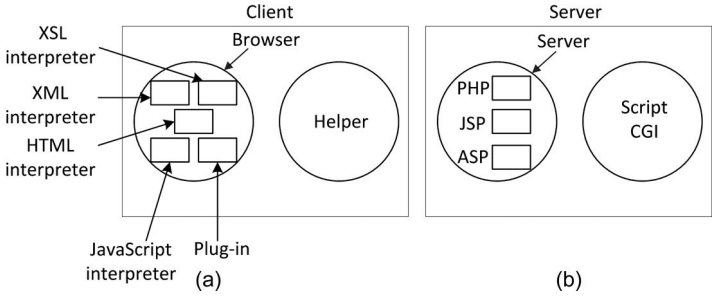


Figure 1.98 Generation mode of Web content on the server side and client side.

Table 1.10 Built-in HTTP commands.

Command	Description
GET	Read request to a Web page
HEAD	Request to read the header of a Web page
PUT	Store request a Web page
POST	Queuing at the indicated resource
DELETE	Removing a Web page
TRACE	Displaying the incoming request
CONNECT	Reserved for future use
OPTION	Interrogation of certain options

performed by a computer shared by all the other computers on the network and can serve several computers at the same time. In many cases, the proxy operates at ISP level for faster navigation on the Internet. In most cases, the local proxy and the ISP proxy operate simultaneously, the local proxy being interrogated first and subsequently the ISP proxy. A setup that covers in sequence several caches is called hierarchical caching an example of which is shown in Figure 1.99.

It is very important is to decide how long pages must remain stored in the cache. If a proxy clears the pages too quickly, it will rarely return an old page but will not be very efficient. Conversely, it would immediately find the requested pages but these would include a number of very old ones. A possible criterion involves checking the date and time of publication of a page, and keeping, correspondingly, this page in memory. In this way, a recent page will be kept in memory for only a short period of time, considering this page variable while an older page will be held in memory for a longer time because this page is considered stable. Dynamic pages should never be stored in the cache because the parameters vary from time to time. In this sense, servers resolve this, with suitable messages, to instruct proxies not to store dynamic pages. In some cases, the so-called proactive caching can be used that involves previously downloading all the pages linked to a requested page in order to have them available for subsequent queries. This technique greatly increases the speed of consultation but at the same time weighs down the network with traffic relating to the download of pages that may never be viewed.

With regard to the replica of the server, there are many techniques to improve performance. The most technique used is to replicate the content of the pages in several distinct positions and far apart, establishing what is called mirroring. The typical case is that of a multinational company with a single site characterised by various links to continental or national sites to which it is quickly rerouted. Mirror sites are usually static. The Web can however suffer the phenomenon called flash crowds, in which, suddenly, a site that is known to be little visited becomes of great interest, experiencing many visits and becoming clogged. In this sense, there must be a mechanism of control that activates auxiliary sites in the event of overload, in order to manage traffic peaks, and that deactivates them once the traffic has

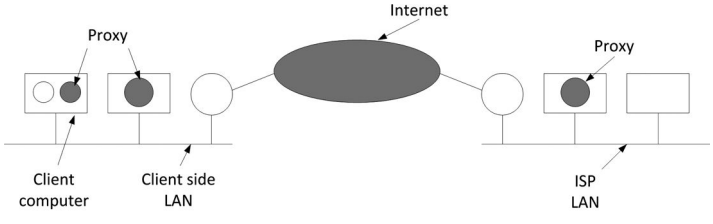


Figure 1.99 Example of hierarchical caching with three proxies.

returned to normal. An even more flexible technique is the creation of dynamic replicas of highly requested individual pages that are placed on servers located in areas from which peak traffic originates.

With regard to content delivery networks, there are companies called content delivery network (CDN), which come to an agreement with content providers (audio, video, newspapers, etc.), committing to deliver such content efficiently in exchange for a small fee. After the agreement, the owner delivers to the CDN the contents of its website for preprocessing and distribution. After this, the CDN contacts various ISPs and agrees to insert a remotely managed server on their LAN, ensuring ISP customers a high response rate. In this sense, the largest CDN have more than 10,000 servers distributed anywhere in the world. In order for this technique to be efficient, requests from users must be rerouted to the closest CDN server.

1.9.4 Multimedia

Multimedia is one of most transmitted forms of content on the Web. Given the relative scarcity of bandwidth still available on wireless systems, the transmission of real-time content that requires a major commitment of bandwidth, such as video, occurs mostly on a wired network.

When multimedia is referred to, what is meant is the combination of two or more continuous supports, that is media that must be reproduced within a well-defined range of time, usually with a certain interaction by the user. The two supports are mostly represented by audio and video. The term multimedia is often used to refer solely to audio but this is not correct because in this case, reference is made to streaming media.

1.9.4.1 Digital audio

A sound wave is practically an acoustic pressure wave that propagates at the speed of sound in space. When this wave enters the ear, it involves a part, called a drum, which, vibrating, causes the subsequent vibration of the small bones that are found within the ear that send electrical nerve impulses to the brain, allowing the perception of sounds. Similarly, when a sound wave involves a microphone, the latter converts it into an electrical signal whose amplitude varies with the amplitude of the sound itself. The representation, storage and transmission of these audio signals are the topic of mainstream study of multimedia systems.

The human ear can perceive sound, the frequency of the latter being between 20 and 20,000 Hz (20 kHz). Certain animals, including dogs and bats, are able to perceive higher frequencies. Auditory perception is logarithmic, for which the ratio between two sounds with powers S_1 and S_2 is expressed in decibels (dB) according to the following formula $10 \log_{10} (S_1/S_2)$.

If the lower limit of audibility (equal to a pressure of 3×10^{-4} dyne/cm^2) is defined for a 1 kHz sinusoidal wave such as a 0 dB, it can be demonstrated that a normal conversation takes place at a level of about 50 dB, while the threshold of pain is around 120 dB. The dynamic range between the threshold of audibility and the threshold of pain is characterised by a factor equal to 1 million.

The human ear is very sensitive to variations in sound that occur in the range of a few milliseconds in contrast to the human eye that does not have the same temporal sensitivity. As a result, a jitter of a few milliseconds during a multimedia transmission has a greater effect on the quality of the sound perceived than on the quality of the image perceived.

Audio signals can be converted into digital format via an Analogue–Digital Converter (ADC). An ADC receives an input electrical signal in the form of voltage and converts it into an output binary stream. To represent an analogue signal as a digital signal, sampling can be performed every NT seconds. If a signal is different from a pure sinusoidal wave but is the linear superposition of sinusoidal waves (practically all periodic signals are), then the Nyquist theorem states that it is sufficient to sample at frequency $2f_{MAX}$, f_{MAX} being the highest frequency present in the signal. Sampling at higher frequencies would be futile since there are no detectable frequencies from the sampling (Figure 1.100).

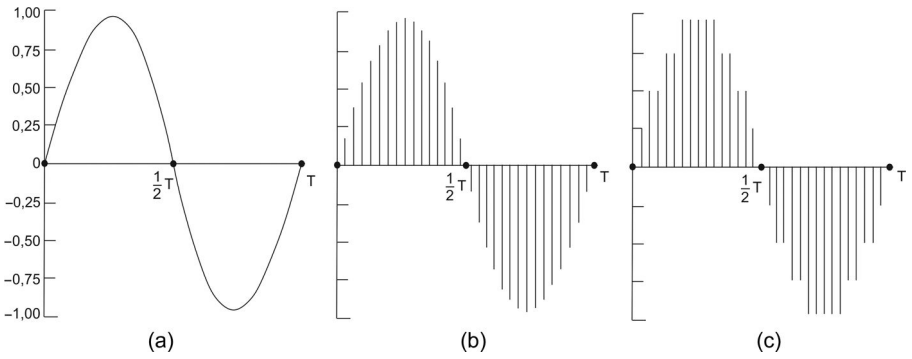


Figure 1.100 Example of (a) sinusoidal wave, (b) sampling of the sinusoidal wave, (c) quantisation of samples with 4 bits.

Digital samples are never accurate. An 8-bit sample permits 256 distinct values while a 16-bit sample allows 65,536 distinct values. When sampling is performed, noise is introduced, called quantisation, equal to the difference between the value of the sample of the analogue signal and the corresponding value calculated from the ADC. If this error reaches excessively high values, the human ear is able to perceive it.

For example, the telephone system uses samples of 8 bits taken 8,000 times per second. All the 8 bits are used in Europe while only 7 data bits are used in the United States and Japan and 1 control bit for which there are 64,000 bps of data in the first case and 56,000 bps of data in the second. If 8,000 samples per second are used, of course frequencies up to 4 kHz can be taken.

In digital audio CDs, sampling is at a frequency of 44,100 Hz for which the maximum frequency is 22,050 Hz. For each sample, 16 bits are used that allow the representation of 16,536 different values that are relatively few compared to the dynamics of a million that can be heard by the human ear and as such, a certain quantisation noise is introduced. An audio CD therefore requires a flow of 44,100 samples per second $\times$ 16 bits per sample $=$ 705.6 kbps for mono audio (single channel) or 1.411 Mbps for stereo audio (dual channel).

1.9.4.2 Audio compression

We have just seen that to transmit an audio stream of fair quality and stereo, 1.411 Mbps are necessary: this value is very high to be transmitted, for example, on the Internet where the band pass is relatively limited. For this reason, compression of the stream must be performed. The compression algorithm most used is MPEG audio, characterised by three levels (or layers) of which the most well-known and used is represented by MPEG audio layer 3 (MP3) that belongs to the audio part of the video compression standard MPEG.

Audio compression can be done in two ways: waveform coding and perceptive coding.

In waveform coding on the audio signal, a Fourier transform is performed. The amplitude of each component in frequency obtained from the processing is encoded with the minimum number of bits in order to minimise the flow.

In perceptive coding, the limits of the human auditory system are exploited, eliminating some of the components, which however does not cause an auditory perception. In this way, the resulting signal is altered and characterised by a flow of lower bits even if the auditory perception is the same as the non-compressed original signal. It is well known in psychoacoustics (which is the science that studies the manner in which human beings perceive sounds) that certain sounds can mask other sounds. In practice, a loud sound on a certain frequency band may mask a weaker sound on another frequency band: this effect is called frequency masking. When the loud sound ends, the ear needs a certain time

interval to adjust the gain and return to hearing the weaker sound: this effect is called temporal masking.

If a large number of people are exposed to a sound at a certain frequency, starting from a very low level and gradually increasing until it is no longer heard and this is done for all the audible frequencies, what is obtained is the so-called audibility curve, characterised by a minimum corresponding to the frequency of maximum sensitivity of the human ear. This curve is shown in Figure 1.101(a).

If, then, in the same sample of persons, the same experiment is repeated, with exposure at the same time to a sinusoidal sound of constant amplitude and frequency, for example at 150 Hz, it can be seen how the curve of audibility is altered and raised in the vicinity of 150 Hz, as shown in Figure 1.101(b).

Performing this last experiment for all the frequencies produces meaningful data on the effect of frequency and time masking that can be used to avoid masked encoding frequencies that would not however be heard by the human ear, significantly reducing the flow of bits needed to encode the sound.

Audio compression can be performed by sampling at 32, 44.1 or 48 kHz, on one or two channels. The output bit rate must also be selected. The samples are processed in groups of 1,152 that correspond to around 26 ms. Each group is made to pass through 32 digital filters obtaining 32 frequency bands. At the same time, the input is sent to a psychoacoustic model to determine the masked frequencies. Thereafter, each of the 32 frequency bands is further transformed to obtain a higher spectral resolution. After this, the number of bits selected in advance in the bit rate is divided among the various bands, dedicating a greater number of bits to the bands with greater non-masked spectral power, a smaller number of bits to the non-masked bands but characterised by a lower spectral power and no bits to the masked bands. The bits are then encoded using Huffman coding that assigns short codes to the numbers that appear often and long codes to the numbers that appear less frequently.

1.9.4.3 Streaming audio

Streaming audio allows listening to sounds on the Internet. It is also called music on demand and represents a possible use of digital audio over the network, together with Internet radio and Voice over IP (VoIP) that will be illustrated later.

The process begins when the user selects a song and the browser becomes active. A TCP connection is first established with the Web server on which the connected song is located. After that, the browser sends a GET request in HTTP to request the song. The server extracts the track from its hard disk and sends it to the browser. Subsequently, the browser downloads the song as a temporary file onto the hard disk, attempts to understand how to play the file using MIME and launches the media player for playback. A diagram of the operation is shown in Figure 1.102.

As the file can be relatively long, resulting in the need to wait for its entire download before starting playback, there are a number of techniques, which will not be described for reasons of space, which allow playback during downloading of the song, avoiding downtime.

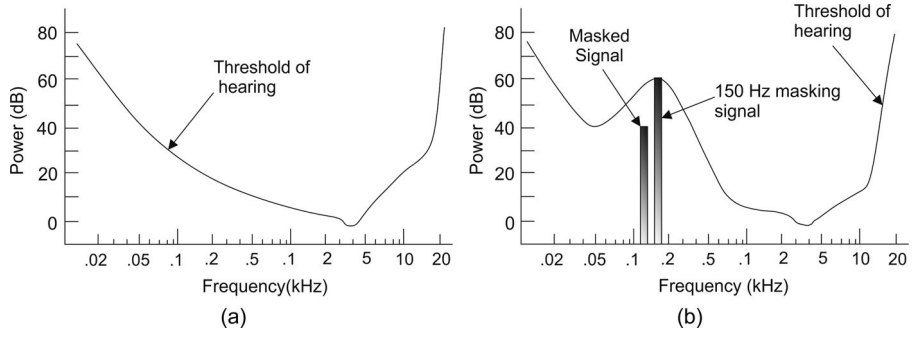


Figure 1.101 (a) Threshold of audibility according to the frequency. (b) Masking effect.

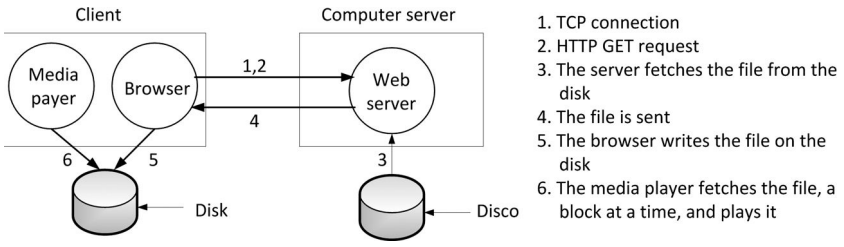


Figure 1.102 Diagram of operation for the reproduction of music in streaming.

1.9.4.4 Internet radio

With regard to Internet radio, there are basically two approaches.

In the first approach, programs are recorded previously and stored on disk and listeners can connect to and download the desired programs.

The second approach involves live direct transmission over the Internet. In this case, to avoid the problem of jitter, data are stored on the buffer of the user at least 10 to 15 s ahead to compensate for irregular delay phenomena, where data are extracted from the buffer.

Streaming audio (first approach) can be sent at a rate higher than that of reproduction since the recipient can stop when the memory limit is reached. In this way, there is the possibility of transmitting lost packets again.

With live radio in the second approach, on the contrary, the generation rate is equal to the speed of playback.

1.9.4.5 Voice over IP

The telephone system was for a long time mainly used for voice traffic and for a minimum of data traffic. In 1999, data bit traffic equalled voice bit traffic while in 2002, the amount of traffic data greatly exceeded the amount of voice traffic as the first has experienced exponential growth while the second, almost flat, is equal to 5% per year.

For this reason, there has been a growing interest in transmission traffic of voice over data networks. In addition, the amount of additional bandwidth required for voice traffic is truly minimised because the packet networks are sized for large amounts of data traffic. Since phone charges are notoriously higher than the rate for access to the Internet, operators of data networks glimpsed the possibility of doing business without having to lay new cables, which then gave rise to Internet telephony also called VoIP.

To avoid a dispersion of energies and an avalanche of different protocols that are perhaps not able to communicate with each other, in 1996, ITU developed the recommendation H. 323 called "Visual Telephone Systems and Equipment for Local Area Networks", which provide a non-guaranteed QoS. This recommendation was revised in 1998 and became the basis for widespread Internet telephone systems.

H. 323 is not a true protocol but rather an overview of architectures of Internet telephony. It refers to various protocols dedicated to voice encoding, call set-up, signalling, the transportation of data and other specific areas without issuing proper specifications. The general layout is shown in Figure 1.103.

The centre of the system is represented by the gateway that connects the telephone system to the Internet, using H.323 protocols on the Internet side and the PSTN on the phone side. The communication devices are called terminals. A LAN connected to the network can be equipped with a gatekeeper that controls the end points within its area of operation.

The set of protocols used in H. 323 for the various services is shown in Figure 1.104.

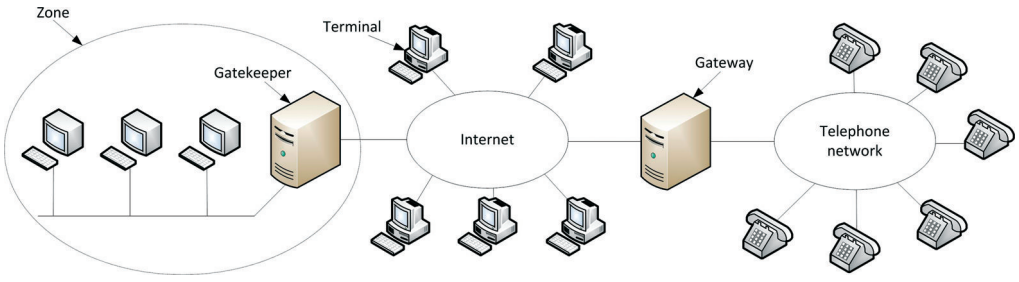


Figure 1.103 Diagram of the H.323 architecture for Internet telephony.

If a terminal wants to call a remote phone, the terminal itself must discover the gatekeeper, transmitting in broadcast a UDP detection packet of the gatekeeper on port 1718. When the gatekeeper responds, the terminal learns the relevant IP address. At this point, the terminal sends the gatekeeper an admission message requesting the bandwidth required. After the band has been assigned, setting up of the call can be commenced. The advance request of the band allows the gatekeeper to check the number of calls, ensuring it does not become overloaded and always ensuring a certain QoS to calls being made. At this point, the terminal begins a TCP connection with the gatekeeper to begin setting up the call. The setting uses the existing protocols of the telephone network that are oriented to the connection and for this reason TCP is required. The telephone system needs something that permits phones to announce their presence and for this reason, both UDP and TCP can be used: the choice fell on UDP due to its lower overhead. Once the band has been allocated, the terminal sends a request message on TCP connection. This message contains the telephone number to be called or the IP address and the port if the request is directed to a computer. The gatekeeper responds with a message of confirmation of receipt of the request and forwards the message to the gateway. The gateway, which is half computer and half telephone switch, makes an ordinary call to the requested phone. The phone rings and sends a message to alert the calling computer to the fact that the phone has started ringing. When the user lifts the handset, the telephone exchange sends a message to indicate to the computer that a connection has been established. After the connection has been established, the gatekeeper is no longer part of the operating cycle unlike the gateway. Subsequent packets bypass the gatekeeper and directly reach the IP address of the gateway. From this point onwards, there is a connection established between the two parties on the physical layer for the exchange of bits. The H.245 protocol is used to negotiate the parameters of the call by using the H.245 control channel that is always open. Each of the two parties begins and declares their own capacities and when these are known to both of them, two unidirectional channels are established, assigning a codec and other parameters to each one. Once everything has been properly configured, the flow of data can start.

<table>
<tr><td>Speech</td><td colspan="4">Control</td></tr>
<tr><td>G.7xx</td><td rowspan="2">R TCP</td><td rowspan="2">H. 225
(RAS)</td><td rowspan="2">Q.931
(call signalling)</td><td rowspan="2">H. 245
(call control)</td></tr>
<tr><td>RT P</td></tr>
<tr><td colspan="3">UDP</td><td colspan="2">TCP</td></tr>
<tr><td colspan="5">IP</td></tr>
<tr><td colspan="5">Data link layer protocol</td></tr>
<tr><td colspan="5">Physical layer protocol</td></tr>
</table>

Figure 1.104 The stack of the protocols in H.323.

When the call is ended, the call signalling channel is used to release the connection and the terminal again contacts the gatekeeper to request the release of the band assigned. The diagram of the logical channels is shown in Figure 1.105.

With regard to the QoS, it is independent of H.323 and depends on the network used: if the latter is able to ensure a connection that is stable and jitter-free from the terminal to the gateway then the QoS will be acceptable. With regard to the phone part, given that the latter uses PCM, it is jitter-free.

It has already been said that H.323 was designed by ITU and is often considered a bulky and complex system characterised by reduced flexibility. For this reason, a new system called Session Initiation Protocol (SIP) was developed that describes how to set up telephone calls via the Internet, videoconferencing and other multimedia connections. It allows the definition of telephone numbers such as URL in such a way that they can be contained within Web pages, allowing calls to be commenced by simply selecting these numbers with the mouse. SIP is able to establish connections between two parties, that is ordinary phone calls, sessions between two parties and multicast sessions, since the sessions can contain audio, video or data. SIP is able to support a variety of different services including identification of the call recipient, determination of the capacity of the call recipient and management of the setting and call ending mechanisms. With regard to telephone numbers, it has already been shown that SIP can support numbers expressed as URL that may also contain IPv4 and IPv6 addresses. SIP is modelled on HTTP on which a party sends an ASCII text message consisting of the command and followed by other lines for passage of the parameters. To establish a connection, the caller creates a TCP connection with the call recipient and sends a message of invitation or sends the invitation message in an UDP packet. If the call recipient accepts the call, they respond with an appropriate HTTP code. Either party may request the end of the call by sending an appropriate message that, acknowledged by the other party, causes termination of the same.

H.323 and SIP have many points in common as well as a number of differences. Both allow calls to two or more parties, using as end points phones and computers. Both support the negotiation of parameters and encryption. H.323 is a telephone industry standard that specifies precisely what is allowed and what is not in the defined protocol stack, facilitating interoperability. For this reason, it is quite a standard complex, rigid and fairly closed to future expansion. SIP, on the contrary, represents a typical protocol for the Internet that is based on the exchange of short lines of ASCII text. It is lightweight and works well with the other IPs but less well with the signalling protocols of the telephone system. A comparison between the two systems is shown in Table 1.11.

1.9.4.6 Video

The human eye is characterised by the property to keep the image that appears on the retina for a few milliseconds and for this reason, if a sequence of images is drawn line by line at 50 images per second, the eye perceives fluid images in the form of scenes in motion rather than a sequence of discrete and separate images. Video systems such as television, therefore, exploit this principle.

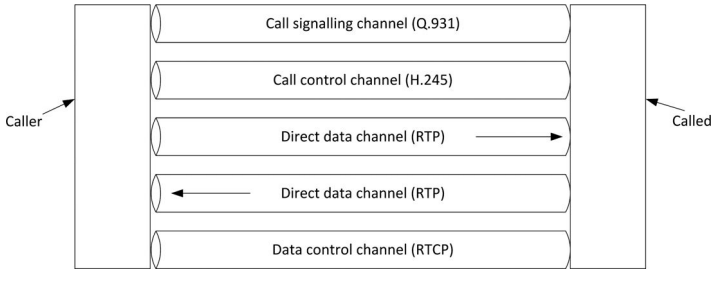


Figure 1.105 Diagram of the logical channels between the two parties during the call.

Table 1.11 Comparison between H.323 and SIP.

Element	H.323	SIP
Compatibility with PSTN	Yes	Largely
Compatibility with Internet	No	Yes
Architecture	Monolithic	Modular
Completeness	Complete protocol stack	SIP handles only the setting
Parameters negotiations	Yes	Yes
Call signalling	Over TCP	Over TCP or UDP
Messages format	Binary	ASCII
Call of several parts	Yes	Yes
Multimedia conferencing	Yes	No
Addressing	Telephone number or host	URL
End of call	Explicit or release TCP	Explicit or timeout
Instant messaging	No	Yes
Cryptography	Yes	Yes
Implementation	Large and complex	Moderate

To represent a two-dimensional (2D) picture with a one-dimensional (1D) signal, the camera performs a horizontal scan, slowly drawing the scan line from the top down while recording the relative intensity of light at each point. At the end of the scan, a frame or frames is/are obtained before returning to the beginning. The receiver synchronises with the transmitter and reproduces, line by line, the original frame. The scanning pattern that takes place in the camera or in the receiving monitor is shown in Figure 1.106.

The parameters used in the scan will vary from country to country. The European system, called PAL, uses 625 scan lines, a horizontal/vertical ratio of 4:3 and 25 frames per second. The system used in America and Japan, called NTSC, uses 525 lines, the same horizontal/vertical ratio and 525 scan

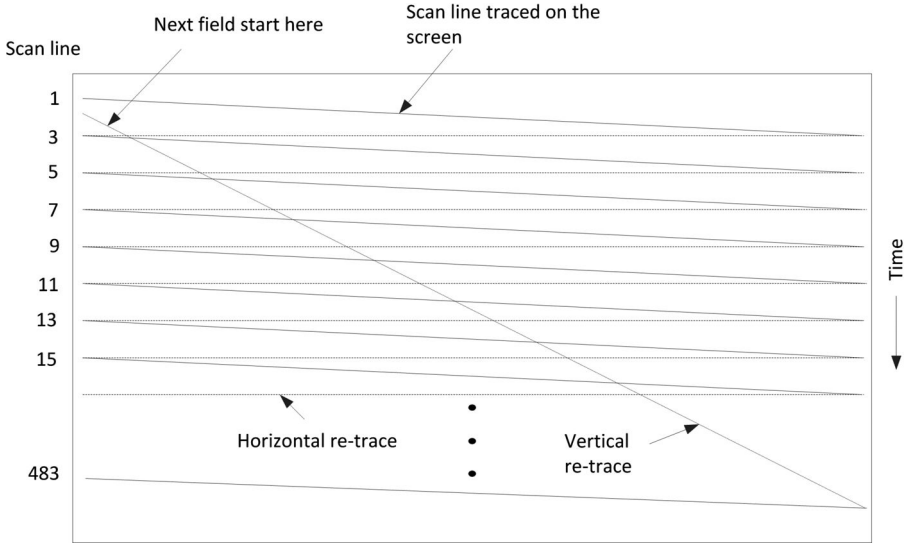


Figure 1.106 Scanning pattern used in cameras and monitors.

lines. In both systems, the first upper and lower lines are not displayed for which, in practice, the number of actual rows decreases to 576 in the PAL system and 483 lines in the NTSC system.

Usually, 25 frames per second is sufficient to avoid the perception of flicker but to prevent the latter being perceived, the system of displaying the scan lines in an interlaced manner is in any case used: a half-frame, called field, containing the even lines and in the next semi-frame the odd lines, increasing by double the number of frames displayed, thus avoiding the effect of flicker. Televisions or video that are not interlaced are called progressive.

The HDTV system is used in the field of television that uses the same principles of scanning as the analogue system with the difference that the image ratio changes from 4:3 to 16:9, to better suit film format.

Video then passed from analogue to digital representation in which each frame is composed of a rectangular grid of elements called pixels. Each pixel can be represented, in minimum quality, using only one pixel that represents its black or white value. If 8 bits are used for each pixel, it is possible to represent 256 levels of grey that ensures a good level of black-and-white image. For colour images, 8 bits are used for each of the three basic colours, thus creating the system of Red, Green, Blue (RGB) representation. This 24-bit representation of colour allows the representation of 16 million different colours, a number that far exceeds the capacity of perception of the human eye. In order to perceive a fluid movement, digital video must use at least 25 frames per second. For Information Technology (I.T.) use, monitors repeat the scan at least 75 times per second, a value that is more than sufficient to avoid recourse to interlaced scanning. In this case, each frame is repeated three times on the video. The most common configurations for monitors are 1,024 × 768, 1,280 × 960 and 1,600 × 1,200 pixels. The lowest resolution, with 24 bits per pixel and 25 frames per second, must be powered at 472 Mbps. To avoid flickering, the storage of frames could be considered and drawing them twice in succession on the monitor. This solution is practicable for digital video but less for analogue video.

1.9.4.7 Video compression

Given the high video stream produced by digital technology, use is almost always made of compression of the same flow. All the compression systems require two algorithms: one for data compression at source and one for the decompression of data at destination. These algorithms are called, respectively, coding and decoding and are mostly asymmetric in the sense that the encoding time (films are encoded once and for all and stored, for example, on a server for distribution) is much greater than the decoding time (compressed film is extracted from the server by the many clients that wish to view it, and then decompress it, in real time). For real-time media files, as in the case of videoconferencing, quick coding is also necessary and for this reason, algorithm parameters vary, significantly reducing the rate of compression.

Another aspect of asymmetry is due to the fact that decompressed video stream is usually slightly different from the original video stream. If this happens, it is said that compression is with loss while, in the case where symmetry is maintained, reference can be made to lossless compression. Compression is performed both on individual frames and on video stream and the respective techniques will be discussed later.

With regard to compression of the frames, the standard most used is JPEG. This standard has four modes of different options. Only the sequential mode with losses will be illustrated below, whose operating diagram is shown in Figure 1.107.

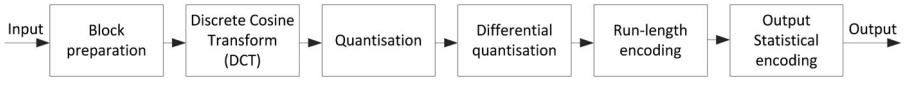


Figure 1.107 Operational diagram of the JPEG algorithm in sequential mode with losses.

To illustrate this mode, we will focus on 24-bit colour images, omitting certain details for the sake of simplicity. Step 1 involves preparation of the block. It is assumed, for the sake of simplicity, that the RGB image has a 24-bit colour 640 × 480 format. Video signal is composed of a Y luminance signal that is responsible for black-and-white vision (grey scale) and two chrominance signals I and Q (in the NTSC system). With the PAL system, the chrominance signals are U and V, which are responsible for colour vision. The relation between the signals Y, I and Q and the RGB signals is as follows: $Y = 0.3R + 0.59\,G + 0.11\,B$; $I = R\,0.60 - 0.28\,G - 0.32\,B$; $Q = 0.21\,R - 0.52\,G + 0.31\,B$. For Y, I and Q, separate matrices are prepared, each with elements in the range from 0 to 255 after which the average of the square blocks of 4 pixels in the matrices I and Q is calculated to decrease them to the sizes of 320 × 240. This reduction is with losses that are not very perceptible to the human eye as the same is more sensitive to luminance with respect to chrominance. This operation allows a reduction by a factor of 2. At this point, 128 is subtracted from each element of all three of these matrices to place zero at the centre of the interval. Then, each matrix is divided into blocks of 8 × 8 sizes. Matrix Y contains 4,800 blocks while the other two matrices contain 1,200 each, as shown in the Figure 1.108.

The second phase of JPEG consists of the application of the discrete cosine transform (DCT) separately to each of the 7,200 blocks. The output of each DCT is represented by an 8 × 8 matrix of DCT coefficients, the average value of the block being the element (0,0) of the matrix, while the other elements represent the spectral power that is present for each spatial frequency. From a theoretical point of view, the DCT is a transformation without losses but the use of floating-point numbers, together with the usual transcendental functions, causes an inevitable rounding error that generates a reduced loss of information. The value of the coefficients decreases, in general, progressively the farther away from the origin you go, as shown in the Figure 1.109.

The third phase, called quantisation, performs elimination of the less important DCT coefficients. This transformation is with losses and is applied by dividing each coefficient in the matrix 8 × 8 DCT by a weight extracted from a table. Of course, if all the weights were 1, the transformation would have no effect. Usually, the weights increase progressively the farther away from the origin you go, in such a way as to quickly eliminate the higher spatial frequencies. A possible example is shown in Figure 1.110, where it is possible to see the initial DTC matrix, the quantisation table and the result obtained by dividing each DCT element by the corresponding element of the quantisation table.

The values in the quantisation table do not belong to the JPEG standard and every time the algorithm is applied, they must be redefined in such a way as to control the quality of loss compression.

The fourth phase reduces the value (0,0) of each block (element in the top left-hand corner) and replaces it with the amount by which it differs from the corresponding element in the preceding block. As these elements represent the average of the respective blocks, they should slowly change and the

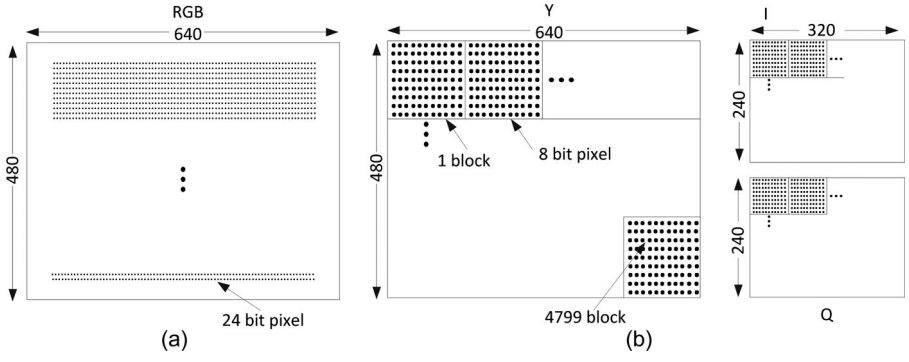


Figure 1.108 Diagram of preparation for JPEG compression. (a) Input data in RGB format. (b) Matrices Y, I and Q after the preparation.

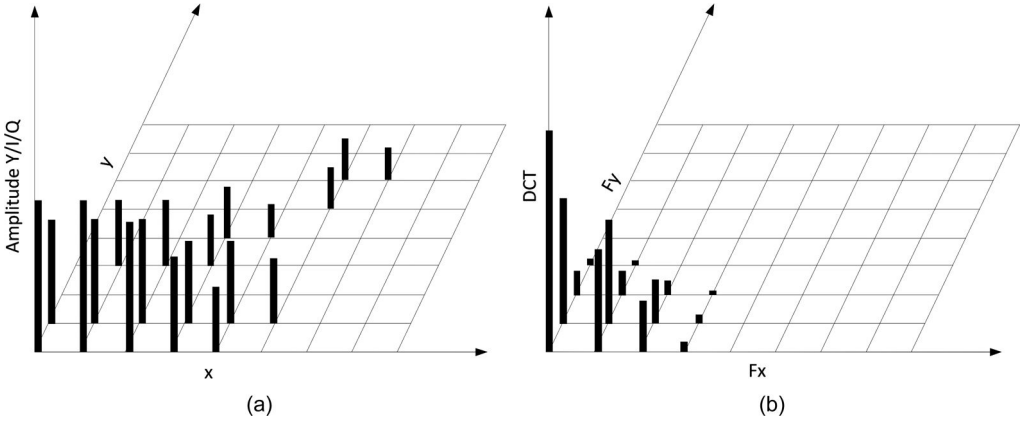


Figure 1.109 Example of (a) block of the matrix Y/I/Q and (b) relevant DCT coefficients.

choice of differential values should reduce them to relatively small values. The values (0,0) are continuous components while the other values are variable components over time.

The fifth step linearises the 64 elements and applies the run-length encoding to the list. Scanning of the block is performed in zig-zag fashion instead of from left to right and from top to bottom, in order to concentrate all the zeros together as shown in Figure 1.111. Final zeroes may be concentrated in a single string that indicates the number of zeros that follow. This technique is called run-length.

At this point, a list of numbers is obtained that represent the image in the transformation space.

The sixth step applies to the numbers obtained Huffman coding for storage or transmission, by assigning shorter codes to more frequent numbers than those that are less frequent.

JPEG due to its compression ability is able to reduce the memory footprint of the image by a factor of 20 and this is why it is intensively used. It is a symmetric algorithm, requiring more or less the same computational effort for both encoding and decoding.

With regard to the compression of video stream, a family of much used standards is represented by MPEG, which was formalised in 1993. Since films can contain both audio and video, MPEG is able to compress both.

The first standard of the MPEG family is represented by MPEG-1. This was developed with the aim of producing in output a video quality similar to video cassette recorders using a bit rate equal to 1.2 Mbps. It is able to ensure factors of compression up to 40 and can be transmitted over short distances on twisted pair. It is also used for storing film on CD-ROM. The second standard of the

DCT coefficients

150	80	40	14	4	2	1	0
92	75	36	10	6	1	0	0
52	38	26	8	7	4	0	0
12	8	6	4	2	1	0	0
4	3	2	0	0	0	0	0
2	2	1	1	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Encoding table

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

Quantisation coefficients

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	2	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 1.110 Example of quantisation: (a) DCT coefficients, (b) quantisation table and (c) quantisation coefficients.

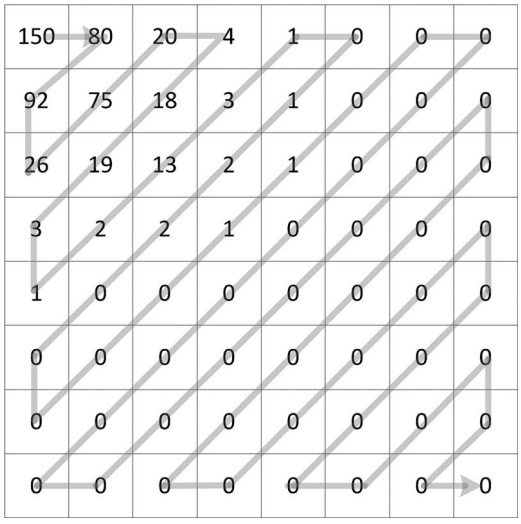


Figure 1.111 Zig-zag encoding.

MPEG family is represented by MPEG-2. At the beginning, it was designed to compress transmission quality video to bit rate between 4 and 6 Mbps, in order to be able to be inserted within a PAL or NTSC transmission channel. Then, it was expanded to be able to ensure higher resolutions, including HDTV. It is very widespread and represents the basic system for digital versatile disk (DVD) and for satellite digital broadcasting. The basic principles of MPEG-1 and MPEG-2 are the same even if they differ in the details.

MPEG-1 is formed of three parts: audio, video and system, as shown in Figure 1.112.

Audio and video coders operate independently and for this reason they need to be synchronised with a 90 kHz clock. MPEG-1 is able to support film with maximum duration of 24 h without overflow.

There are two types of redundancies in film: spatial and temporal. Spatial redundancy can be exploited by encoding each frame with JPEG independently with respect to the others. In this mode, a bandwidth of between 8 and 10 Mbps can be reached. Temporal redundancy can be used by exploiting the fact that the consecutive frames are nearly identical and this ensures a higher rate of compression than the separate compression of individual frames. MPEG-1 output is formed from four types of frames:

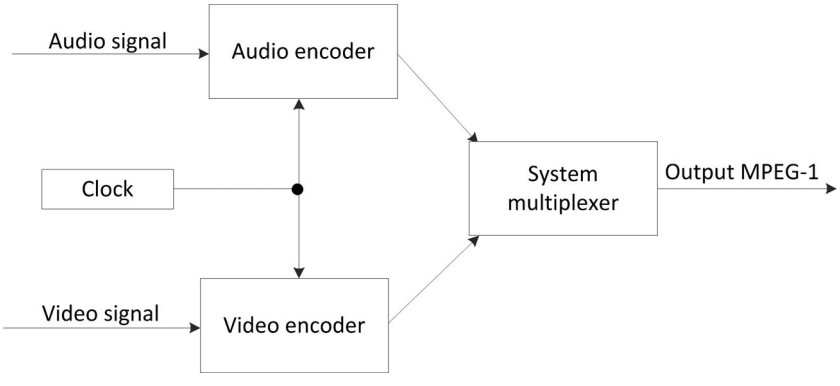


Figure 1.112 MPEG-1: synchronisation of audio and video streams.

1. I frames (intra-coded) that are self-contained static images coded with JPEG;
2. P frames (predictive) that are differences, block by block, with the last frame;
3. B frames (bidirectional) that are differences between the last frame and the next;
4. D frames (encoded DC) that are averages of the blocks and are used for quick winding.

I-frames are simple static images that are encoded using a variant of JPEG that uses full-resolution luminance and half-resolution chrominance on both axes. It is very important that the I-frames appear periodically in the output stream for various reasons. The first is that MPEG-1 is used for multicast transmissions: for this reason, any user can enter into receipt at any time and quickly link into a reference frame to use the following ones. The second is that if an error occurs on a frame received, further decoding would not be possible. The third reason is to allow a fast rewind of frames. For the reasons just described, I-frames are inserted once or twice per second.

P-frames are used to encode the differences between frames. They base their operation on macroblocks that cover 16 × 16 pixels in the luminance domain and 8 × 8 pixels in the chrominance domain. A macroblock is encoded by searching the previous frame or something that is only slightly different. The MPEG-1 standard does not instruct how to search, where to search and the degree of correspondence necessary, leaving these aspects to the implementation. For each position in which a change is searched for, the number of matches in the luminance matrix must be processed. The position with the highest score is considered as long as the same is above a predefined threshold, otherwise the macroblock is considered missing. If a macroblock is found, this is encoded calculating the difference with the value in the previous frame for luminance and for both the chrominances. Such matrices of the differences are subjected to DCT, quantisation, run-length encoding and Huffman coding, in a manner similar to JPEG. The value for the macroblock in output flow is represented by the value of the vector of movement, representing the distance by which the macroblock has moved from its previous position in each direction, followed by a list of numbers with Huffman coding. If the macroblock is not in the previous frame, the current value is encoded with JPEG, in a manner similar to an I-frame. This algorithm is strongly asymmetric.

Since each implementation is free to decide what constitutes an identified macroblock, this freedom has allowed implementers to choose the quality and speed of their own algorithms, always remaining in MPEG-1 context.

Thus far, it has been illustrated how the decoding of I-frames is equivalent to the decoding of JPEG images while the decoding of P-frames requests from the decoder insertion within the buffer of the previous frame and assembly of the current one in a second buffer on the basis of fully coded macroblocks and macroblocks that contain differences compared to the previous frame. At this point, the new frame is assembled macroblock by macroblock.

B frames are similar to P frames, with the difference that they allow the reference macroblock to be both on a previous frame and on a subsequent frame. This freedom allows better compensation of the movement and is very useful when objects pass in front of or behind other objects. To perform coding of B frames, the encoder uses three decoded frames: the previous one, the current one and the subsequent one. Not all implementations use B-frames even if the latter ensure the best compression rate.

D-frames are used to allow the display of a low-resolution image during fast winding as it is already very difficult to perform MPEG-1 encoding in real time: at a much higher speed, it is virtually impossible. Each D-frame represents the average value of a block, without other encodings, in such a manner as to simplify real-time display. D-frames are usually placed before the corresponding frames in such a manner that, with an interrupt to fast winding, display can be recommenced at the normal speed.

At this point, it is appropriate to address MPEG-2. The latter encoding is similar to MPEG-1 as the frames I, P and B are used while D-frames are not used. Another difference is represented by the use of 10 × 10 blocks instead of 8 × 8 blocks in DCT, in order to ensure 50% more coefficients and

production of a higher quality of output video. As MPEG-2 is used in broadcast transmissions and in DVD, it supports both progressive and interlaced images, unlike MPEG-1 that only supports progressive images. MPEG-2, in addition, supports four levels of resolution unlike MPEG-1. They are low (352 × 240), main (720 × 480), high-1440 (1,440 × 1,152) and high (1,920 × 1,080). Low resolution is used for video cassette recorders and for compatibility with MPEG-1; main resolution is the normal resolution that is used for broadcasting; high resolution-1440 and high is used for HDTV. To obtain a video stream of high quality, MPEG-2 is used at 4 to 8 Mbps.

CHAPTER 2

CRYPTOGRAPHY

2.1 Introduction

Cryptography, or encryption, is the science that allows a sender to securely send a message to an intended recipient, without third parties being able to read it.

A message is also called plaintext or cleartext, that is clear or legible text. The message encoding and protection process is also called encryption. An encrypted message is also called ciphertext. The conversion process of the encrypted message in the relevant clear message is called decryption. The chain of the processes described is shown in Figure 2.1.

It has already been said that the science that deals with keeping messages secure is called cryptography while the science that deals with decrypting messages is called cryptanalysis. The sector that deals with both cryptography and cryptanalysis is called cryptology.

Cleartext is usually indicated by M (message) or P (plaintext). It can be represented by a stream of bits, from a text file, from an image, from a video, from a voice stream, etc. A computer treats it as an M binary file. Plaintext can be used both for transmission and storage.

Ciphertext is usually indicated by the letter C. This is also a binary file that can be the same length as M but, in most cases, is characterised by a greater length for the reasons which will be illustrated in the following. In some cases, it is compressed in order to reduce its size. Encryption is performed by applying an appropriate mathematical function E (from Encoding, i.e. coding) to M to produce C, that is in mathematical notation:

$$E(M) = C \tag{2.1}$$

The reverse process of decryption D operates on the ciphertext C to obtain M, namely in mathematical notation:

$$D(C) = M \tag{2.2}$$

Since the whole process of encrypting and decrypting a message must return the original message M, the following mathematical relation must apply:

$$D(E(M)) = M \tag{2.3}$$

In addition to confidentiality, other characteristics are often demanded of encryptions which are listed as follows:

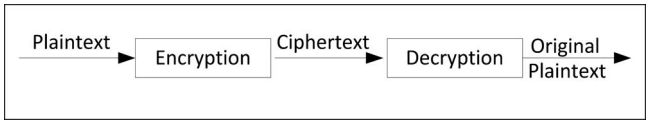


Figure 2.1 Schematisation of encryption and decryption.

1. Authentication: that allows a recipient to know the identity of the sender and does not allow an attacker to assume any identity.
2. Integrity: that allows a recipient to verify that the message has not been altered compared to the original and does not allow an attacker to replace an original message with a false message.
3. Non-repudiation: that does not allow a sender to subsequently deny the sending of a message.

A cryptographic algorithm, also called cipher, is a mathematical function used for encryption and decryption. Even though closely related, the two functions are usually different.

If the security of an algorithm is based on keeping the algorithm secret, this is referred to as reserved algorithm. These algorithms, even if they have a historical interest, are of no practical interest by today's standards as they cannot be used by large groups of users and immediately lose their effectiveness if their operation is revealed. They are still used for low-security applications.

Modern cryptography solves the problem of secrecy of the algorithm by using a key, denoted by the letter K (which comes from Key). This key can take an arbitrary value and the interval within which the key can vary is called the key space. Both the encryption and decryption algorithms use the key and, in this sense, the functions expressed by (2.1) and (2.2) can be rewritten as:

$$E_K(M) = C \tag{2.4}$$

$$D_K(C) = M \tag{2.5}$$

observing in any case a similar equality of (2.3):

$$D_K(E_K(M)) = M \tag{2.6}$$

The situation indicated in the above equation is shown schematically in Figure 2.2.

Some algorithms use different keys K_1 and K_2 for encryption and decryption, allowing (2.4), (2.5) and (2.6) to be rewritten as:

$$E_{K_1}(M) = C \tag{2.7}$$

$$D_{K_2}(C) = M \tag{2.8}$$

$$D_{K_2}(E_{K_1}(M)) = M \tag{2.9}$$

The situation indicated in the above equation is shown schematically in Figure 2.3.

All of the security of these algorithms is based entirely on the key and not on the composition of the algorithm. This means that such algorithms can be disclosed, analysed and used by everyone. It is unimportant that an intruder may learn the algorithm: the important thing is that without a key the message cannot be read.

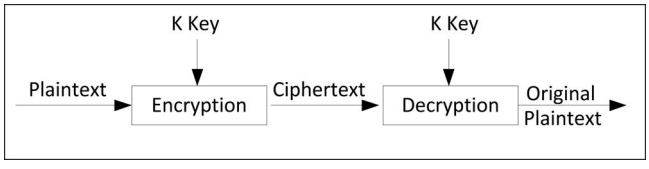


Figure 2.2 Schematisation of encryption and decryption with K key.

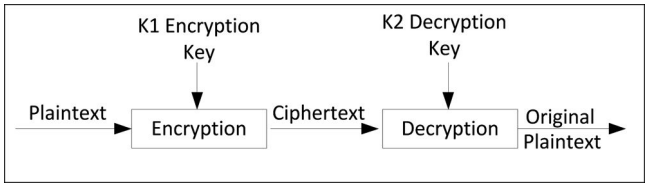


Figure 2.3 Schematisation of encryption and decryption with different keys.

Cryptosystem is the set comprising the algorithm, all the possible plaintext, all the possible encrypted texts and all the possible keys.

There are two general sets of algorithms based on key: symmetric algorithms and asymmetric algorithms or public-key algorithm. In the latter, the encryption key can be calculated from the decryption key and vice versa.

In most of the symmetric algorithms, the key for encryption and decryption are the same. These algorithms, also called secret key, single key or key single algorithms, require that the sender and the recipient agree on the key to be used before starting to encrypt and to communicate securely. The security of a symmetric algorithm is linked to the secrecy of the key used: as long as the communication must remain confidential, the key used must remain secret. Symmetric algorithms can be divided into two categories: the first group, called flow algorithms, work on the plaintext at the level of single bits; while a second group, called block algorithms, works on plaintext at a bit block level. The algorithms currently used operate on typical blocks of the size of 64 bits, large enough to preclude easy analysis and small enough to be processed without great difficulty.

Public-key algorithms, also called asymmetric encryption, are designed to use an encryption key different from the decryption key. These algorithms are called public-key algorithms because the encryption key can be public enabling anyone to use this key for encryption but allowing only the holder of the decryption key (which is only known to the same) to decrypt the message. For this reason, in such systems, the encryption key is also called public key, while the decryption key is also called private key. The private key is also called secret key, but the latter name is avoided in order to avoid confusion with symmetric algorithms.

The purpose of cryptography is to keep secret the plaintext and the relevant key used to protect against possible attackers, also called eavesdroppers, interceptors, attackers, opponents or simply enemies. It is assumed, currently, that intruders can have free access to the channel of communication used between the sender and the recipient.

Cryptoanalysis is the science that allows decryption of an encrypted message without knowing the encryption key. It may also allow identification of the key and can find a weakness of the cryptosystem that permits rapid decryption of the message. A cryptoanalysis attempt is also called attack. A basic assumption of cryptoanalysis, announced for the first time by the Dutchman A. Kerckhoffs in the nineteenth century is that secrecy, must rely entirely on a key, assuming that cryptoanalysts have complete knowledge of the encryption algorithm used and the relevant implementation. Even if in real cases such knowledge cannot be the case, it is wise to assume that this is so: if an attacker cannot successfully conduct his/her attack based on knowledge of the algorithm, there is even more certainty that he/she will not be able to succeed without knowing the key.

There are various types of cryptoanalytic attacks. These assume full knowledge of the encryption algorithm being used. They are as follows:

1. Ciphertext-only attack: in this case, the cryptoanalyst has several encrypted messages all with the same key and must go back to the various plaintexts and to the relevant key to decipher additional messages that use the same key, that is data $C_1 = E_K(P_1)$, $C_2 = E_K(P_2)$, …, $C_n = E_K(P_n)$ to find P_1, P_2, …, P_n or an algorithm to calculate P_{n+1} from $C_{n+1} = E_K(P_{n+1})$.

2. Plaintext attack: in this case, the cryptoanalyst is aware not only of the ciphertext of many messages but also of the plaintext relating to some of the messages. His/her job is to deduce the key used to encrypt messages or an algorithm to decipher each new message encrypted with the same key, that is data P_1, $C_1 = E_K(P_1)$, P_2, $C_2 = E_K(P_2)$, …, P_n, $C_n = E_K(P_n)$ to find K or an algorithm to calculate P_{n+1} from $C_{n+1} = E_K(P_{n+1})$.
3. Chosen-plaintext attack: in this case, the cryptoanalyst has not only access to the plaintext and to the corresponding ciphertext for many messages, but may also choose the plaintext to be encrypted. This type of attack is more powerful than the attack on known plaintext as the cryptoanalyst can choose specific blocks of plaintext to be encrypted and climb with greater ease to the key. His/her job is to deduce the key used to encrypt messages or an algorithm to decipher each new message encrypted with the same key, that is data P_1, $C_1 = E_K(P_1)$, P_2, $C_2 = E_K(P_2)$, …, P_n, $C_n = E_K(P_n)$, where the cryptoanalyst can choose P_1, P_2, …, P_n, to find K or an algorithm to calculate P_{n+1} from $C_{n+1} = E_K(P_{n+1})$.
4. Adaptive-chosen-plaintext attack: this type of attack is a variation of chosen-plaintext attack. In this case, the cryptoanalyst can not only choose the plaintext to be encrypted, but can also change his/her choice on the basis of the previous encryption operation. In chosen-plaintext attack, the cryptoanalyst may choose a relatively large block of plaintext to be encrypted, while in the attack in consideration he/she can choose a smaller block and then select another block based on the results of the previous one and so on.
5. Chosen-ciphertext attack: in this case, the cryptoanalyst can choose different encrypted texts to be decrypted and has access to decrypted plaintext. This type of attack is mainly used for public-key algorithms. His/her job is to deduce the key used for encryption, that is data P_1, $C_1 = E_K(P_1)$, P_2, $C_2 = E_K(P_2)$, …, P_n, $C_n = E_K(P_n)$ to find k.
6. Chosen key attack: this attack does not mean that the cryptoanalyst can not only choose the key but only that he/she has knowledge of the relationship between the various keys.
7. Rubber-hose cryptanalysis: in this case, the cryptoanalyst pursues the direct one concerned with any means to get the key. It is also called key acquirement attack.

Different algorithms provide different levels of security: it all depends on how difficult it is to violate them. If the cost required to violate an algorithm is greater than the value of the encrypted data, they are certainly valid or still, and if the time required to violate an algorithm is greater than the time for which the information must remain protected, they are certainly valid.

There are different categories of violation of an algorithm. They are, in decreasing order of severity:

1. total breakage: in this case, the cryptoanalyst finds the K key in such a manner that $D_K(C) = P$;
2. global deduction: in this case, the cryptoanalyst finds an alternative algorithm without knowing the K key;
3. local deduction: in this case, the cryptoanalyst finds the plaintext of an encrypted message that has been intercepted;
4. deduction of information: in this case, the cryptoanalyst acquires certain information on the key or on the plaintext. This information can be represented by a few bits of the key, from a few sentences of the plaintext, and so on.

An algorithm is called unconditionally secure if, regardless of the amount of ciphertext that the cryptoanalyst has at his/her disposal, there is not enough information to reconstruct the plaintext. All cryptosystems can be cracked by attack on ciphertext-only by simply trying all the possible keys and checking, at each attempt, if the corresponding plaintext obtained assumes a valid meaning. This attack is also called brute force.

An algorithm is called computationally secure, or strong, if it cannot be violated with the available resources, present or future.

The complexity of an attack can be measured in different ways:

1. Data complexity, which represents the amount of data required in input to start the attack.
2. Complexity of the processing, which represents the time needed to perform the attack.
3. Request for storage, which represents the amount of memory needed to perform the attack.

As a rule, the complexity of an attack is calculated by taking the smallest of the above-mentioned factors. Some attacks require a compromise solution between the above-mentioned factors: a faster attack may, for example, require a greater occupation of memory.

The complexity is expressed in orders of magnitude: if an algorithm is characterised by a computational complexity of 2^{64}, then 2^{64} operations will be needed to violate the algorithm itself.

While the complexity of an attack is constant, the computing power increases steadily with the technological evolution. A good cryptosystem must be attacked not only with the current computing resources but also with those of the years to come, at least for a useful period of time.

2.2 General elements of cryptography

2.2.1 Replacement ciphers and transposition ciphers

Before the advent of computers, cryptography was based on algorithms that operated on characters, replacing one character with another or by transposing one character with another. The best algorithms do both of those things several times. Even if these days the complexity has increased, the philosophy remains the same with the difference that the algorithms operate on bits instead of characters with an alphabet of the 26 characters (English alphabet) to one of the two characters (binary alphabet).

In classical cryptography, there are four types of replacement ciphers:

1. Simple replacement cipher or monoalphabetic cipher where each plaintext character is replaced by a corresponding character of the ciphertext.
2. Replacement homophonic cipher, similar to the preceding case with the difference that each plaintext character can map different characters of the encrypted text.
3. Replacement polygram cipher in which blocks of characters are enciphered in a group.
4. Replacement polyalphabetic cipher that is composed of multiple replacement ciphers.

A historic example of replacement cipher is represented by Caesar cipher, in which each character of plaintext is replaced by a corresponding character obtained by sliding to the right or left (encryption key) one alphabet on another and performing direct correspondence.

In transposition ciphers, in contrast to replacement ciphers, the plaintext remains the same, with only the order varying according to the predetermined rules (and keys). A typical case is shown by column transposition simple cipher, in which the plaintext is written in a space of a predetermined width (encryption key): the encryption is performed by reading the text according to the columns obtained. The decryption is performed by writing the ciphertext according to the columns in the same space of predetermined width (decryption key) and reading it according to the lines (Figure 2.4).

In transposition ciphers, since the letters of the plaintext and those of the ciphertext are the same, it is relatively easy to find the statistical correlations between the two texts and to decipher the message.

There are obviously very complex transposition ciphers which, however, thanks to the computing power of modern computers, can be violated within a reasonable space of time.

Currently, transposition ciphers are little used due to both the relatively large memory occupation required and the constraints on the length of the message to be encrypted. In this sense, replacement ciphers are preferred.

Around 1920 various mechanical devices were developed to automate the process of encryption, most of which was based on the rotor concept that was a mechanical wheel used to perform

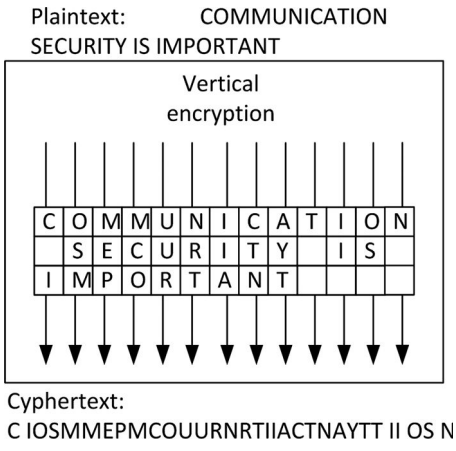


Figure 2.4 Example of a column transposition simple cipher.

replacement operations. A rotor machine consisted of a keyboard and a series of rotors, implementing what is called Vigenère cipher. Each rotor performs an arbitrary permutation on an alphabet of the 26 characters and then replaces the corresponding character in plaintext to obtain a corresponding character in ciphertext. When there are multiple rotors, each character is enciphered with a rotor and a permutation to vary the replacement sequence is then performed. As the rotors move with different speeds, a machine composed of n rotors is characterised by a period of repetition of sequences equal to 26^n, an extremely high number that ensured a high level of security for this type of machine. The most known rotor machine is Enigma, a decryption device used by the Germans during the Second World War.

2.2.2 XOR operation

XOR operation, also called exclusive OR, indicated by the symbol $(+)$, is a standard operation on individual bits defined in Table 2.1.

The following equations are of course valid:

1. $a(+)a = 0$
2. $a(+)a(+)b = 0$

The XOR operation is nothing other than a Vigenère polyalphabetic cipher. It is very useful because encryption can be performed by the XOR operation between the plaintext and the key (which must have a length equal to the plaintext), while the decryption can be followed by performing the XOR operation between the ciphertext and the same key. Mathematically, this can be written as:

1. $P(+)K = C$
2. $C(+)K = P$

Table 2.1 Definition of the XOR function.

Input A	Input B	Output (XOR)
0	0	0
0	1	1
1	0	1
1	1	0

The encryption performed by XOR can be violated, according to the length of *P* and *K*, using the appropriate computing resources.

2.2.3 One-time pad

The one-time pad represents the perfect encryption system. It was designed in 1917. In its classic structure, it consists of using an extensive group of random and non-repetitive characters that are literally fused together with the plaintext, using a 26 modulo sum (the same number as the letters of the English alphabet) between each plaintext letter and each one-time pad letter. It is clear that the one-time pad should be as long as the message that is to be sent. Each key letter is only used once for a single message. The recipient has the same one-time pad (exchanged appropriately and in conditions of maximum security) and performs the same 26 modulo addition between the ciphertext and the one-time pad obtaining plaintext. After receipt of the message the tab is destroyed: if a new one is to be made, a new one-time pad must be generated. For example, if the following message were to be sent:

SECURITY,

using the following table:

FNJUYRAC,

the corresponding ciphertext would be:

WSMPPAVA

as:

(S + F) mod 26 = W
(E + N) mod 26 = S
(C + J) mod 26 = M
(U + U) mod 26 = P
(R + Y) mod 26 = P
(I + R) mod 26 = A
(T + A) mod 26 = V
(Y + C) mod 26 = A

If an intruder does not have access to the one-time pad, this diagram represents a perfectly safe cryptographic system to the extent to which a given encrypted message of a certain length may correspond to any plaintext message of the same length.

In fact, since every character in the table is chosen at random, each key sequence has the same probability and a potential attacker does not have any information with which to encrypt the encrypted message.

It is important to remember that, since each plaintext is equally likely, there is no way for the cryptoanalyst to determine the correct plaintext. A random key sequence added to non-random plaintext generates an entirely random encrypted message and no power of calculation, though high, may vary this.

The other important factor is that the tab must be used only once: even if using a tab of a few Gigabits, if the attacker has a tab in which several characters coincide with the original ones, he/she can decipher the message through subsequent attempts, by gradually checking the characters that take on a certain meaning.

The concept of a single-use tab can easily be extended to thye binary alphabet, precisely by using binary characters (1/0) instead of letters of the alphabet. In this case, an XOR operation is performed

between the bits of the message and those of the table. In order to decipher, an XOR operation between the encrypted message and the tab is performed, re-obtaining the original message.

Although the system described the best and most secure that we can expect from cryptography, there are however limits. In fact, given that the table must be used only once, and that its length must be equal to that of the message to be encrypted, the system is suitable for relatively short message but becomes very difficult to manage when attempting to encrypt communication over a transmission channel of a few megabits per second.

The single-use tab is currently used for ultra-safe channels with reduced bandwidth.

2.2.4 Computer algorithms

There are currently many cryptographic algorithms of which the most popular are described as follows:

1. Data Encryption Standard (DES), which represents the most widespread encryption algorithm until the advent of AES. DES is an international standard. It is a symmetric algorithm because the same key is used both to encrypt and decrypt.
2. Advanced Encryption Standard (AES), which represents the encryption algorithm that replaced DES.
3. RSA (from the name of its developers Rivest, Shamir and Adleman), which represents the most widespread public-key encryption algorithm. It is used for both encryption and digital signature.
4. Digital Signature Algorithm (DSA), used as part of the Digital Signature Standard, which is another public-key algorithm that cannot be used for encryption but only for digital signature.

2.2.5 Introduction to protocols

A cryptographic protocol consists of a series of steps, involving two or more parties, designed to achieve a given purpose.

The protocol is characterised by the following properties:

1. Each subject involved in the protocol must know in advance, *a priori*, the protocol and all the steps to be followed.
2. Each subject involved must agree to follow it.
3. The protocol must not be ambiguous, every step must be well defined, and there should be no possibility of misunderstanding.
4. The protocol must be complete and every action must be specified for every possible situation.

It is now customary, at an international level, to use arbitrary names for the individuals involved in cryptographic procedures. The names are the following:

1. Alice, who represents the first participant in all protocols.
2. Bob, who represents the second participant in all protocols.
3. Carol, who represents the participant in protocols that require a third party.
4. Dave, who represents the participant in protocols that require a fourth party.
5. Eve, who represents the interceptor and eavesdropper.
6. Mallory, who represents the malicious active attacker.
7. Trent, who represents the trusted arbitrator.
8. Walter, who represents the guardian of Alice and Bob in certain protocols.
9. Peggy, who represents the evidence.
10. Victor, who represents the verifier.

In cryptography, an arbitrator is very often used in protocols and procedures. It is very common in civil society (solicitors, banks, etc.), but is a little more difficult to carry this subject in the world of computers for the following reasons:

1. It is easy to find a neutral trusted third party if we know this party, especially from a physical point of view. Two individuals suspicious of one another are led to be distrustful of a third party that they don't know directly, especially on the network.
2. The network must support the cost of maintaining an arbitrator, and this is very difficult.
3. There is always a delay inherent in each arbitration protocol.
4. The arbitrator must intervene in every operation and is a bottleneck for large-scale implementations of each protocol. An increase in the number of arbitrators may reduce this problem but inevitably increases costs.
5. As everyone on the network must have confidence in the arbitrator, it represents a point of vulnerability for those wishing to attack the system.

Despite this, referees are needed and this role is performed by the subject named Trent Figure 2.5.

2.2.5.1 Adjudication protocols

Owing to the relatively high cost of hiring arbitrators, arbitration protocols can be broken down into two lower level protocols: the first is a sub non-arbitration protocol that performed every time the parties wish to complete the protocol, whereas the second is a sub-protocol that performed only in the event of dispute. This special type of arbitrator is called adjudicator, who represents a trusted and impartial third party. Unlike the arbitrator, he/she is not involved in the protocol unless directly called into the matter.

In practice, the first sub-protocol operates according to the following steps:

1. Alice and Bob negotiate the terms of the contract.
2. Alice signs the contract.
3. Bob signs the contract.

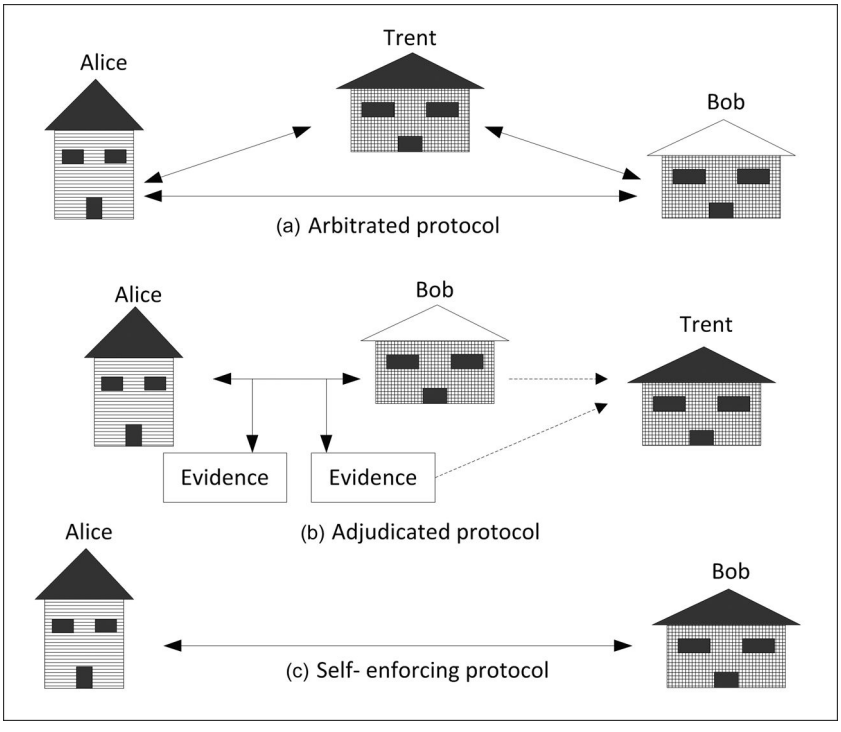


Figure 2.5 Various types of protocols.

In the event of dispute, the second sub-protocol intervenes that operates according to the following steps:

1. Alice and Bob appear before the judge.
2. Alice presents her evidence.
3. Bob presents his evidence.
4. The judge weighs up the evidence.

The difference between an adjudicator and an arbitrator is that the adjudicator is not always necessary as the latter is summoned only in the event of a dispute.

2.2.5.2 Self-reinforcing protocols

The self-reinforcing protocol represents the best type of protocol as it guarantees correctness without resorting to external arbitrators. It is designed to avoid any dispute: if one of the two parties behaves incorrectly, the other party immediately becomes aware of this and the protocol is interrupted.

All the protocols should be self-reinforcing, but unfortunately there is no similar protocol for every situation.

2.2.5.3 Attacks on protocols

Cryptographic attacks can be conducted against the cryptographic algorithms used in the protocols, against the cryptographic techniques used to implement the algorithms and protocols, and against the protocols themselves. In this section only, attacks on protocols are discussed.

Attacks can be passive or active. In passive attacks, a subject not involved in the protocol can intercept some parts or all of the protocol. In this type of attack, the attacker does not interfere in the protocol as its only action is to observe the protocol to try to acquire the information. Since passive attacks are difficult to locate, protocols try to prevent this type of attack rather than intercept it. In this type of attack, the role of an interceptor is performed by the subject named Eve.

In active attacks, the attacker tries to alter the protocol to gain a personal advantage by trying to impersonate someone else, by introducing new messages into the protocol, by deleting existing messages, by substituting a message with another, by sending out new old messages, by interrupting communication on a channel or by altering the information contained in a computer. This type of attack depends on the type of network used.

While the passive attack tries to acquire information about the parties involved in the protocol, by collecting messages exchanged between the parties and trying to decrypt them, the active attack is aimed at acquiring information, by degrading the performance of the system, altering existing information or gaining unauthorised access to resources.

Active attacks are far more dangerous in a particular manner when the parties involved do not trust one another. In this type of attack, the attacker does not need to be an outsider but can be a legitimate user of the system or even the system administrator or several subjects simultaneously. In this case, the role of the attacker is performed by the subject named Mallory.

The attacker can also be represented by one of the parties involved in the protocol, lying during execution of the protocol itself or not following it at all. This type of attacker is also called deceiver. Passive deceivers follow the protocol but seek to obtain more information than the protocol can offer. Active deceivers damage the protocol during its execution in an attempt to deceive.

It is extremely difficult to guarantee the security of a protocol if the majority of the subjects involved are active deceivers even if, in some cases, it is possible to detect this type of situation.

2.2.6 Communication by symmetric cryptography

Symmetric cryptography develops, in a simplified manner, according to the following steps:

1. Alice and Bob agree on the cryptosystem to be used.
2. Alice and Bob agree on the key to be used.
3. Alice encrypts the plaintext which she wants to send using the cryptosystem and the key referred to above, generating an encrypted message.
4. Alice sends the encrypted message to Bob.
5. Bob decrypts the encrypted message with the same algorithm and the same key used by himself and reads it.

Eve intercepts the message sent in step 4 and attempts a ciphertext only passive attack. To increase the likelihood of attack, Eve should intercept the communications that occur in steps 1 and 2 in order to be able to decipher with ease the message exchanged in step 4.

It has already been said that a cryptosystem is of high quality if its security is based on a key and not on the algorithm used. For this reason, key management is of primary importance. If symmetric cryptography is used, the communication referred to in step 1 can be conducted in public, but the communication referred to in step 2 must be carried out in a confidential manner, as the key must remain secret, before, during and after the use of the cryptographic algorithm.

An active attacker (Mallory) could perform additional actions, such as interrupting the communication in step 4: in this case, Mallory could intercept the message sent by Alice and replace it with his own message. If Mallory is able to intercept the communication referred to in step 2, he is able to encrypt his message and send it to Bob in place of Alice's message, without letting Bob noticed the change in sender. If Mallory does not know the key, he may also send a message to Bob which, deciphering it and realising that the same is incomprehensible, may deduce that there was an error in the communication.

Alice also could participate in the process of compromising the secure communication by giving a copy of the key to Eve, in such a manner that the latter subject can intercept to decipher Bob's encrypted communications and Bob can do the same: for this reason, in symmetric cryptography communication, it is assumed that Alice and Bob trust one another.

From what we have seen above, it can be inferred that symmetrical cryptosystems have the following critical issues:

1. Keys must be distributed secretly. In this sense, different communication channels are used with respect to those used for the exchange of encrypted messages.
2. If the security of a key is compromised (burglary, theft, etc.), then Eve is able to decrypt all the encrypted traffic with the key in question. Eve can also exchange her own identity with one of the parties involved to produce false messages aimed at deceiving the other party.
3. Since, in order to ensure maximum security, a key must be employed for each pair of users, the number of keys overall grows rapidly with the number of users themselves as n users require $n(n-1)/2$ keys. This problem can be obviated by reducing, as far as possible, the number of users.

2.2.7 One-way functions

One-way functions provide a fundamental concept for public-key cryptography, since they represent the basic elements for most of the algorithms that are illustrated in the following.

One-way functions are relatively easy to calculate but are extremely difficult to reverse: if f is the function in question and x is the related topic, it is simple to calculate $f(x)$ but difficult to calculate x once $f(x)$ is known.

It is clear that, in itself, a one-way function cannot be used for cryptographic purposes because if encryption can be performed with ease by calculating $f(x)$, it is not so simple to decipher, given the difficulty of tracing x starting from $f(x)$.

There are also a number of one-way functions with service port that allow the performance of reverse calculation once the secret system of doing so is known.

2.2.8 One-way hash functions

One-way hash functions are known by many names: compression functions, contraction functions, message digest, fingerprints, cryptographic checks, message integrity check (MIC) and manipulation detection code (MDC). They represent another fundamental cornerstone of the modern cryptography.

A hash function takes an input string of variable length, called pre-image, and converts it into a fixed-length string, one that is usually less than the input string.

These functions operate in one direction only, since it is extremely simple to calculate a hash value from a pre-image but extremely difficult to calculate the pre-image from the hash value.

A high-quality hash function must be collision-proof, that is it must be extremely difficult to obtain two pre-images from the same hash value.

The hash function is public because the calculation process is not absolutely secret.

In this function, the output may not be related in any way to the input and the change of a single bit in input causes, on average, variation of half the bits in output.

The message authentication code (MAC) is a hash function to which is added a secret code and the hash value calculated is a function of both the pre-image and the key. This function works in a similar manner to the hash function with the only difference being that only those who are in possession of the key can verify that hash value.

2.2.9 Communication by public-key cryptography

Public-key cryptography came about in 1976 was invented by Whitfield Diffie and Martin Hellman and named it as such.

They used two different keys: one public and one private, where it is computationally very difficult to derive the private key from the public key. Any person in possession of the public key can encrypt the message, but only the subject in possession of the related private key can decrypt this message.

The process is mathematically based on a one-way function with service port described previously where encryption, via public key, is represented by direct passage and decryption, extremely difficult for those who are not in possession of the private key, is simplified at the service port that is opened thanks to the private key itself.

Public-key cryptography develops, in a simplified manner, according to the following steps:

1. Alice and Bob agree on the cryptosystem to be used.
2. Bob sends Alice his public key.
3. Alice encrypts the plaintext with Bob's public key.
4. Alice sends the encrypted message to Bob.
5. Bob decrypts the encrypted message sent by Alice with his private key and reads it.

Public-key cryptosystems greatly simplify the key management problem. In fact, in symmetric-key systems, the sender and the recipient must communicate, in secret, the key that they wish to use to encrypt and decrypt, whereas in public-key systems the public key need merely be exchanged on the same channel, not secure, on which the encrypted message will be exchanged, since only those who are in possession of the private key can decrypt the message.

To simplify the process of retrieving public keys, these are usually included in a public database such that anyone wishing to send an encrypted message to a given subject may recover the public key without difficulty.

2.2.10 Hybrid cryptosystems

In everyday applications, public-key algorithms are not used as substitutes of symmetric algorithms in that they are not used to encrypt messages, but only the symmetric keys that will be used for the subsequent encryption. The main reasons are the following:

1. Public-key algorithms are hundreds or thousands of times slower than symmetric algorithms.
2. Public-key algorithms are vulnerable to selected head plaintext attacks.

Hybrid cryptosystems use public-key algorithms for the exchange of session keys according to the following steps:

1. Bob sends his public key B to Alice.
2. Alice generates a random key session K that encrypts using Bob's public key by calculating $E_B(K)$.
3. Bob decrypts the message from Alice using his private key to recover the session key by calculating $D_B(E_B(K)) = K$.
4. Alice and Bob encrypt their communication using the same session key K.

The use of hybrid systems greatly increases the level of security by symmetric key as these are generated from time to time as needed and are destroyed after use.

2.2.11 Digital signature

Manual signature has the following characteristics:

1. Authenticity: the signature certifies that the signatory has deliberately signed the document.
2. Authenticity: the signature certifies that the signatory alone has deliberately signed the document.
3. Non-reusability: the signature is an integral part of a document and a potential counterfeiter cannot move it to another document.
4. Inalterability: after the signature has been affixed, it cannot be altered.
5. Non-repudiability: both the signature and the document are physical objects and as such the signatory cannot later deny the signature.

In reality, this is not entirely true as there are counterfeiters capable of infringing, which are explained above.

To take full advantage of the traditional signature, attempts were made to transfer the same to the world of computers with appropriate adjustments of the case.

2.2.11.1 Digital signing of documents by symmetric and arbitrator cryptosystems

Symmetric cryptosystems allow the digital signature of a document by using a third-party arbitrator.

To do this, there must be a trusted arbitrator, Trent, considered as such by both Alice and Bob. Trent shares a secret key K_A with Alice and a secret key K_B with Bob. These keys are shared beforehand in a confidential manner and can be reused several times.

The whole operation is carried out according to the following sequence:

1. Alice encrypts the message that she wants to send to Bob with the key K_A.
2. Alice sends the encrypted message to Trent.
3. Trent decrypts the message with the key K_A.

4. Trent adds to the decrypted message a statement that the same message was received from Alice and encrypts the whole package with the key K_B.
5. Alice sends the encrypted packet to Bob.
6. Bob decrypts the packet with the key K_B, being able to read both Alice's message and the certification of Trent.

Trent, of course, can verify the identity of Alice from the fact that they have exchanged previously, and secretly, the encryption key K_A.

Now let us see if the whole process referred to above has the characteristics of a classic signature:

1. The signature is authentic since Trent is a trusted arbitrator and he also knows that the message was sent by Alice. The certificate of Trent provides Bob with evidence.
2. The signature is not falsifiable, since only Alice and Trent possess the key K_A. If someone else tries to impersonate Alice, he/she, not possessing the same key, can never encrypt the message correctly and Trent would realise this.
3. The signature cannot be reused as if Bob tried to take the proof of Trent and attempted to add it to another message, Alice could use it. In this case, an arbitrator (Trent or another subject able to access the same information) would request from Bob both the message and the message encrypted by Alice. The arbitrator would encrypt the message with Alice's key K_A and would check that it did not coincide with the message he/she was given by Bob because Bob cannot produce an encrypted message that corresponds to the original one without knowing the key K_A.
4. The signed document is unalterable as if Bob tries to alter the document after having received it, Trent could prove this as stated in the previous point.
5. The signature cannot be repudiated as if Alice were to do this, the certification of Trent would prove the contrary.

If Bob needs to show Carol a document signed by Alice, which is unable to provide her with his secret key, he would need to appeal again to Trent, according to the following points:

1. Bob takes the message and the certification of Trent, he encrypts them with his own secret key K_B and sends them to Trent.
2. Trent decrypts the packet with Bob's key K_B.
3. Trent consults his own archive and certifies that the message was sent by Alice.
4. Trent ciphers the packet with the secret key K_C, which he shares with Carol and sends it to her.
5. Carol decrypts the packet with the key K_C, being able to read both Alice's message and the certification of Trent.

The protocols discussed above are effective but extremely challenging for Trent as the latter must spend most of his time encrypting and decrypting messages and managing his own archive of messages and certification, becoming a true bottleneck in the whole process.

Trent must also ensure the utmost confidence and reliability.

2.2.11.2 Digital signature of documents by public-key cryptosystems

There are a number of public-key algorithms that can be used for digital signature. In some algorithms, such as RSA, both the private and public keys can be used for encryption. If encrypting with our own key, an operation of digital signature is in effect being performed.

The basic protocol operates according to the following steps:

1. Alice encrypts the document with her own private key, signing the document.
2. Alice sends the signed message to Bob.
3. Bob decrypts the document with Alice's public key, ensuring, in practice, the identity of the latter.

The process is extremely fast and immediate and does not require the intervention of an arbitrator. Furthermore:

1. The signature is genuine because when Bob goes to check the document with Alice's public key, he is certain of the identity of the latter.
2. The signature is forgery-proof since only Alice knows her own private key.
3. The signature is not reusable because the signature itself is a function of the document and cannot be transferred to another document.
4. The signature is unalterable because, where this situation arises, the document could not be decrypted with the Alice's public key.
5. The signature cannot be repudiated and Bob does not need Alice's intervention to carry out such verification.

2.2.11.3 Digital signature and stamping of the documents

In some cases, documents must be digitally stamped, inserting date and time of their production and then encrypting all of this appropriately.

2.2.11.4 Digital signature of documents using public-key cryptosystems and one-way hash functions

In practical applications, public-key algorithms are too slow to sign very long documents. To reduce signature times, one-way hash functions are often used. In this case, instead of signing the whole document only the hash of the document is signed. In this sense, the procedure develops according to the following steps:

1. Alice computes the hash of the document that she wants to sign digitally.
2. Alice encrypts the message with her private key, signing, in fact, the document.
3. Alice sends Bob the document and the signed hash of the document.
4. Bob computes the hash of the document that Alice has sent and then decrypts the hash of the document sent by Alice and compares them: if they are equal, the digital signature is considered valid.

In this case, the speed of the process is considerably higher because the probability that two documents having the same hash of 160 bits (typical value of the algorithms that are currently used) is 1 in 2^{160}; fidelity of the digital signature with that of the hash of the document can be reasonably compared.

This protocol is very useful for a number of reasons:

1. The signature can be kept separate from the document.
2. The occupation of memory required by the document and the relative signature are reduced to a minimum.
3. An archiving system can automatically check documents without having to necessarily store them.
4. A central archive should only store the signed hash of the documents and not the documents themselves.
5. There is no need to send the entire document for the stamping operation, only its hash.
6. In the event of dispute over the identity of the creator of the document and the relevant date and time, the central archive need merely compute the hash of the document, allowing a subject to demonstrate the intellectual property of a document while keeping it confidential.

2.2.11.5 Algorithms and technologies

There are currently many algorithms for digital signature. In most cases, the signature process is called private-key encryption, while the verification process is called public-key decryption: this can be misleading and it is in any case valid only for the RSA algorithm, which was mentioned previously and which will be discussed more extensively in the following, as different algorithms can have different implementations.

In general, the process of signing with a key K is referred to as $S_K(M)$, while the signature verification process using the corresponding public key is indicated as $V_K(M)$.

The string of bits added to the signed document is called digital signature or simply signature. The entire protocol through which the message recipient verifies the identity of the sender and the integrity of the message is called authentication.

2.2.11.6 Multiple signatures

To affix multiple signatures on a document without recourse to the hash functions, there are two possible solutions:

1. Alice and Bob sign separately the document concerned obtaining two copies of the same, each with their signature.
2. Alice signs the document first and then Bob signs Alice's signature. In this case, it is impossible to verify the signature of Alice without having to verify the signature of Bob.

If using the hash functions, the process of multiple signatures is divided into the following steps:

1. Alice signs the hash of the document.
2. Bob signs the hash of the document.
3. Bob sends his signature to Alice.
4. Alice sends the document, her signature and the signature of Bob to Carol.
5. Carol verifies both the signature of Alice and the signature of Bob (this process can be performed separately for each signature).

2.2.11.7 Non-repudiation and digital signatures

Any subject may, at any time, find an expedient to deny having digitally signed a document. In fact, he/she could sign the document and then intentionally lose his/her key, saying that someone else has signed in his/her place. This action of denial of affixing a signature is called repudiation.

The stamping operation may reduce the occurrence of this risk, even if the subject repudiator can always declare that he/she has lost the key before the stamping operation.

There is, however, a protocol to avoid invalidating signatures affixed on a date prior to loss of the key. This is structured according to the following steps:

1. Alice signs the document.
2. Alice generates a header containing certain identifying information that concatenates with the document and signs everything, sending it to Trent.
3. Trent verifies the external signature and confirms the identifying information. He adds a stamp to Alice's document and to its information header, signing it all, and sends it to both Alice and Bob.
4. Bob verifies the signature of Trent, the identifying information and the signature of Alice.
5. Alice verifies the message sent by Trent to Bob. If she did not generate the message, the same is able to complain to the appropriate authorities.

2.2.12 Digital signatures with encryption

If the digital signature is combined with public-key cryptography, it is possible to develop a protocol that exploits the security of cryptography with the authenticity of the digital signature. The protocol develops according to the following steps:

1. Alice signs the message M with her private key, performing the operation $S_A(M)$.
2. Alice encrypts the message with Bob's public key B, performing the operation $E_B(S_A(M))$.
3. Bob decrypts the message with his private key, running the operation $D_B(E_B(S_A(M))) = S_A(M)$.
4. Bob verifies using Alice's public key and retrieves the message, performing the operation $V_A(S_A(M)) = M$.

2.2.12.1 Attacks against public-key cryptography

The best way to obtain someone's public key, avoiding any attacks, is to resort to a public and trusted database. This database should be write protected by all the subjects except Trent, otherwise a malevolent subject like Mallory could replace Bob's public key with his own, substituting with his own identity.

Even if public keys are stored in a trusted database, replacement is always possible, by Mallory, during transmission. To avoid this, Trent can sign each public key with his own private key, generating a key certification authority or key distribution centre (KDC).

In practical implementations, the KDC signs a message consisting of the user's name, their public key and any other useful information about the user and makes it available to users who have requested it. When Alice needs to retrieve Bob's key, she verifies the signature of the distribution centre of the keys to ensure its validity.

This does not of course make an attack by Mallory impossible, but it does make the situation more complicated.

2.2.13 Generation of random or pseudo-random sequences

Random number generators are very important in cryptographic applications and for this reason are described in this section.

Unfortunately, there are generators in most computers that are only formally random and, in this case, they are called pseudo-random. In fact, these generators appear to be random for reduced numbers of sequences but for long sequences of numbers correlations begin to appear which make them inevitably pseudo-random and therefore unsuitable for valid cryptographic applications.

Unfortunately, all the finite state machines, which are computers, tend to generate periodic sequences that are repeated after relatively long periods.

In a perfectly random binary sequence, the number of zeroes must be equal to the number of 1 and their arrangement must be free of correlations.

A generator is called pseudo-random if it seems to be perfectly random and passes all the statistical tests for randomness.

In all computers, generators are pseudo-random with very long repetition periods which can reach 2^{256} bits.

Pseudo-random sequences for cryptographic use must enjoy the property of being unpredictable and where it is computationally impossible to predict the value of the next random bit starting from the sequence of previous bits and from the knowledge of generation algorithm.

Random sequences are called real if they cannot be reproduced reliably and if the respective generator, starting from the same point and in the same initial conditions, produces a different sequence each time.

2.2.14 Exchange of keys

A commonly used technique is to encrypt each individual conversation with a separate key. This key is known as a session key since it is used exclusively for every communication session. They are very useful because, as seen previously, their existence is linked to the existence of the communication. The problem is represented by the difficulty of getting the session safely to the communicators.

2.2.14.1 Exchange of keys using symmetric cryptography

This mode assumes that Alice and Bob use a network where there is a KDC, represented by Trent. These keys must be available before the start of the protocol. The protocol is performed by following the following steps:

1. Alice contacts Trent and requests a session key to communicate with Bob.
2. Trent generates a random session key and encrypts it into two copies using the key of Alice and Bob, respectively.
3. Alice deciphers her copy of the session key.
4. Alice sends Bob her session key.
5. Bob deciphers her copy of the session key.
6. Alice and Bob use this session key to communicate securely.

The security of this protocol is based on the security of Trent, generally represented by a network-connected computer. If Mallory is able to bribe Trent, the entire security is jeopardised, as Mallory may have access to all the users' keys, being able to read all the old communications that have been recorded as well as future communications.

Moreover, Trent is the bottleneck of the entire system: just one of his malfunctions would undermine all the communications of the system itself.

2.2.14.2 Exchange of keys using public-key cryptography

The basic hybrid cryptographic system has already been previously discussed. In this situation, Alice and Bob use cryptography to agree on a session key and use the session key to encrypt data. In some cases, Alice and Bob use a KDC where they are registered. In this way, each user can communicate with any user with whom they have never communicated before.

The protocol works according to the following steps:

1. Alice retrieves Bob's public key from the KDC.
2. Alice generates a session key, she encrypts it using Bob's public key and sends it to the latter.
3. Bob decrypts the message from Alice using his private key.
4. Alice and Bob start to communicate in a secure manner using the session key exchanged.

2.2.14.3 Man-in-the-middle attack

While Eve can only try to violate the public-key algorithm or attempt an attack on ciphertext only, Mallory can not only listen to messages of Alice and Bob, but can also edit messages, delete messages and generate completely new messages. Furthermore, he can impersonate Alice in her dealings with Bob and vice versa.

The attack sequence takes place according to the following steps:

1. Alice sends Bob her public key; Mallory intercepts this key and sends Bob her public key.
2. Bob sends Alice his public key; Mallory intercepts this key and sends Alice his public key.

3. When Alice sends a message to Bob, actually using the public key of Mallory, the latter is able to decrypt it using his private key, read it, encrypt it with Bob's public key and send it to the latter.
4. When Bob sends a message to Alice, actually using the public key of Mallory, the latter is able to decrypt it using his private key, read it, encrypt with Alice's public key and send it to the latter.

Even if we resort to a database of external keys, Mallory can always intervene at the moment of communication of keys, replacing them with his public key and can even attack the database to replace the keys.

The attack in question is successful because Alice and Bob have no way of verifying that they are communicating with one another. If Mallory is so skilful such that no delay in communications is caused, Alice and Bob do not have the slightest suspicion that there is a third entity present in the midst of their conversations that is intercepting everything.

2.2.14.4 Interlock protocol

Interlock protocol represents a good system to prevent a man-in-the-middle attack.

This protocol works according to the following steps:

1. Alice sends Bob her public key.
2. Bob sends Alice his public key.
3. Alice encrypts the message using Bob's public key. She only sends half of the encrypted message to Bob.
4. Bob encrypts the message using Alice's public key. He only sends half of the encrypted message to Alice.
5. Alice sends the remaining half of the message to Bob.
6. Bob joins the two halves of Alice's message and decrypts it using his private key.
7. Bob sends the remaining half of the message to Alice.
8. Alice joins the two halves of Bob's message and decrypts it using her private key.

It is clear that it is not possible to decipher the message if the two halves are not available.

Let us see how such a system can hinder the activity of Mallory. He can still replace the public keys of Alice and Bob with his own key, but when he intercepts half of Alice's message, he cannot decrypt it with his private key or encrypt it with Bob's public key, as it is incomplete. The only thing he can do is to invent a completely new message and send half of it, encrypted with Bob's public key, to the latter. He can do the same for Alice. When the second half of the messages arrives, it is too late to change the messages that the same has invented and previously sent and the conversation between Alice and Bob becomes totally different.

Mallory could in any way conduct his attack if he knows Alice and Bob well enough in such a way as to be able to reproduce their conversation, but this is still more difficult to perform than the simple man-in-the-middle attack.

2.2.14.5 Exchange of keys via digital signature

The use of digital signature during a session key exchange can prevent a man-in-the-middle attack. In this case, use is made by Trent who signs the public keys of both Alice and Bob. The signed keys include a signed certificate of ownership. When Alice and Bob receive the keys, each verifies the signature of Trent. In this way, they are sure that each public key actually belongs to the other person and the key exchange protocol can continue.

In this case, Mallory has significant difficulties attacking. He cannot, for example, assume the identity of Alice or Bob as their keys are signed by Trent. All he can do is to listen to the encrypted traffic and damage the relevant line of communication.

This protocol, even if it uses Trent, has a reduced risk of compromising the capabilities of the KDC with respect to the first protocol. In fact, if Mallory is able to bribe Trent, all he can do is to acquire the private key of Trent that allows him to sign new keys but does not allow him to decipher each session key or read any message. To read the traffic, Mallory must impersonate a user on the network and try to legitimise himself by encrypting messages with his false public key. In fact, thanks to Trent's trusted key, Mallory can create false signed keys to deceive Alice and Bob which can be exchanged in the database for valid keys or even to intercept requests from users to the database and respond with his false keys, making a man-in-the-middle attack possible.

This type of attack can occur but do not forget that Mallory should be able to intercept and modify the messages. In most networks, it is much simpler to listen passively to messages that pass through rather than actively operate on them. In broadcast channels, such as radio networks, it is practically impossible to replace a message with another even if the entire network can be disrupted by actions of jamming. In computer networks, this is easier, as will be seen in the chapters 5, 6 relating to the security of networks.

2.2.14.6 Transmission of keys and messages

Alice and Bob do not need to complete the key exchange process before exchanging messages. In the protocol shown in this section, Alice sends Bob a message M without any prior exchange of keys, according to the following steps:

1. Alice generates a random K session key and encrypts M using K. She completes, that is to say, the operation $E_K(M)$.
2. Alice takes Bob's public key from the relevant database.
3. Alice encrypts K with Bob's public key B, performing the operation $E_B(K)$.
4. Alice sends both the encrypted message $E_K(M)$ and the encrypted key $E_B(K)$, and to increase the level of protection against a man-in-the-middle attack Alice may also sign the transmission.
5. Bob decrypts the message from Alice using his private key.
6. Bob decrypts the message from Alice using the session key.

This hybrid system shows how public-key cryptography is used in communication systems, but may, in any case, be combined with other techniques, such as digital signatures and stamping to increase the level of security.

2.2.14.7 Transmission of keys and messages

In some cases, the need may arise, on the part of a subject, to send an encrypted message to several subjects. In the protocol, given in this section, it is shown how Alice can send an encrypted message to Bob, Carol and Dave, according to the following steps:

1. Alice generates a random K session key and encrypts the message using K. She completes, that is to say, the operation $E_K(M)$.
2. Alice retrieves the public keys of Bob, Carol and Dave.
3. Alice encrypts K with Bob's public key, encrypts K with the public key of Carol, encrypts K with the public key of Dave, that is to say the operation and $E_B(K)$, $E_C(K)$ and $E_D(K)$.
4. Alice passes the encrypted message and all the encrypted keys to anyone who wants to receive them, transmitting $E_K(M)$, $E_B(K)$, $E_C(K)$ and $E_D(K)$.
5. The only ones capable of decrypting the key K and deciphering the message M are Bob, Carol and Dave through their private keys.

A central server, not necessarily trusted and secure, can pass the message to Bob, Carol and Dave together with the related encrypted keys.

2.2.15 Authentication

The typical process of authentication on a remote server takes place by providing a username and a password so that the server itself knows the identity of who is accessing it: both the user and the remote server share security information (username and password) to allow access to the server itself.

2.2.15.1 Authentication using one-way functions

Using one-way functions allows the server to distinguish valid passwords from those that are not, without the latter knowing all the passwords. This can be done according to the following steps:

1. Alice sends the remote server her password.
2. The remote server calculates a one-way function on the received password and compares the result with that in its possession previously stored.

Since the remote server does not store the passwords of all enabled users, the risk of a breach of the same and the theft of passwords of users becomes negligible, since in the same are stored only the results of one-way functions of passwords that do not allow returning to the original password.

2.2.15.2 Dictionary and salt attacks

A password file encrypted with a one-way function still remains vulnerable as Mallory is able to prepare a complete list of all the most common passwords and calculate their one-way function. Mallory can acquire the encrypted password file fraudulently on the server and compare it with the file that he himself has developed and see the degree of similarity. This type of attack is called dictionary attack and has great potential to become a reality. To make this type of attack more difficult, the so-called *salt* is used that represents a random string that is added at the end of each password before the calculation of the one-way function. If the number of possible values of *salt* is great enough, a dictionary attack can be predicted as Mallory should calculate the one-way function for each possible *salt* value.

The use of *salt* is not a universal security solution in that it only protects from a dictionary attack on the file of the password, without protection from a targeted attack directed towards individual passwords. It protects only those who have the same password on multiple remote hosts.

2.2.15.3 Authentication using public-key cryptography

When Alice sends her password, anyone with access to the communication channel can read it. This problem can be avoided by using public-key cryptography. To do this, the remote server has a file of all the public keys of users and of course all users have their own private key.

A simple authentication process could develop according to the following steps:

1. The remote server sends Alice a random string.
2. Alice encrypts that string with her private key and sends it to the server along with her name.
3. The remote server retrieves from its database Alice's public key and decrypts the message.
4. If the decrypted string coincides with the string that the remote server has sent to Alice, then the remote server allows the latter access to the system.

Since only Alice has the private key, and this private key has never been exchanged on any channel, no one else can assume the identity of Alice. Even if Eve intercepted the communication, she could not acquire any useful information.

2.2.16 Authentication and key exchange

The protocols for authentication and key exchange solve the general problem of allowing two users to communicate securely, being quite certain of the identity of the person who is on the other side of the channel. They resort to a trusted person (Trent) who shares with each subject a secret key that has been exchanged before commencement of the various protocols.

The most well known of these protocols are: Wide-Mouth frog, Yahalom, Needham – Schroeder, Otway – Rees, Kerberos, Neuman – Stubblebine, Distributed Authentication Security Service (DASS) and Denning – Sacco.

These protocols will not be described in the following, for reasons of space, and the reader can refer to the bibliography section provided at the end of this book for further information.

2.2.17 Multiple public-key cryptography

Public-key cryptography uses two keys (usually one public and one private) and a message that is encrypted with a key can be decrypted with the other.

This concept can be generalised by resorting to the use of multiple keys, for example three keys (K_A, K_B and K_C) distributed as indicated in Table 2.2.

Given the distribution of the keys, Alice can encrypt with the key K_A in such a manner that Ellen, who owns the keys K_B and K_C, can decrypt it. Bob and Carol can do the same by putting their keys together. Bob, moreover, can encrypt a message in such a manner that Frank can read and Carol can encrypt a message such that Dave can read it. Dave can encrypt a message with the key K_A in such a manner that Ellen can read it, with K_B in such a manner that Frank can read or with K_A and K_B in such a manner that Carol can read it. In a similar way, Ellen can encrypt a message in such a way that Alice, Dave or Frank can read it.

This scheme can be extended to the case of n keys taking into account that if a subgroup is used to encrypt a message, the remaining subgroup is necessary to decipher the message.

2.2.17.1 Dissemination of a message

If working with an extensive group of people, the need may arise to send a message to a specific subgroup not defined beforehand. In this case, the message can be encrypted separately for each user, requiring several messages, or the keys for every possible combination of people distributed, requiring many keys.

In this case, it can be extremely useful to use multiple-key encryption. Assuming there are only three users: Alice, Bob and Carol. In this case, Alice will be given the keys K_A and K_B, Bob the keys K_B and K_C and Carol the keys K_A and K_C. From this point onwards, it is possible to communicate with any subgroup. In fact, if we want to send a message in such a way that only Alice can read it, it will be encrypted with the key K_C. When Alice receives the message, she will be able to decrypt it with the keys K_A and K_B. Using the combinations shown in Table 2.3 it can be seen how messages can be sent to any group of users as desired.

Table 2.2 Three key distribution schema.

User	Key
Alice	K_A
Bob	K_B
Carol	K_C
Dave	K_A, K_B
Ellen	K_B, K_C
Frank	K_A, K_C

Table 2.3 Three key message encryption.

If the following key is used to cipher	The following key must be used to decipher
K_A	K_B, K_C
K_B	K_A, K_C
K_C	K_A, K_B
K_A, K_B	K_C
K_A, K_C	K_B
K_B, K_C	K_A

When the number of users becomes substantial, the described system becomes very efficient.

2.2.18 Division of a secret

The division of a secret is used every time information is to be split into more than one subject, in such a manner each may know a part of the secret without, however, being able to trace back to the secret itself unless by putting together the different subjects.

The simplest schema is between the two subjects, in which Trent wishes to divide the secret between Alice and Bob. The protocol works according to the following steps:

1. Trent generates a random string of R bits of the same length as the message M.
2. Trent performs the XOR operation between M and R to generate S.
3. Trent provides R to Alice and S to Bob.

To reconstruct the secret original message, Alice and Bob must perform a single operation that consists of putting together the pieces of the message in their possession and performing the XOR operation.

The technique just described, if performed correctly, is absolutely safe and each piece, if taken individually, is absolutely meaningless. In practice, Trent encrypts the message with a single use tab and provides the ciphertext only to one subject, and the tab to the other subject.

This scheme can be extended to several subjects, running the XOR operation with numerous random strings. For example, if Trent wants to divide the secret message between four subjects: Alice, Bob, Carol and Dave, the following layout should be followed:

1. Trent generates three random strings R, S and T of the same length of the message.
2. Trent performs the XOR operation between M and the three strings to generate U.
3. Trent provides R to Alice, S to Bob, T to Carol and U to Dave.

To reconstruct the secret message M, Alice, Bob, Carol and Dave put together their information and carry out the overall XOR operation.

The proposed system is an award protocol in which Trent can do what he wants. In any case, this protocol has a problem: in the event of the loss of a part of the message by one of the subjects, it is not possible to reconstruct the original M message.

2.2.19 Secret sharing

In some cases, it is necessary to share a secret message between n subjects but it should be possible for a subset of m subjects to merge the information in their possession to reconstruct the original message without putting together the information of the n subjects.

This schema is called threshold and, in this specific case, threshold (m, n).

2.2.20 Cryptographic protection of archives

Cryptography can be used to improve the safety of the archives in such a manner that it is relatively easy to extract the information relating to a single user but extremely difficult to extract, all together, the information of all users.

The schema used is based on the concepts illustrated hitherto, by resorting to a one-way hash function and to a symmetric encryption algorithm. It is assumed that the record for each user is composed of two fields, of which the index field is represented by the last name of the user, to which the hash function applies, while the data field is represented by the name and the address on which a symmetric encryption using as key the user's last name is used in such a manner that, if the latter knows the information, it is not possible to decipher the data field.

The search for a specific surname in the archive is performed by calculating the hash of the same surname and searching in the archive for the record characterised by the same value. Once found, the data field can be encrypted by using its name as the key.

This system is also susceptible to a brute-force attack, trying all possible last names until finding the ones that are contained in the archive itself.

2.2.21 Stamping services

The services of stamping to affix date and certain time on an electronic document have been previously described. They must enjoy the following properties:

1. The data itself must bear the stamping regardless of the physical medium on which it resides.
2. It must be impossible to change a single bit of the document stamped without this having been highlighted.
3. It must be impossible to print a document with a time and date different from the current one.

2.2.21.1 Arbitration solution

This type of protocol uses an arbitrator (Trent) who can provide a service of reliable stamping and refers to Alice who wants to stamp the document.

It operates according to the following steps:

1. Alice sends a copy of the document to Trent.
2. Trent records the date and time of having received the document and keeps a copy in the archives for security reasons.

Where required, Alice asks Trent to send a copy of the stamped document.

This protocol, even if it is viable, presents a series of contraindications: first, the lack of privacy, since Alice must send a copy of the document that may be intercepted during transmission. She could encrypt it but a copy would always be in Trent's archives. In addition, given the need to store a large number of documents, the size of the archive would become significant.

The need for channels with very high bandwidth to send large documents in a reasonable amount of time should, moreover, be highlighted.

Another significant problem is the presence of possible transmission errors that would mean that Trent would be affixing his stamp on a document different from the original.

2.2.21.2 Improved arbitration solution

The arbitration solution can be improved using a one-way hash function and the digital signature operates according to the following steps:

1. Alice computes the hash value of the document using the one-way function.
2. Alice sends the hash to Trent.
3. Trent adds the stamping to the hash received and digitally signs it.
4. Trent sends the signed hash to Alice, together with the date and time.

The solution is better than the previous one in that:

1. confidentiality is protected because Alice sends the hash of the document and not the document itself;
2. the archives of Trent should not be too large as, in this case, the hash values of the documents and not the documents themselves are stored;
3. transmission errors are detected by Alice by controlling the stamped hash value that Trent sends back to her.

2.2.22 Delegated signature

In some cases, a third party may need to affix a signature in place of another subject that authorises them to perform this action.

In this sense, reference is made to delegated signature. Alice can delegate to Bob in such a manner that the following are verified:

1. Distinguishability: the delegated signatures are distinguishable from normal signatures by anyone.
2. Non-forgeability: only the original signatory and the delegate can create a valid delegated signature.
3. Deviation of the signature in delegation: a delegate signatory cannot create a delegated signature without being able to identify the latter as such.
4. Verificability: from the delegated signature it must be possible for the verifier to determine if the delegator has authorised this signature.
5. Identifiability: an original signatory can determine the identity of the delegated signatory starting from the delegated signature.
6. Non-repudiability: a delegated signatory cannot repudiate a delegated signature that he himself/she herself has created.

There are suitable algorithms that allow use of the delegated signature but that will not be shown for reasons of space.

2.2.23 Group signature

Group signature enjoys the following properties:

1. Only members of the group can sign messages.
2. The receiver of a signature can verify that it is a valid signature belonging to the group.
3. The receiver of the signature cannot determine which member of the group is the signatory.
4. In the event of dispute, the signature can be opened to reveal the identity of the signatory.

A solution for implementing a group signature consists of the use of a third arbitrator, operating according to the following steps:

1. Trent generates a group of public – private key pairs and provides each member of the group with a different list of unique private keys in such a way that no key on any list is the same: if there are n members of the group and m pairs of keys with which each member is provided, then the total number of pairs of key copies will be $n*m$.
2. Trent publishes the main list of all the public keys of the group in random order keeping apart a reserved register in which the key assignments are indicated.

3. When a member of the group wishes to sign a document, he/she randomly selects a key from his/her personal list.
4. When a person wants to verify that a signature belongs to a member of the group, he/she needs only consult the list published by Trent.
5. In the event of a dispute, Trent knows to which member of the group the public key belongs and, in this sense, can intervene.

2.2.24 Key escrow

Key escrow is used every time an authorised third party (typically the security forces or judicial authorities) needs to acquire keys for decrypting the messages of a given subject.

To ensure maximum security, Alice may split the key into several parts, as seen in the previous section, and may send it to various trusted subjects. Only the approved person, where necessary, can ask all the trusted subjects to give him/her the key parts in their possession to reconstruct the original key and to be able to decrypt Alice's messages. The system is secure and resistant to attacks as a possible attacker (Mallory) could corrupt all the trusted subjects in order to be able to reconstruct the key.

The protocol operates according to the following steps:

1. Alice creates her public–private key pair and divides it into many key pieces, both public and private.
2. Alice sends a piece of public key and the corresponding private one to each trusted subject, encrypting appropriately into messages. She also sends her own public key to a KDC.
3. Each trusted subject independently processes their own piece of private and public keys to verify its correctness. In addition, each trusted subject securely stores the piece of private key and sends the piece of public key to a KDC.
4. The KDC performs another calculation on the pieces of public key and private key. Assuming that everything is correct, it signs the public key and sends it back to Alice or sends it to an archive.

Where required by the judicial authorities, a request is sent to the KDC which calls on the various trusted subjects to provide the parts in their possession in order to reconstruct the private key and supply it to the applicant.

2.2.25 Digitally certified email

Digitally certified email is very useful when we want to confirm that the recipient has received the message, without the latter being able to read it.

It can be obtained by making Alice generate a symmetric key and sending half of it to Bob who confirms receipt. After that, Alice sends the second half of the symmetric key to Bob who confirms receipt. Once this has been done, Alice may send the encrypted message that can be finally deciphered by Bob.

2.2.26 Length of the symmetric key

It has already been said above that the security of a symmetric cryptosystem depends on the strength of the algorithm and key length.

If the security of the algorithm is indisputable, the only way to attack the system is by brute force, that is by trying all the possible keys.

Calculation of the complexity of a brute-force attack is relatively simple. If a key is 8 bits long, there are $2^8 = 256$ possible keys. A total of 256 attempts are thus needed with a 50% probability of finding the key after half the attempts. If a key is 56 bits long, there are 2^{56} possible keys. If we had, for example,

a computer capable of running a million attempts per second, it would take 2,285 years to find the correct key. If a key is 64 bits long, the same computer would take 585,000 years to find the correct key. If a key is 128 bits long, the same computer would take 10^{25} years, which is an extremely long time taking into account the fact that the age of the universe is 10^{10} years.

There are two parameters that determine the speed of a brute-force attack: the number of keys to be tried and the speed of each test. In theory, it is assumed that the speed of each test does not change from algorithm to algorithm but in practice this is not true.

The majority of the studies of cryptoanalysis were conducted in relation to the DES algorithm, which has already been mentioned above and which will be described in more detail below.

In this sense, parallel machines have been developed, each capable of trying millions of keys per second. Each machine operates on a subset of keys and the same does not communicate with each other. They only communicate in the event of a positive search outcome.

Specialised machines have recently been developed that, for the cost of 1 million dollars they are able to break the DES 56-bit key in 3.5 hours. This machine is characterised by a price/linear speed ratio. In addition, Moore's law should be remembered in these circumstances that say that the computing power of current processors doubles every 18 months: this means that the costs will decrease by a factor of 10 every 5 years.

With 56-bit keys, all that is needed is the economic availability of large companies or organised criminal organisations to purchase a computer able to crack the code in a reasonable space of time. For 64-bit keys, military budgets at the disposal of large industrialised nations are necessary. Currently, 80-bit keys present objective difficulties that may still be overcome, in the future, thanks to the continuous increase in computing power of computers.

In practical cases, to break a key that is not excessively long, all that is needed is the investment of an adequate budget. For this reason, it is more practical to estimate the minimum cost of the key that is directly related to the value of the message to be protected.

To perform cryptographic attacks, dedicated hardware that is specifically designed to perform the desired attack is used. When the calculation needs are not that crucial, it is possible to resort to a software attack that, being non-dedicated, is not able to reach the same speed of the specialised hardware but that can be implemented, at a reasonable cost, on computers that are powerful enough. With current powers of calculation, an attack against DES can be engineered by resorting to 40 workstations that allow 2^{34} keys to be tried in a single day, necessitating 4 million days to try all the possible keys. Using a 64-bit key instead of only 56 bits, the attack becomes 256 times more difficult. With 40 bits, a network of 400 computers, each capable of performing 32,000 attempts per second, is capable of completing a brute-force attack in a day. From what we have seen so far, it is evident that a 128-bit key ensures, in the current state, a high level of security since there are no available resources and sufficient time to successfully conduct a brute-force attack.

In certain cases, use can be made of advanced software techniques such as neural networks that are able to learn by proposing solutions that are increasingly efficient.

At times, solutions have been created that are not excessively correct, such as the use of viruses, which, by infecting, if they can, all the computers on the network, force them to work, in background and in a hidden manner, to attempt a brute-force attack using a subset of keys. This technique makes it possible to make available a large number of computers and computing resources if the virus succeeds in entering, fraudulently, a considerable number of machines. To get an idea of the potential of this technique, if it is assumed that the virus is able to infect 10 million computers, each able to carry out a thousand attempts per second, it would take 83 days to violate a 56-bit key and 58 years to crack a 64-bit key.

At this point, a number of thermodynamic considerations should be made. The second principle of thermodynamics, transferred to the field of information, states that a certain amount of energy to represent the information is always required. To record a single bit, the status of a system must be varied and the thermodynamics tell us that the minimum amount of energy is kT, where k is the

Boltzmann constant (which is 1.38×10^{-16} erg/Kelvin) and T is the absolute temperature, expressed in Kelvin degrees. If the temperature is assumed to be the ambient temperature of the universe, that is 3.2 K, an ideal computer would consume 4.4×10^{-16} erg to vary the value of a bit. The annual energy emitted by the sun is 1.21×10^{41} erg that represents the energy required to vary 2.7×10^{56} bits on our ideal computer, corresponding to all the possible variants of a 187-bit binary string. If it were possible to use the whole energy of the sun for 32 years we might get 2^{192} bits on our ideal computer. If we consider the energy emitted annually by a supernova, equal to 10^{56} erg, this would allow trying of all the combinations of a binary string of 219 bits. These numbers do not, of course, take into consideration the power of calculation actually available but represent exclusively the energy type considerations. From what we have seen so far, it is evident that a 256-bit key can be considered as extremely secure.

2.2.27 Public-key length

Public-key cryptography uses a one-way function with trap door represented by the ease of multiplying, for example, two prime numbers but with the difficulty of factoring their product in components prime numbers. The factorisation of two numbers is a problem of great mathematical difficulty that so far has never been solved by an algorithm but which can only be resolved through successive attempts: the larger the number, the greater the number of attempts to find it.

There are, however, other factoring algorithms based, for example, on the discrete logarithm problem, which are used for public-key cryptography.

Computing power is generally expressed in mips/year corresponding to a million operations per second (mips) performed by one computer for a year. This number corresponds to 3×10^{13} operations overall.

If it is assumed that an expert cryptoanalyst has 10,000 mips/year, a large company of 10^7 mips/year, a substantial national government of 10^9 mips/year, that the computing power doubles every 5 years and that we can factor to the maximum speed allowed by the best encryption algorithms regardless of the length of the key, we can compile a table of recommended key length according to the year and type of attacker (see Table 2.4).

It is clear that Table 2.4 is only valid with the current knowledge, that is, as far as possible, it is to find an algorithm capable of factoring in a faster manner. If such algorithm were to be found, it would skip the entire security of public-key encryption.

Table 2.4 Recommended length of the public key (in bits) according to the year and type of attacker.

1.1	Attacker		
Year	Single person	Large enterprise	Government
1995	768	1,280	1,536
2000	1,024	1,280	1,536
2005	1,280	1,536	2,048
2010	1,280	1,536	2,048
2015	1,536	2,048	2,048
2020	1,750	2,400	2,650
2025	1,990	2,750	2,850

2.2.28 Comparison between the length of the symmetric key and the length of the public key

A system is usually attacked at its weakest point. If we employ a system that uses both the symmetric and public keys, the length of the latter must be chosen in such a manner that is equally difficult to attack the system via one of the two keys.

Table 2.5 compares the lengths of the public keys subjected to factorisation attack with the equivalents of symmetric keys subjected to brute-force attack.

2.2.29 Birthday attacks in relation to one-way functions

There are two fundamental techniques to attack a one-way hash function with brute force. The first is the more immediate since given a message M and its hash $H(M)$, we want to find a message M' such that $H(M') = H(M)$.

The second technique is to find two random messages M and M' such that their hash value is the same, that is, such that $H(M') = H(M)$. This situation is also called collision and represents an attack that is much simpler to conduct with respect to the first.

Let us now examine a statistical problem known as birthday paradox. The question that is posed is: how many people must be in a room so that at least one of them has the same birthday date as the individual concerned with a probability greater than 0.5? The response, having made the appropriate calculations, is 253. And again: what is the minimum number of people that must be in a room so that two of them are born on the same day with a probability greater than 0.5? The answer is 23 because with 23 people 253 different combinations can be made.

The search for someone born on a specific day is analogous to the first type of attack, while the search for two people born on the same day is similar to the second type of attack which, because of its characteristics, is called birthday attack.

It is supposed to have a secure hash function where the only way to tackle it is by brute force and that it produces in output a string of m bits. The search for a message that provides the same string in output would require an attempt of 2^m random messages. Instead, a search for two messages that provide the same hash value would require the attempt $2^{m/2}$ random messages. A computer capable of calculating hash values at a rate of 1 million messages per second would take 600,000 years to find a message that provides the same hash value of 64 bits in a given message. The same machine, to find two messages with the same hash value of 64 bits, would require only 1 hour.

This means that if we are afraid of a birthday attack, we need to choose a hash string length that is at least double that deemed necessary.

2.2.30 Optimal key length

There is no well-defined rule that permits us to choose, with precision, the length of a given key.

Table 2.5 Length that ensures the same level of safety for public and symmetric keys.

Length of the public key (bit)	Length of the symmetric key (bit)
384	56
512	64
768	80
1,792	112
2,304	128

To correctly select the length of a key, a number of questions must be asked concerning the value of the message to be protected, the duration of the time of our protection and the resources of the attacker.

Table 2.6 reports the safety requirements, in terms of time, of different types of information and the relative length of the required key.

Future power calculation is extremely difficult to estimate. The only rule that can currently be defined is that the divided processing efficiency divided by the relevant cost doubles every 18 months and increases by a factor of 10 every 5 years. This rule obviously applies to general use computers without thinking about possible future developments of machines for cryptanalytic techniques based on different technologies.

2.2.31 Key management

In everyday use, assuming the use of algorithms characterised by perfect security, a weak point is represented by key management, as it is extremely difficult to keep a key secret and cryptoanalysts mostly attack cryptographic systems through vulnerabilities of the key management system.

For example, it is very easy and cheap to bribe staff within an organisation to acquire a key rather than purchasing an expensive system of cryptoanalysts.

Keys must therefore be protected with the same attention with which encrypted data are protected.

Many systems encrypt data stored within them but retain the relevant keys in a file that is easy to attack.

2.2.32 Key generation

DES, for example, uses 56-bit keys and, in theory, each string of 56 bits can be used as a key. However, some programs pose limitations, such as the use of letters, lowercase or uppercase only, drastically reducing the size of the space of the keys.

To get an idea of the numbers involved, Table 2.7 shows the numbers of possible keys depending on the type of string used in input and the length, in bytes, of the string itself, whereas Table 2.8 indicates the time needed to try all possible keys referred to in Table 2.7, using a computer at 1 million attempts per second.

In most cases, when a key is chosen, the selection is made by focusing on weak keys because generally first name, surname, date of birth, the name of a spouse or partner, the name of children, the name of our dog, etc., are used. In this case, an intelligent brute-force attack does not need to try all possible keys, only those that are more obvious and taken for granted. This type of attack is referred to as dictionary because the attacker uses words taken from a dictionary of common words. A dictionary

Table 2.6 Security requirements for various types of information and the length of the required key.

Kinds of information	Requested protection time	Minimum length of the key (bit)
Tactics military information	Minutes/hours	56 to 64
Want ads	Days/weeks	64
Long-term business plans	Years	64
Trade secrets	Years	112
Military projects	>40 years	128
Spies identities	>50 years	128
Personal affairs	>50 years	128
Diplomatic information	>65 years	>128

Table 2.7 Number of possible keys depending on the type of string in input and the length of the same in bytes.

Kinds of string	4 byte	5 byte	6 byte	7 byte	8 byte
Lowercase (26)	460,000	1.2×10^7	3.1×10^8	8.0×10^9	2.1×10^{11}
Lowercase and numbers (36)	1,700.00	6.0×10^7	2.2×10^9	7.8×10^{10}	2.8×10^{12}
Alphanumeric characters (62)	1.5×10^7	9.2×10^8	5.7×10^{10}	3.5×10^{12}	2.2×10^{14}
Printable character (95)	8.1×10^7	7.7×10^9	7.4×10^{11}	7.0×10^{13}	6.6×10^{15}
ASCII character (128)	2.7×10^8	3.4×10^{10}	4.4×10^{12}	5.6×10^{14}	7.2×10^{16}
8-bit ASCII characters	4.3×10^9	1.1×10^{12}	2.8×10^{14}	7.2×10^{16}	1.8×10^{19}

attack is very powerful if applied to a file of keys rather than to an individual key: if there are several keys in the file, the dictionary attack will be more likely to find at least one valid key.

The best keys are represented by strings of random bits generated by automated processes. If a key is characterised by a certain length n, then the process of generation is of good quality if all the keys generated are equiprobable. For this reason, perfectly random generators should be used.

Some cryptographic algorithms are also characterised by weak keys as some specific keys are weaker than others even if their number is negligible compared with the total number of keys. DES, for example, has only 16 weak keys in relation to all possible 2^{56} keys and thus the probability of generating these keys, via a random process, is extremely low.

In the case of public-key systems, generation of the key is more complex, since it is not possible to directly use a random process, since the keys themselves must adhere to all the mathematical laws, such as the factoring of prime numbers which will be discussed in more detail later.

Since a perfectly random key is usually difficult to remember, pass phrases are often used, which are relatively easy to remember which are used for the generation of keys. One widely used technique is the so-called trituration of the key or number crunching. In essence, this consists of a hash function that transforms a computed string into a pseudo-random binary string, of shorter length. If the pass phrase is long enough, the key generated will be random. To determine a security length, it should be remembered that, according to the information theory, the English standard (a language of international reference) is characterised by 1.3 bits of information per character. Using a dozen words in the English language, corresponding, on average, to about 49 characters, secure key of 64 bits can easily be generated. As a general rule, it can be said that it takes five words for every four key bytes, using a conservative criterion which consists of ignoring the punctuation characters.

This technique can also be used for the generation of public–private key pairs. In this case, the string could be given in input to a random generator and the output could be used as the seed of a deterministic generator to produce the key pair.

Table 2.8 Time required (in seconds) to find all the possible keys referred to in Table 2.7.

Kinds of string	4 byte	5 byte	6 byte	7 byte	8 byte
Lowercase (26)	0.5 seconds	12 seconds	5 minutes	2.2 hours	2.4 days
Lower case and numbers (36)	1.7 seconds	1 minute	36 minutes	22 hours	33 days
Alphanumeric characters (62)	15 seconds	15 minutes	16 hours	41 days	6.9 years
Printable character (95)	1.4 minutes	2.1 hours	8.5 days	2.2 years	210 years
ASCII character (128)	4.5 minutes	9.5 hours	51 days	18 years	2,300 years
8-bit ASCII characters	1.2 hours	13 days	8.9 years	2,300 years	580,000 years

2.2.33 Key transfer

It has already been said that key transfer represents a significant problem in the field of cryptography.

It can also be seen that this problem can additionally be obviated by exchanging keys through public cryptography.

What must never be done is to exchange the key in letter form on the same unsecured channel on which the encrypted message will be exchanged.

The key must thus be exchanged either directly or by resorting to another secure channel.

The ideal situation would be to split the key into multiple parts, according to the technique described above, and then send these parts using several media (phone, email, ordinary track, etc.): a possible attacker who was aware of the parts of keys could not do anything if he/she was unable to take possession of all of the pieces that make up the key.

In networks consisting of many users, the exchange of keys represents a significant problem, since each pair of users must be provided with a key, as seen above, and if there are n users, a number of keys equal to $n(n-1)/2$ must be exchanged.

With an increase in the number of users, it is convenient to have access to a central server for the exchange of keys, taking into account that the latter must ensure a high level of safety and reliability since its attack by a malicious person would undermine the entire security of the network.

2.2.34 Key verification

Key verification is another significant problem in cryptography because it is extremely important to be sure of the identity of the subject with which we want to exchange encrypted messages.

The safest way is to exchange the key directly but this can, in some cases, be difficult if the two subjects are located far apart. Alice could send the message to Bob via courier but, in this case, Bob would need to be confident about the courier used. If the message is encrypted using a different key, Bob would need to be certain that only Alice has that key. If Alice uses a digital signature protocol then Bob should be able to trust the public-key databases that sign the key on Alice's behalf and of course trust that Alice will keep the key safe. If Alice uses a key distributions centre then Bob must be certain that this key has not been tampered with.

In practice, a malicious subject able to control the network around Bob can deceive the same in many ways.

To avoid this, combined techniques can be used such as cryptography public key together with digital signature and a trusted key distributions centre. In doing so, all attackers who do not have sufficient technical or economic resources to conduct the attack are excluded.

Another sure way to verify the public key of a subject is to exchange it directly via telephone, to identify vocally the other subject.

In some cases, the keys may be altered during transmission on real communication channels precluding the decryption of documents or data of considerable size. For this reason, keys should be transmitted using techniques for the identification and correction of errors. One of the most used techniques is to encrypt a constant value with the key to be exchanged and to send the first 2 to 4 bytes of the encrypted value together with the key. In addition, decryption is performed of the encrypted constant using the key sent: if the two values match this means that the key was exchanged in the correct manner.

2.2.35 Using keys

The software applications of encryption may be, in some cases, extremely vulnerable during the encryption process if the latter is interrupted during its normal operations because the applications

themselves can save temporary data (such as encryption keys) on a hard disc, making this data vulnerable to external attacks.

Hardware applications are generally more secure than software as the keys are destroyed in the event of a malfunction or external attacks.

2.2.36 Key update

When, due to security requirements, a daily exchange of new keys is necessary, the mechanism of transmission and management could become excessively burdensome. In such cases, it becomes extremely useful to resort to a system, called key refresh, which makes it possible to generate a new key from the old one without any new key needing to be transmitted.

The simplest system consists of calculating the old key in hash value to generate the new one: if the keys are the same, the same result from both parties will be obtained. Once the result has been obtained, the number of bits needed for the correctness of the new key can be achieved.

The system works, of course, if the old key is secure: if Eve came into possession of the old key, she could easily generate a new key and decrypt all the encrypted communications.

2.2.37 Key storage

In many cases, encryption systems are used to protect the data on a computer. In this case, only the user knows the encryption and decryption keys and, at any time by typing it, can have access to encrypted data.

Another solution is to store the key on a magnetic band card in which a ROM chip (smart card which will be dealt with in more detail later) has been inserted and submit it to the computer reader every time access to encrypted data is required. This system is even more secure than the previous one in that the user himself/herself does not know the content of the chip, and thus directly the key.

The ROM chip technique can be made even more secure by dividing the key into two parts, one contained in the chip and the other contained on the computer itself: if the two parts are not put together, the key cannot be obtained. This protects the system from any loss or theft of the card containing the ROM chip.

Particularly, complex keys can be stored in encrypted form using an encryption key. For example, an RSA key can be encrypted using a DES key and the RSA encrypted key could be registered, without problems, on the computer itself. To retrieve the RSA key, the user needs only type the DES key into a special decryption program.

The general rule prevails that the encryption key should never appear in letters outside the decryption device.

2.2.38 Compromising of keys

In the event of loss, theft or spread of a key, all the protocols, techniques and algorithms seen from that point onwards are compromised and the security of any system based on them is compromised.

If the compromised key is symmetric, all Alice needs to do is to change the key and hope that, in the meantime, damage is minimised. If the key is private, the problem assumes greater importance, as the same key can be used to impersonate Alice herself on all servers where the public key is stored, reading email, signing documents and so on. It is therefore extremely important that information regarding compromise of a private key propagates as quickly as possible within a network. If Alice uses a KDC, she should notify the news of compromise of the relative key in the shortest time possible in such a manner that the KDC disseminates notification of compromise. If Alice does not use a KDC, then she herself should send the information to all interested parties.

To minimise the damage caused by the compromise of a key, the use of multiple keys depending on the uses that are to be made of it is recommended.

From what we have seen so far, the importance of closely guarding keys, with particular attention to private keys, is evident.

2.2.39 Lifespan of keys

As a general rule, it can be said that no cryptographic key should be used for an indefinite period of time: it should be renewed periodically. There are various reasons to do this:

1. The longer a key is used, the greater its chances of being compromised.
2. The longer a key is used, the greater the loss in case of compromise of the key itself.
3. The longer a key is used, the greater the temptation for an attacker to want to attack, possibly using a brute-force attack.
4. Cryptoanalysis can be much easier to perform on a large amount of ciphertext with the same key.

Depending on the cryptographic application used, a different lifespan of keys should be established. For example, for a phone call, a disposable key should be provided, only valid for the duration of the conversation itself.

With regard to the systems that operate on dedicated channels, it is the general rule that the keys should have a relatively short lifespan and in any case should depend on the value of the data exchanged and the amount of data to be encrypted. A good rule is to change the key every day, as long as there is a secure key exchange system.

Encryption keys of keys should not, however, be changed frequently as they are used sporadically and on a small amount of data: in this way, a possible attacker has a small amount of ciphertext on which to operate an attack. In any event, the compromise of an encryption key of keys is an extremely dangerous event because it compromises the security of all the keys exchanged. In some applications, the encryption keys of keys are replaced once a month or once a year since there has to be the right balance between the risk of always using the same key and the need to distribute a new key.

The keys used to encrypt data on a computer should not be changed too often because the files themselves may remain on the disc for a long time before being used, because time is wasted deciphering with the old key and encrypting with the new one and because a large amount of encrypted material is generated for a possible attack. A valid solution is represented by encrypting every file with a different key and then encrypting the file containing the keys, suitably storing this last key.

Private keys used for public encryption are characterised by variable lifespans, depending on the application. Private keys used for digital signature and confirmation of identity can also last for several years.

2.2.40 Destruction of keys

Once a new key has been generated, it is a good rule to destroy the old keys as anyone who came into possession of such could read old messages encrypted with it.

Destruction must occur in a secure and reliable manner. In particular:

1. If the key was written on a sheet of paper, the same sheet should be destroyed by high-quality shredding in such a way that it is impossible to reconstruct the original sheet from the individual pieces.
2. If the key was stored on an Electronically Erasable Programmable Read Only Memory (EEPROM), the same should be overwritten several times.
3. If the key was written on an EPROM or ROM, the same chip should be thoroughly destroyed.
4. If the key was written on a removable memory support (pen disc or other), the related bits concerned should be overwritten.

2.2.41 Key management in public-key systems

Key management in public-key systems is relatively simple but has also some drawbacks.

If Alice wants to send a message to Bob, she can get the key from Bob, take Bob's key from a centralised database or retrieve the key from her own storage. This allows Mallory to conduct a series of attacks, which we have already examined, and which will not be detailed in this section for the sake of brevity.

To avoid attacks, the so-called public-key certification can be employed that is in fact a certificate which is combined with a key and issued by a trusted entity that is called certification authority (CA). The certifier retains all data related to the certificate and verifies the identity before releasing the corresponding certificate.

2.2.42 Algorithm types and modes

There are two fundamental symmetric algorithms: block ciphers and stream ciphers. Block ciphers operate on blocks of plaintext or ciphertext typically of 64 bits even if some are sometimes longer. Stream ciphers operate on streams of plaintext or ciphertext 1 bit or byte at a time.

In block ciphers, the same plaintext is always encrypted with the same ciphertext while in stream cipher the same plaintext is encrypted in a different manner.

A cryptographic method usually uses a base cipher, feedback and a number of simple operations that are usually relatively simple because the security must be based on the cipher used and not on the method in its entirety. The method, moreover, should maintain the same efficiency of the cipher of which it is composed and the encrypted messages produced should possibly be of the same length as plaintext messages.

2.2.42.1 Electronic codebook mode

Electronic codebook mode (ECM) is the most immediate method according to which a block cipher can be used where a block of plaintext is used encrypted within a block of ciphertext.

Each block is encrypted independently and there is no need to insert the block in order, because the desired block can be inserted from time to time.

The problem with this type of system is that if a possible attacker comes into possession of a large quantity of plaintext and ciphertext, he/she can assemble a decryption code without being in possession of the key. In fact, if the attacker discovers that a plaintext string corresponds to another encrypted string, he/she can replace all the encrypted strings found of this type with the plaintext one and start decoding the document, thanks to the high presence of redundancies in the written documents.

In addition, it is extremely risky to encrypt several messages with the same key because each block can be considered as a separate message encrypted with the same key.

The advantage of this type of method is that any error in transmission affects only and exclusively decoding of the block in which it is present, leaving unaltered the decoding of the other blocks.

2.2.42.2 Cipher block chaining mode

If we want to increase the security in the encryption of a block mode, a type of feedback can be added, bringing the output into input, in such a manner that encryption of a block depends on the previous one. This operating mode is also called cipher block chaining (CBC) using an XOR operation between the plaintext and the block just encrypted before encrypting the text itself. The operational schema is shown in Figure 2.6.

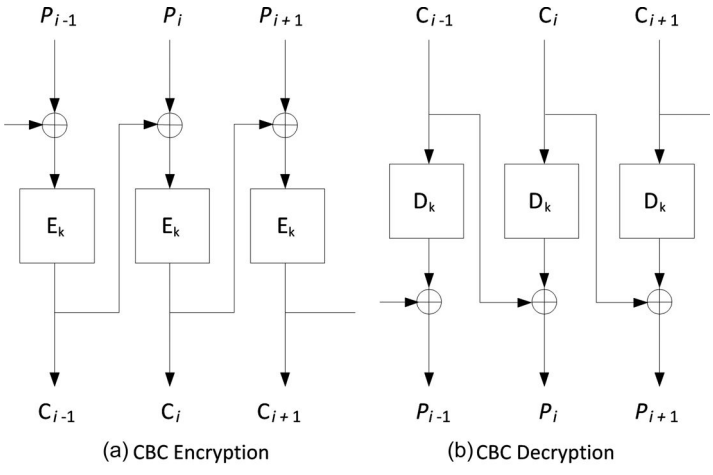


Figure 2.6 Operating schema of a cipher block chaining.

Figure 2.6 shows how a block cipher that has just been encrypted is stored in a special register, and before the next block is encrypted, an XOR operation is performed between the next plaintext block and the block just encrypted. In this way, the encryption of each block depends on all the preceding blocks.

Decryption is performed by means of a mirror system: the block just decrypted is saved in a log and is executed on the XOR between this block and the next block still encrypted, repeating this operation for all blocks, until completion of decryption of the entire encrypted message.

The method in question remains vulnerable because it always encrypts two identical messages in the same manner. This can be avoided by inserting at the beginning, from time to time, a totally random block of data, called initialisation vector (IV). It is totally meaningless but allows two identical messages to be encrypted in completely different manners. It need not necessarily remain secret but can be transmitted in plaintext along with the encrypted message.

2.2.42.3 Stream ciphers

Stream ciphers convert plaintext into ciphertext 1 bit at a time. The easiest way to do this is to use a stream generator of bit keys (keystream generator) k_1, k_2,. . ., k_i with which an XOR operation is performed, bit by bit, with plaintext bits p_1, p_2,. . ., p_i of the plaintext in order to obtain an encrypted test. To decipher the message, a similar XOR operation is performed, bit by bit, between the ciphered text and the bits produced by a similar key generator on the side of the receiver. A diagram of the operation is shown in Figure 2.7.

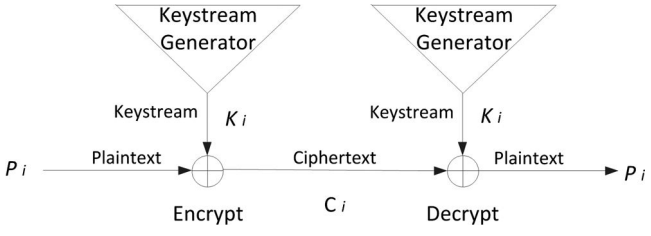


Figure 2.7 Schematic diagram of the operation of a stream cipher.

The security of this system depends on the quality of the key generator: if the generator is perfectly random, it generates, in practice, a single use tab, already shown above and practically perfect security is obtained.

Since the key generator, each time it is started, could generate the same stream of keys, making the system vulnerable, generators capable of accepting in input a generation key that is used as the basis for the generation of the flow of keys are usually used: varying the input key inevitably varies the flow of keys generated.

Stream ciphers are used for continuous communications, such as those that occur between two computers when they are connected.

2.2.42.4 Self-synchronising stream ciphers

In self-synchronising stream ciphers, each key bit is a function of a certain number of bits previously encrypted. The layout is shown in Figure 2.8.

In this case, the internal status of the generator of keys depends on a number of bits *n*, arbitrarily selected, of the ciphered text and this system allows the receiver key generator to synchronise perfectly with that of the transmitter after receiving the same number of *n* bits of ciphered text, hence the name self-synchronising.

This system is extremely vulnerable to transmission errors since the random variation of a single bit would cause the generation of incorrect key of *n* bits, producing a similar error of *n* bits in the text to be deciphered.

2.2.42.5 Feedback cipher mode

Blocks ciphers may be implemented as self-synchronising stream ciphers, producing a feedback cipher. In this system, decoding may not begin until a whole block has been received.

Figure 2.9 shows a schematic diagram illustrating the operation of an 8-bit feedback cipher.

Initially, the queue is filled with an initialisation vector and is encrypted. Thereafter, an XOR operation is performed between the 8-bit output of the queue and the first 8 bits of the input plaintext, encrypting it. The ciphertext is both sent and inserted again in the queue, not before having emptied it of the bits of the initialisation vector. This process is repeated for all the bits that make up the plaintext until it is fully encrypted.

It is clear that the security of the system is also based on the initialisation vector that must be changed every time a new encryption operation is started.

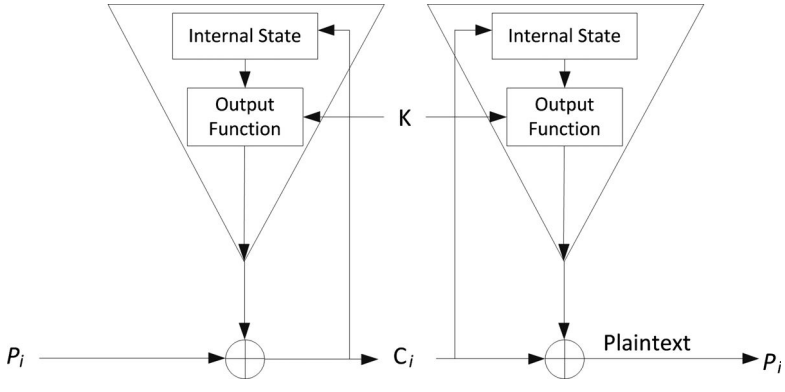


Figure 2.8 Schematic diagram of the operation of a self-synchronising stream cipher.

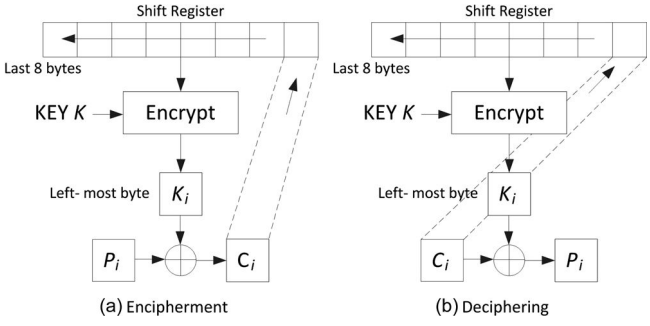


Figure 2.9 Schematic diagram of the operation of an 8-bit feedback cipher.

2.2.42.6 Synchronous stream ciphers

In synchronous stream ciphers, the key is generated independently of the message flow. It is clear that, since, in this case, the stream of keys does not depend on the message itself, there must be a system to synchronise the key generator from the transmitting side with the key generator from the recipient side. If a single bit of the ciphertext is lost in transmission, the two generators lose the synchronism and a totally incorrect decrypted message will be generated: in this case, the two key generators must synchronise again before the start of a new decoding operation.

Since the two generators must generate the same stream of keys from both sides, they must be deterministic and being implemented on finite-status machines, the same will generate periodic sequences that are in any case long. These sequences must be at least as long as the message itself in order to generate a single use tab system. Failing this, the encrypted messages could be attacked.

Stream ciphers are open to insertion – deletion attacks as this situation would immediately lead to loss of synchronism between the key generators, generating an incorrect decrypted message and revealing the attack.

2.2.42.7 Output-feedback mode

Output-feedback mode (OFB) is used by joining together a block cipher and a synchronous stream cipher. A diagram of the operation is shown in Figure 2.10.

An advantage of this mode is that the work of coding/decoding can be performed even if not directly in-line, but previously, on the transmission side, and, subsequently, on the receiving side.

2.2.42.8 Block chaining mode

In block chaining mode (BCM) the function of XOR between the input block and all the previously encrypted blocks is performed using from the start a suitable initialisation vector. The disadvantage of such a system is the fact that an error of only 1 bit during transmission generates a decrypted message that is totally different from the one transmitted.

2.2.42.9 Comparison between block ciphers and stream ciphers

Even if the two groups of ciphers are conceptually different, it is always possible to implement block ciphers as stream ciphers and vice versa.

In everyday applications, block ciphers are more general while stream ciphers are mathematically easier to analyse.

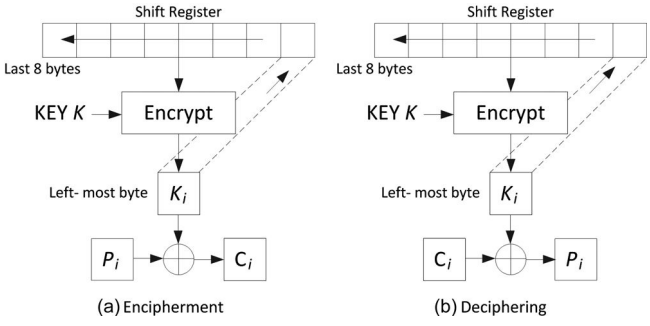


Figure 2.10 Diagram of the operation of an 8-bit output-feedback mode.

Their greatest difference is in the implementation mode. Block ciphers can easily be implemented via software, being able to operate directly on blocks of data stored on a computer. Stream ciphers, on the contrary, are implemented via hardware, having to operate on bit streams in transit.

2.2.43 Use of algorithms

The system's security is linked to the security of the weakest element that makes up the system itself. For this reason, each element of the system (cryptographic algorithm, protocol, key management, random number generator, etc.) must ensure the same level of security.

A secure communication systems designer must think of every possible means of attack and all possible means to prevent such attacks.

2.2.43.1 The choice of algorithms

The choice of an algorithm is not that simple and straightforward and a series of alternatives must be considered, such as:

1. selecting a known algorithm, relying on the fact that this algorithm has been sufficiently analysed, since it is public: if no one has managed to breach it, this means that it is, at the time of selection, secure enough;
2. selecting a trusted manufacturer, trusting their reputation and the reliability of their products;
3. resorting to a private consultant, relying on the fact that the same, not being bound to any manufacturer, is able to advise on the better algorithm for our needs;
4. relying on the advice of the government, trusting that the government itself is in a position to assure the best choices, in terms of security for its citizens;
5. writing our own algorithm, trusting directly in your own ability.

In the United States, the export of cryptographic algorithms must be previously approved by the Federal Government. It is widely held belief that algorithms declared exportable would be violated by the National Security Agency (NSA).

2.2.43.2 Comparison between symmetric-key and public-key algorithms

Symmetric-key and public-key algorithms are not easy to compare, given the different purposes for which they are created.

It can however be stated that, given the same message to be encrypted, symmetric-key algorithms produce a message encrypted with a length of less than that produced with public-key algorithms,

relying on a greater efficiency of encryption and a higher speed of the same compared to the other group of algorithms.

It can also be said that, on the one hand, symmetric algorithms are better in the encryption of data and are not particularly vulnerable to chosen-ciphertext attacks. Public-key algorithms, on the other hand, are able to perform a series of features that cannot be carried out by the other group, are better from a point of view of the management of keys and are able to perform a series of other essential services, such as one-way hash functions, the authentication of messages and other aspects, as shown above. Table 2.9 provides a summary of comparisons between the various cryptographic algorithms seen so far.

2.2.43.3 Encryption of the communication channels

When we want to exchange an encrypted message between two subjects, it is possible to operate at any level of the International Organisation for Standardization/Open System Interconnection (ISO/OSI) stack. If working at the lowest levels of the stack, this is referred to as link-by-link encryption, while if working at the highest levels of the stack, this is called end-to-end encryption.

The easiest way to add an encryption service is to insert it on a physical level, implementing a link-by-link service. The interfaces used at a physical level are usually standardised and it is extremely easy to add dedicated devices of cryptography at this level. These devices encrypt all the data transiting through them, including the routing information and communication protocol. They can be used in every type of digital communication channel. This type of encryption, traffic flow security, is very effective as any type of information (data and traffic control) is encrypted and a possible attacker will not know anything about who is sending what. In addition, the security of the traffic does not depend on the key management system, since each pair of terminals on the network must share a key that can be changed at any time in the event of a breach, without invalidating the entire security of the network. The major problem of a link-by-link system is the need to insert an encryption device in each node of the network: if the network is very large, the cost might become excessive. The advantages of a link-by-link system are: ease of operation, request for a single key for every link of the network, implementation of traffic encryption service at a low level and online encryption. The disadvantages are the vulnerability of the intermediate nodes.

If the encryption service is inserted at the highest level of the ISO/OSI stack, typically between the network and transport layers or even higher, this is referred to as end-to-end service. In this case, encryption is only carried out on data, as it cannot act on the information of traffic that is generated in the lower layers of the ISO/OSI stack: this can be a security problem since a potential attacker can obtain a great quantity of information concerning the flow of traffic. The realisation of an end-to-end encryption service can be quite complex to implement as, at the level at which it must be introduced, various types of protocols operate and the system has to interface with these. If the encryption service is introduced at the application or presentation level, it can be independent of the type of

Table 2.9 Comparison between the various cryptographic algorithms.

Algorithm	Confidentiality	Authentication	Integrity	Key management
Symmetric key	Yes	No	No	Yes
Public key	Yes	No	No	Yes
Digital sign	No	Yes	Yes	No
Key agreement	Yes	Optimal	No	Yes
One-way hashfunction	No	No	Yes	No
Message authentication	No	Yes	Yes	No

communication network used. At these levels of implementation, the encryption service interacts directly with the application software of the user. In this case, the encryption can be implemented directly in the software of the user. The greatest advantage of the end-to-end encryption is the greater level of security, while the disadvantages are its vulnerability to the analysis of traffic and the need to use a more complex key management system.

If link-by-link encryption systems are appropriately combined with those end-to-end ones, it is possible to significantly increase the level of security of the communication system even if with higher costs due to the increase in the number of devices and the software necessary. In this case, the manager of the network is responsible for management of the keys at the physical level while individual users are responsible for management of the keys at the end-to-end level.

2.2.43.4 Data encryption for storage

In communication channels, encrypted messages that are sent may not have an extremely high value: if the recipient does not receive a message, the same can always request retransmission. This is not true for messages encrypted for storage: if a subject is no longer able to decipher a message that he himself/she herself has encrypted in his/her time, the message and its value are lost forever. For this reason, data encrypted for storage should include a recovery system.

The keys, in this case, may have a relatively long lifespan and data can be encrypted with these keys and remain on the storage media for years. For this reason, the keys should be scrupulously kept.

If an entire hard disc must be encrypted, two solutions can be adopted. The first solution is to encrypt all the data using a single key. This solution provides the attacker with a large amount of ciphertext which can be analysed and makes it impossible for multiple users to access individual files without having to decode the entire encrypted file. The second solution is to encrypt each file with a different key and encrypt the file containing the decryption keys with a key known only to those users who have access to the files themselves. In the latter case, different users may have subgroups of keys in order to access the files to which they are enabled. This solution is much more secure as the encryption keys of the files are random and therefore less susceptible to dictionary attacks.

The hard disc can be encrypted in two ways: at the file level and at the level of driver. Encryption at the file level takes place through encrypting, in fact, every single file separately. If we want to access a particular file, we must first decrypt it, then access it and then encrypt it again. Driver encryption generates a logical drive on the computer in which all the data encrypted is stored. If implemented well, it can provide higher security compared to the first solution, requiring reduced operational action by the user. The latter case is more complex to implement on the desired computer because the driver must be installed on the machine, appropriate sectors must be allocated on the hard disc, a mechanism of random access to encrypted data must be operated and so on. Before starting, each time, the driver asks the user for the password.

2.2.43.5 Comparison between encryption via hardware and encryption via software

Although encryption software is currently used in most applications, encryption hardware always remains a reference for military applications and for commercial applications with a high degree of security.

Encryption via hardware undoubtedly ensures a higher operating speed, as the machines are designed to perform the operations required by an algorithm of specific encryption with maximum efficiency in contrast to normal computers that are equipped to perform more general operations

Hardware encryption is also more secure as the hardware itself is designed to prevent any intrusion during encryption and decryption operations, which is not the case with encryption software. In some cases, the same hardware devices are sealed in sabotage-proof containers. Since every electronic device

during its normal operation emits electromagnetic waves and, in the case of encryption, this constitutes a valuable source of information if suitably intercepted, the containment boxes are designed to reduce the electromagnetic emission of the circuits contained therein to practically zero, using advanced techniques such as TEMPEST, which will be explained in the chapter 8 on protection against eavesdropping.

Encryption hardware is, furthermore, easier to install than encryption software that must be properly installed on the desired machine, always requiring in any case the assistance of a computer even for simple applications such as the encryption of a normal telephone transmission or by fax.

There are three major systems currently being used for encryption hardware: self-consistent devices for encryption communications and cards to be inserted into the computer. The software, on the other hand, allows the implementation of any encryption algorithm, being characterised by a lower speed and ease of manipulation of the software itself.

2.2.43.6 Compression, encoding and encryption

The use of compression algorithms together with encryption algorithms has the following advantages: compression reduces the redundancy in the text to be encrypted that represents an element of vulnerability to attacks and compression reduces the volume of data to be treated that increases the speed of encryption.

Compression should be carried out before the encryption operation because if the encryption algorithm is of high quality, it is practically uncompressible.

2.2.43.7 Destruction of the information

To delete a file from a computer, it is not sufficient to provide a simple erase operation because the file itself is still on the hard disc, with it always being possible to recover it using suitable programs. For this reason, to delete a file from our hard disc, we must write on the space it occupied several times, for security reasons.

2.3 Elements of basic maths for cryptography

2.3.1 Information theory

The information theory, as we know today, was formulated for the first time by Claude Shannon in 1948. In the following, we reported only the basic concepts which are useful in the field of cryptography.

It defines information quantity of a message as the minimum number of bits needed to encode all the possible meanings, assuming that all the messages are equally likely.

The amount of information in a message M is measured by the entropy of the message itself and is referred to as $H(M)$. If it is measured in bits, the same can be calculated as $\log_2 n$, n being the number of possible meanings of the message.

Given a certain language, rate of language r is defined as the magnitude $H(M)/N$, N being the length of the message. For example, the rate of the English language is between 1.0 bits/letter and 1.5 bits/letter for large values of N.

It defines absolute rate of a language as the maximum number of bits that can be encoded in each character, assuming that every sequence of characters is equally likely. If, in a given language, there are L characters, then the absolute rate R is equal to $\log_2 L$, which is also equal to the maximum entropy of individual characters. If we consider the English language, with an alphabet of 26 characters, the

absolute rate is equal to about 4.7 bits/character even if the normal rate of the spoken language is less, being the language spoken characterised by a high number of redundancies.

The redundancy of a language is defined as $D = R - r$. Considering the English language, if it is assumed that the average rate is equal to 1.3, the redundancy is equal to 3.4 bits/letter, meaning that each character of the language carries 3.4 redundancy bits.

Shannon was also involved in developing a mathematical model describing the cryptographic systems. Recalling that a cryptanalyst ensures the perfect confidentiality, if from any message encrypted with it, is not possible to acquire any information on its plaintext message, Shannon has shown that this is only possible if the number of possible keys is at least as large as the number of possible messages. In practice, this means that the key must be at least as long as the message itself and must be changed from time to time: this is what happens with the single use tab which is the only system able to provide perfect security.

In general, cryptoanalysts use natural redundancies present in language to acquire valuable information about the encrypted message. For this reason, in order to eliminate redundancies, compression of the message takes place before encryption, at the same time reducing the size of the message and the time of encryption.

The entropy of a cryptosystem is a measure of the size of the space of the keys K and is approximately equal to $\log_2 K$: a cryptanalyst that uses an n-bit key is characterised by an entropy of n bits. In general, the greater the entropy of the system, the greater the difficulty in violating the system itself.

Given a message of length n, the number of different keys that can correctly decipher that language is $2^{H(K)-nD} - 1$. This magnitude was defined by Shannon as the distance of uniqueness U, also called point of uniqueness. This means that relatively short encrypted texts have a high likelihood of being deciphered in multiple plaintext messages of valid meaning in the language used, making it difficult for the attacker to choose the right one.

Distance of uniqueness U of a cryptosystem is defined as the magnitude $U = H(K)/D$. This distance of uniqueness provides the minimum amount of ciphertext for which there are high chances of correspondence with a single plaintext when a brute-force attack is used. For example, if a DES is used with 56-bit key and an ASCII message in English, the relative distance of uniqueness is equal to 8.2 ASCII characters, equivalent to 66 bits. The distance of uniqueness is not the measure of how much ciphertext is requested to perform successful cryptoanalysis but represents the amount of ciphertext required so that there is only one valid text by an operation of cryptoanalysis. A cryptosystem can be computationally difficult to crack even if it is theoretically possible to compromise it with a small amount of ciphertext. The distance of uniqueness is inversely proportional to the redundancy: if the latter tends to zero, a cipher that is not particularly robust can also become inviolable to a ciphertext-only attack.

A cryptosystem is said to have ideal secrecy, according to Shannon, if its distance of uniqueness is infinite.

The concepts seen are often used by cryptoanalysts as a starting point for their attacks. It should be remembered that the distance of uniqueness guarantees insecurity if it is too small but does not guarantee high security if it is great.

Shannon has defined two basic techniques to hide the redundancies of plaintexts that are represented by confusion and diffusion.

Confusion hides the relationship between plaintext and ciphertext. The easiest way to achieve this is to use a replacement operation as in the case of the simple Caesar cipher. In modern ciphers, an entire block of plaintext is replaced with a whole cipher block, suitably varying the replacement mechanism according to each bit of plaintext or of a key.

Diffusion disperses the redundancy of the plaintext spreading it evenly throughout the ciphertext. The easiest way to achieve this is to use a permutation operation as in the case of column transposition cipher.

Stream ciphers are based on a confusion mechanism although some systems of feedback also introduce diffusion. Block ciphers are based on the mechanism of both confusion and dissemination.

2.3.2 Complexity theory

Complexity theory is a methodology for the computational analysis of techniques and cryptographic algorithms. It is able to compare the different algorithms and different techniques to determine security. If, on the one hand, information theory says that any cryptographic algorithm can be breached, with the exception of single use tab, the complexity theory is able to tell us how long this would take.

The complexity of an algorithm depends on the computing power required to perform it. This computational complexity is measured through the use of two variables: the time complexity T and the spatial complexity S. Both variables depend on the size n of input.

The computational complexity of an algorithm is also expressed as "O", which represents the order of magnitude. If, for example, the time complexity of an algorithm is $5n^2 + 3n + 8$, its computational complexity is of the order of n^2 and is expressed as $O(n^2)$.

Using these parameters, the measure of the complexity becomes a value that is independent of the type of computer used, since it allows immediate calculation, as the input sizes affect the time and space required. For example, if $T = O(n)$, if the size of the input is doubled, the computation time is doubled; if $T = O(n^2)$, by adding a single bit in input, the computation time is doubled.

Algorithms are usually classified according to their spatial and temporal complexity: an algorithm is called constant if its complexity does not depend on n (notated $O(1)$); an algorithm is called linear if its complexity is equal to $O(n)$; an algorithm is called quadratic if its complexity is equal to $O(n^2)$; and so on. These algorithms are also called polynomial and their class of membership is called "polynomial time".

Algorithms whose computational complexity is equal to $O(t^{f(n)})$, where t is a constant greater than 1 and $f(n)$ is an appropriate polynomial, called exponentials. The subset of this class such that their complexity is equal to $O(c^{f(n)})$, where c is a constant and $f(n)$ is more than a constant but less than linear, is defined as "super-polynomial".

The best encryption algorithms from the viewpoint of resistance to compromise are ideally represented by those with exponential temporal complexity, while most algorithms of attack are characterised by super-polynomial time complexity but this does not exclude the fact that different class algorithms can be discovered.

Table 2.10 shows a summary of the various classes of algorithms and their performance.

If we consider a brute-force attack, the complexity of the attack itself is directly linked to the number of keys that must be tested an exponential function of the length of the key: if n is the length of the key, the computational complexity of such an attack is equal to $O(2^n)$.

Complexity theory attempts to define a minimum amount of space and time needed to solve a given problem using an ideal machine, called Turing machine, which represents a finite state machine with infinite storage capacity.

Table 2.10 Summary of the various classes of algorithms.

Class of the algorithm	Complexity	Number of operations ($n = 10^6$)	Time for 10^6 *O/S*
Constant	$O(1)$	1	1 μs
Linear	$O(n)$	10^6	1 s
Quadratic	$O(n^2)$	10^{12}	11.6 days
Exponential	$O(2)$	$10^{301.030}$	$10^{301.006}$ times the age of the universe

The problems that can be solved with algorithms in polynomial time are called treatable because they can be resolved, for reasonable inputs, in a reasonable amount of time. Problems that cannot be resolved with polynomial time are called intractable. The problems that can be solved only with super-polynomial algorithms are called computationally intractable. If an algorithm cannot be found for the solution regardless of the complexity time of a problem, then the same is called undecidable.

Using the concept of complexity, it is possible to divide and classify the various types of problems.

Figure 2.11 shows the various classes of complexity, with the mutual relations.

The lowermost part of Figure 2.11 shows the class of type *P* which contains all of the problems that are characterised by polynomial time. In the upper class is the *NP* class which contains all of the problems that are characterised by polynomial solution time using a non-deterministic Turing machine, that is able to carry out random solution attempts. The *NP* class is very important, since many symmetric algorithms and all public-key algorithms can be violated in a non-deterministic polynomial time as, given a ciphertext *C*, the system attempts a random text *X* with a key *K* and encrypts it, in a polynomial time, to see if the result coincides with *C*. The *NP* class includes class P, since the same Turing machine can be used where the non-deterministic component is removed. If all *NP*-type problems can be solved by a non-deterministic Turing machine then it can be said that $P = NP$. Even if it is obvious that the *NP* class problems are more complex to solve than *P* class ones, it has never been shown that $P \neq NP$, even if this is a popular belief. There is still an *NP* problem class that may be proven to be as difficult as every other issue of this class: the same class is defined *NP*-complete.

Above the *NP*-complete class is the PSPACE class, represented by those problems that can be solved in polynomial space with a time that is not necessarily polynomial. There is a class of problems, called PSPACE-complete, that is to PSPACE problems what *NP*-complete class is to *NP* problems.

Above all the classes is the class of problems, called EXPTIME, that can only be solved through exponential time.

2.3.3 Numbers theory

The basic elements of the numbers theory which are essential in the field of cryptography are illustrated in the following, referring the reader to specific texts for further details.

2.3.3.1 Modular arithmetic

Modular arithmetic is used at times when there are systems of periodical and circular counting, such as, for example, the time measuring system that is typically 24 modulo. In fact, if 7 hours must be added to 21 hours, the final result will be 4 and not 28, given $(7 + 21) \bmod 24 = 28 \bmod 24 = 4 \bmod 24$. It can also be said that 28 is equivalent to 4 modulo 24.

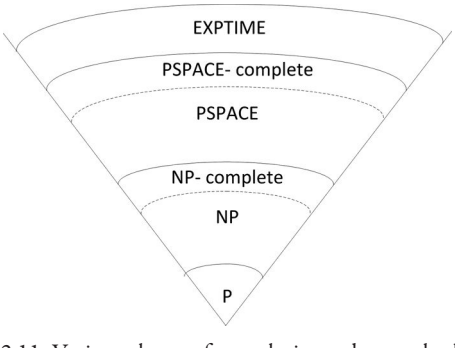


Figure 2.11 Various classes of complexity and mutual relations.

In mathematical terms, it can be said that $a \equiv b \bmod n$ if $a = b + kn$, where a and b are two numbers and k an integer. If a is non-negative and b ranging between 0 and n, then b is the remainder of the division between a and b is also called the residue of a modulo n.

The set of integers between 0 and $n - 1$ is said to be a complete set of residues, as for every integer a, its residue modulo n is a number between 0 and $n - 1$.

The operation $a \bmod n$ is also called modular reduction because it provides, as a result, the residue of a which is a number between 0 and $n - 1$.

Modular arithmetic enjoys the commutative, associative and distributive properties, such as normal arithmetic. Given three numbers a, b and c, the following properties therefore apply:

$$(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n \tag{2.10}$$

$$(a^{*}b) \bmod n = ((a \bmod n)^{*}(b \bmod n)) \bmod n \tag{2.11}$$

$$(a^{*}(b + c)) \bmod n = (((a^{*}b) \bmod n) + ((a^{*}c) \bmod n)) \bmod n \tag{2.12}$$

In the field of cryptography, with particular reference to public-key cryptography, intensive use of modular arithmetic is made. It is very practical in computer processing because it avoids the production of very large intermediate results as, if any operation of addition, subtraction or multiplication of k-bit modulo n is performed, the intermediate result will always be lower than $2k$ bits.

In addition, since the operations have a distributive property, if the exponent of a given number is to be calculated, all that is required is the performance of a series of modulo n multiplications, always working with limited intermediate results. For example, if we want to calculate $a^8 \bmod n$, instead of calculating $(a^{*}a^{*}a^{*}a^{*}a^{*}a^{*}a^{*}a) \bmod n$, the equivalent operation $((a^2 \bmod n)^2 \bmod n)^2 \bmod n$ can be performed. This technique can also be used for different elevations of power of multiples of 2, suitably decomposing the exponent. This type of operation is also called addition chaining and is able to reduce a more complex operation to an average of $1.5k$ simpler operations, where k is the length of the number in bits.

2.3.3.2 Prime numbers

A prime number is an integer greater than 1 that is not divisible by any other number except for 1 and by itself. For example, numbers 2, 3, 5, 7, 11, 13, etc., are prime numbers. Prime numbers are infinite and are used extensively in the field of cryptography, especially if they are very large.

2.3.3.3 Maximum common divisor

Two numbers are called relatively prime if they have no factor in common different from number 1. In practice, their greatest common divisor is equal to 1.

A technique to calculate the greatest common divisor of two numbers is the Euclidean algorithm described in his book dating back to AD 300.

2.3.3.4 Inverse modulo of a number

Given any number a, its inverse, referred to as a^{-1}, is a number such that $a^{*}a^{-1} = 1$.

If this operation is relatively simple in normal arithmetic, it is not possible to say the same in modular arithmetic, where the problem, given a certain number a and a number n, consists of finding x such that $(a^{*}x) \bmod n = 1$ in such a manner that $a^{-1} \equiv x \bmod n$.

It can be said that the problem admits solution if a and n are relatively prime between each other. If n is a prime number, then each number between 0 and $n - 1$ is relatively prime with n and possesses an inverse modulo n in this range.

A valid system for calculation of the inverse of a number a modulo n is represented by the Euclidean algorithm reverse, which will not be described in this section for the sake of brevity. This algorithm is iterative and can become very slow when the number is very high.

2.3.3.5 Resolution for coefficients

The Euclidean algorithm can be used for the so-called resolution for coefficients that consists of finding m coefficients $u_1, u_2, \ldots, u_m$ such that given m variables $x_1, x_2, \ldots, x_m$, the following equation is satisfied: $u_1x_1 + u_2x_2 + \cdots + u_mx_m = 1$.

2.3.3.6 Fermat's little theorem

Fermat's little theorem demonstrates that if m is a prime number and a is not a multiple of m, then $a^{m-1} \equiv 1 \bmod m$.

2.3.3.7 The Euler's totient function

There is also another method for the calculation of the inverse modulo n of a given number that can be used if certain conditions are satisfied.

A reduced set of modulo residues n is defined as the complete set of residues that are relatively prime with n. If n is a prime number, then the reduced set of modulo residues n is represented by all the numbers between 1 and $n - 1$, the zero not belonging to any of this set, for n any such set excluding the number 1.

The Euler's totient function, indicated by $\varphi(n)$, represents the number of elements that belongs to the reduced set of residues of n, that is the number of positive integers less than n that are relatively prime with n.

If n is a prime number, then $\varphi(n) = n - 1$. If $n = pq$ and p and q are prime numbers, then $\varphi(n) = (p - 1)(q - 1)$. The latter property is very useful as it is used in public-key cryptographic systems.

Thanks to Euler, it is possible to generalise Fermat's little theorem, being able to state that if the greatest common divisor between a and n is equal to 1, then $\alpha^{\varphi(n)} \bmod n = 1$ allowing calculation of $x = a^{-1} \bmod n = \alpha^{\varphi(n)^{-1}} \bmod n$.

Both the Euler method and the Euclid method can be used to solve the general problem $(ax) \bmod n = b$, being the greatest common divisor of a and n equal to 1.

In the case of Euler, we will have $x = (b\alpha^{\varphi(n)^{-1}}) \bmod n$, while in the case of Euclid we will have $x = (b(a^{-1} \bmod n) \bmod n$.

The Euclidean algorithm is typically faster than that of Euler, especially for numbers less than 500 bits.

What has been seen so far is valid if the greatest common divisor between a and n is equal to 1. If this is not true, there may be multiple solutions to the above problem or there may be no solutions at all.

2.3.3.8 Chinese remainder theorem

The Chinese Remainder Theorem is used to resolve an entire system of equations if factorisation of a number n in prime numbers is known.

If p_1, p_2, ..., p_m is the factorisation of n in prime numbers, then the equation system $(x \bmod p_i) = a_i$, con $i = 1, 2, \ldots, m$, has only one solution. In practice, this means a number (less than the product of certain prime numbers) that is uniquely identified by its modulo residues form those numbers first.

In general, given two numbers a and b such that $a < p$ and $b < q$, where p and q are prime numbers, there is a single number x, lower than the product pq, such that $x \equiv a \bmod p$ and $x \equiv b \bmod q$. To find this number x, the Euclid theorem is used to find the number u such that $uq \equiv 1 \bmod p$ and then $x = (((a - b)u) \bmod p)*q + b$ is calculated.

The Chinese remainder theorem can be used to solve the problem of finding a single number x, lower than the product pq, where p and q are prime numbers and $p < q$, such that $a \equiv x \bmod p$ and $b \equiv x \bmod q$. This shows that, under these conditions, there are two different solutions if $a > b \bmod p$ or $a < b \bmod p$. In the first case, $x = (((a - (b \bmod p))u) \bmod p)\, q + b$, while in the second case $x = (((a + p - (b \bmod p))\, u) \bmod p)\, q + b$.

2.3.3.9 Quadratic residues

It is said that a number a is a quadratic residue modulo p, where p is a prime number and a number greater than 0 and less than p, if the following equation is valid: $x^2 = a \bmod p$, for certain values of x. Not all the values of a meet this equality: in order for a to be a quadratic residue, modulo p must be the quadratic residue modulo of all the prime factors of p.

If p is equal, there are exactly $(p - 1)/2$ quadratic residues modulo p and $(p - 1)/2$ non-quadratic residues. In addition, if a is a quadratic residue modulo p, then a has two square roots: one between 0 and $(p - 1)/2$ and the other between $(p - 1)/2$ and $(p - 1)$, a quadratic residue modulo also being one of the two roots and for this reason is called main square root.

If n is the product of two prime numbers, p and q, then there are $(p - 1)\,(q - 1)/4$ quadratic residues modulo n. A quadratic residue modulo n is also a perfect square modulo n.

2.3.3.10 Legendre symbol

The Legendre symbol $L(a, p)$ is defined when a is an integer and p is a prime number greater than 2. It is equal to 0, 1 and −1. The following properties apply:

1. $L(a, p) = 0$ if a is divisible by p.
2. $L(a, p) = 1$ if a is a quadratic residue modulo p.
3. $L(a, p) = -1$ if a is not a quadratic residue modulo p.

$L(a, p)$ can be calculated as $L(a, p) = a^{(p-1)/2} \bmod p$.

2.3.3.11 Jacobi symbol

The Jacobi symbol $J(a, n)$ is the generalisation of the Legendre symbol and is applied to every integer a and every integer n. This function depends on a set of small residues of the dividers of n and can be calculated in different ways. One possible way is demonstrated as follows:

1. $J(a, n)$ is defined only if n is equal.
2. $J(0, n) = 0$.
3. If n is prime, then $J(a, n) = 0$ if n divides a.
4. If n is prime, then $J(a, n) = 1$ if a is a quadratic residue modulo n.
5. If n is prime, then $J(a, n) = -1$ if a is not a quadratic residue modulo n.
6. If n is composite, that is $n = p_1{}^*p_2{}^*\ldots{}^*p_m$ being $p_1, p_2, \ldots, p_m$ prime numbers, then $J(a, n) = J(a, p_1)^*J(a, p_2)^*\ldots{}^*J(a, p_m)$.

The Jacobi symbol cannot be used to determine if a is a quadratic residue modulo n unless n is prime.

2.3.3.12 Blum integers

Given two prime numbers p and q, both congruent to 3 modulo 4, then the product $n = pq$ is called Blum integer. This number has the property that each quadratic residue has four square roots, one of which is also a square that represents the main root square.

2.3.3.13 Generators

Given a prime number p and a g (number lower than p), then g is said to be a generator modulo p if for every number b variable between 1 and $p - 1$ there exists at least one number a such that $g^a \equiv b \bmod p$. This can also be expressed by saying that g is primitive compared to p.

To check whether a given number is a generator is not a simple task: it involves, instead, knowing the factorisation of $p - 1$. In fact, if $q_1, q_2, \ldots, q_m$ are the different prime factors of $p - 1$, if we want to ensure that g is a generator modulo p, it is possible to calculate $g^{(p-1)/q} \bmod p$ for all the values $q_1, q_2, \ldots, q_m$. If the result is equal to 1 for any one of q values q_i, then g is not a generator. On the contrary, if the result is not equal to 1 for any of these values, then g is a generator.

The search for a generator modulo p can be made by choosing a random number between 1 and $p - 1$ and checking if it is a generator.

2.3.3.14 Calculation in a Galois field

If n is a prime number or the power of a very large prime number, then we have a Galois field, indicated as $GF(p)$, where p is a given number. This field contains well-defined operations of addition, subtraction, multiplication and division for numbers that are non-null and the number 0 is an additive identity and number 1 is an multiplicative identity. Each non-null number possesses a unique reverse number, this is not the case if p is not prime. In this field, the commutative, associative and distributive properties are also valid.

The arithmetic in the Galois field is extremely useful for cryptography. To do this, the irreducible polynomial arithmetic modulo of n degree was introduced, whose coefficients are modulo q integers, where q is prime. These fields are referred to as $GF(q^n)$ and all the arithmetic is performed as modulo $p(x)$, since $p(x)$ is an irreducible polynomial. It should be noted that when polynomials are involved, the first term is replaced with the irreducible term: this means that a given polynomial, if irreducible, cannot be expressed as the product of other polynomials (with the exception of itself and the number 1).

For further details about the applications of Galois fields, the reader can refer to the bibliography section provided at the end of this book.

2.3.4 Factorisation

The factorisation of a number n represents the search operation of prime numbers that, multiplied, gives the value of the number itself. Factorisation is one of the historical problems of the number theory. This, in itself, is not very difficult but it is extremely difficult from the point of view of the calculation time. One of the most efficient algorithms is the number field sieve (NFS), which is the more efficient routing algorithm currently known for the factorisation of numbers of more than 110 digits. In its original version, it was not particularly useful but, over time, has followed a series of changes that has led it to be an extremely efficient algorithm.

2.3.5 The generation of prime numbers

Prime numbers are the basis of public-key cryptographic algorithms.

Given the intensive use that is made of prime numbers, it might be feared that their number may be limited, seriously compromising the security of the algorithm that uses them, but that is not true because in a number of 512 bits there are 10^{151} prime numbers that represent an extremely high value. For a given number n, the probability that a number close to n is prime is equal to $\ln(n)$: this means that the total number of prime numbers less than n is equal to $n/\ln(n)$ which represents an extremely high value, making the probability extremely low that two people randomly choose the same prime number.

To use prime numbers, a set of algorithms that are able to generate them efficiently must be developed. The wrong way to create this is to generate a random number and then try to factor, while the right way is to generate a random number and then check if this is primary.

There is currently a series of algorithms, which uses the knowledge of the theory of numbers shown above, which are capable of efficiently generating prime numbers. The algorithms are named after their discoverers: Solovay–Strassen, Lehmann and Rabin-Miller.

There are prime numbers that are called strong prime numbers as they enjoy some of the properties that make factorisation of their product that is extremely difficult if certain algorithms are used. If p and q are the relevant prime numbers and $n = pq$, these properties are as follows:

1. The greatest common divisor between p and q should be very small.
2. Both $p - 1$ and $q - 1$ should have very large primary factorisations, respectively p^* and q^*.
3. Both $p^* - 1$ and $q^* - 1$ should have very large primary factorisations.
4. Both $p + 1$ and $q + 1$ should have very large primary factorisations.
5. Both $(p - 1)/2$ and $(q - 1)/2$ should be prime.

These properties ensure that at least the old existing algorithms encounter difficulty in factorisation of the product of $n = pq$. New algorithms were however discovered that are able to overcome these difficulties.

2.3.6 Discrete logarithms in finite fields

Another one-way function frequently used is the modular exponentiation, which presents $a^x \bmod n$ mathematical expression, where a, x and n are appropriate numbers.

The inverse of this problem is to find the discrete logarithm of a number that represents a problem with a not particularly easy mathematical solution.

There are three main groups whose discrete logarithms are of interest to cryptography, which are:

1. the multiplicative group of the $GF(p)$ primary fields;
2. the multiplicative group of finite fields of $2GF(2^n)$ characteristic;
3. the group of Elliptic Curves over $EC(F)$ finite fields.

Since the security of the majority of public-key algorithms is based on the research problem of discrete logarithms, this problem has been intensively studied.

If p is a prime number, then the complexity of finding a discrete logarithm in $GF(p)$ is the same level in the factorisation of an integer n of the same size, where n is the product of two prime numbers of the same length.

Calculation of discrete logarithms is closely linked to factorisation: if the first problem can be solved then it is also possible to factor (the inverse has not yet been mathematically demonstrated).

2.4 Data Encryption Standard

DES, also known as Data Encryption Algorithm (DEA), has been used throughout the world for several years. Even if a little dated, it continues to be a good algorithm.

In 1972, the American National Bureau of Standards (NBS), which later became the National Institute of Standards and Technology (NIST), implemented a program to protect computers and communications with the objective of providing a cryptographic standard. In May 1973, it published a public proposal request for a cryptographic standard algorithm according to the following criteria:

1. The algorithm should ensure high standards of security.
2. The algorithm should be completely specified and easy to understand.
3. Security of the algorithm should be based on the key and not on the secrecy of the algorithm.
4. The algorithm should be available to all users.
5. The algorithm should be adaptable for use in various applications.
6. The algorithm should be economically implementable in electronic devices.
7. The algorithm had to be efficient in use.
8. It should be able to validate the algorithm.
9. The algorithm should be exportable.

From the responses obtained, it was understood that interest in the field was high but there were few experts in the field and none of the proposals received was able to meet all the specifications.

NBS decided to publish a second request in August 1974, receiving a valid proposal based on Lucifer, an algorithm developed by IBM in 1970. This apparently complex algorithm used simple logic operations operating on bits and could be implemented without major problems on hardware devices. NBS asked for the opinion of NSA and, after having performed the due subsequent evaluations, approved this algorithm in March 1975. NSA reduced the length of the keys from 128 bits to 56 bits, and the general public feared that the same there had introduced the trap doors to facilitate decryption in case of need.

DES was finally adopted as a federal standard in November 1976.

The American National Standards Institute (ANSI) approved the use of DES as an algorithm for private use in 1981, calling it the Data Encryption Algorithm (DEA).

2.4.1 The DES algorithm

The DES is in practice a block encoder that operates on blocks of 64 bits and is a symmetrical type cipher. The key length is 56 bits and the entire security of the algorithm is based on the key. Essentially, the algorithm is based on two fundamental concepts: confusion and diffusion (described above). The basic building block of DES operates a combination of these techniques, first performing a substitution and then a permutation. This operation is also called round. DES performs 16 rounds, executing the same sequence 16 times on plaintext in order to encrypt it.

The algorithm performs only arithmetic and logical operations on groups of 64 bits and their repetitive nature makes it easily implementable on dedicated hardware.

DES operates according to the operating diagram shown in Figure 2.12.

From Figure 2.12 it can be seen how the algorithm, after an initial permutation, divides the 64-bit block into two parts, a right half and a left half, each 32 bits long. Subsequently, six identical round operations, called *f* functions, are performed, during which the data are combined with a key. After following these operations, the right and left halves are put together and a final permutation is performed that is inverse with respect to the initial one.

The process that takes place during every round is shown schematically in Figure 2.13.

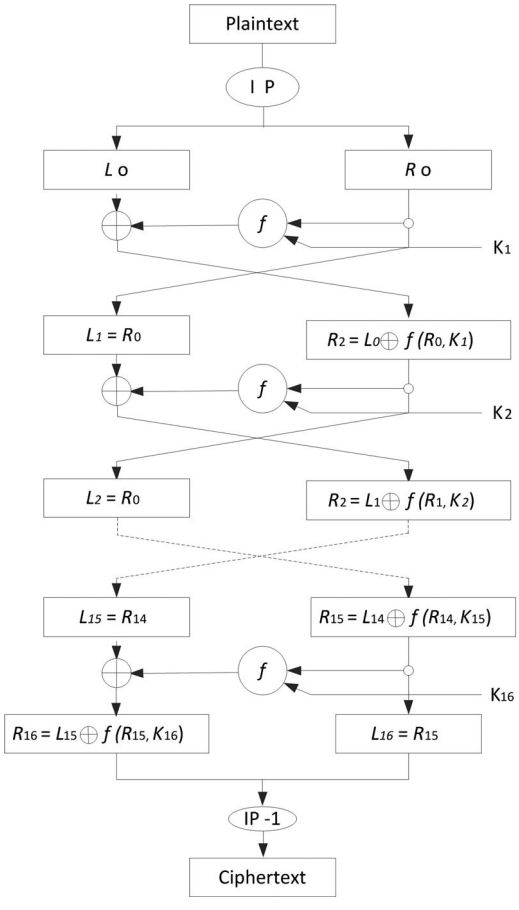


Figure 2.12 DES operating schema.

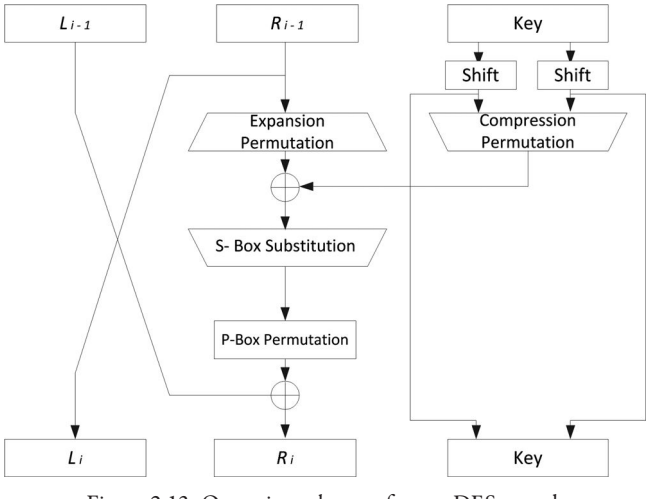


Figure 2.13 Operating schema of every DES round.

Table 2.11 Initial permutation of DES.

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Figure 2.13 shows how to shift the key in each round and 48 bits of the 56 keys are selected. The right half of the data is expanded to 48 bits through a permutation expansion, combined with 48 bits of the shifted key and the key permutated through an XOR, sent in the S-boxes (which will be explained later), producing 3 bits, and permutated again. These four operations are the so-called f function. Output of the function f is combined with the left half through an XOR operation. The result of such operation becomes the new right half while the old right part becomes the new left half. This operation is performed 16 times, performing 16 rounds.

If B_i is the result of the ith iteration, L_i and R_i represent, respectively, the right and left halves of B_i and K_i, the 48-bit key represents the round i and f represents the function that performs all the replacements, the permutations and the XOR operations with the key and as such each round can be mathematically written as:

$$L_i = R_{i-1} \tag{2.13}$$

$$R_i = L_{i-1}(+)f(R_{i-1}, K_i) \tag{2.14}$$

The initial permutation is followed before the first round, by transposing the input block as indicated in Table 2.11.

Table 2.11, as well as the remaining tables in this section, should be read from left to right and from upwards to downwards. Using this convection, it can be seen how the initial permutation moves the plaintext of 58 bits into first position, the 50-bit plaintext into second position and so on. Initial permutation is not an essential element of DES security.

Now let us look at details the surrounding key transformation. It is initially reduced to 48 bits ignoring every eighth bit, as shown in Table 2.12.

These bits can be used as a parity check to ensure the key is free of errors. After 56 bits are extracted, a sub-key of 48 bits for each of the 16 rounds is generated. Each sub-key k_i is determined by dividing the 56-bit key into two 28 bits halves and then working a shift to the left of 1 or 2 bits, according to the round considered, as shown in Table 2.13.

Table 2.12 Permutation of the DES key.

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Table 2.13 Number of bits in the key shifted depending on the round.

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Table 2.14 Permutation compression.

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

After permutation is performed, 48 of the 56 bits are selected. Since this operation changes the order of the bits and selects a subset, this is also called compression permutation. This operation is shown in Table 2.14.

It can be seen how 14 bits of the key shifted are moved to position 1, 17 bit of the key shifted into position 2 and so on.

Thanks to the shifting operation, a different subset of key bits is used for each sub-key. Each bit is used on average in 14 of the 16 sub-keys even if not all bits are used for the same number of times.

We come now to the description of the operation of the said expansion permutation. This operation expands the right half of the R_i, where data i ranges from 32 to 48 bits. Because it exchanges the order of the bits and repeats a number of bits, it assumes the name of expansion permutation. This operation has two purposes: to give the right half the same dimensions as the key for the subsequent XOR operation and to provide a more lengthy result for the subsequent compression. The main advantage of this operation, from a cryptographic point of view, consists of and allows every single bit to influence two substitutions in such a manner that the dependence of the output bits on the input bits spreads more rapidly, giving rise to what is defined as avalanche effect. The DES is designed in such a manner that, at the end of the process, each output bit of the ciphertext depends on every bit of the input plaintext and on each bit of the key. Figure 2.14 shows a diagram of expansion permutation. This expansion is also called E-box.

From Figure 2.14 it can be seen how, for every 4 bits of the input block, the first and fourth bits represent two bits of the output block while the second and third bits represent a bit of the output block. Table 2.15 shows the correspondence between the input and output bits. Although the output block is larger than the input block, each block of input generates a single output block.

After the XOR operation between the compressed key and the expanded block is performed, the resulting 48 bits are sent to the stage that performs the replacement operation. This operation is performed using eight boxes, called S-box. Each S-box is characterised by six input bits and four output bits and eight S-boxes are used to perform this replacement operation. The 48 bits are divided into eight sub-blocks of 6 bits and each block is sent to a different S-box: the first block is sent to S-box number 1, the second to S-Box number 2 and so on, as shown in Figure 2.15.

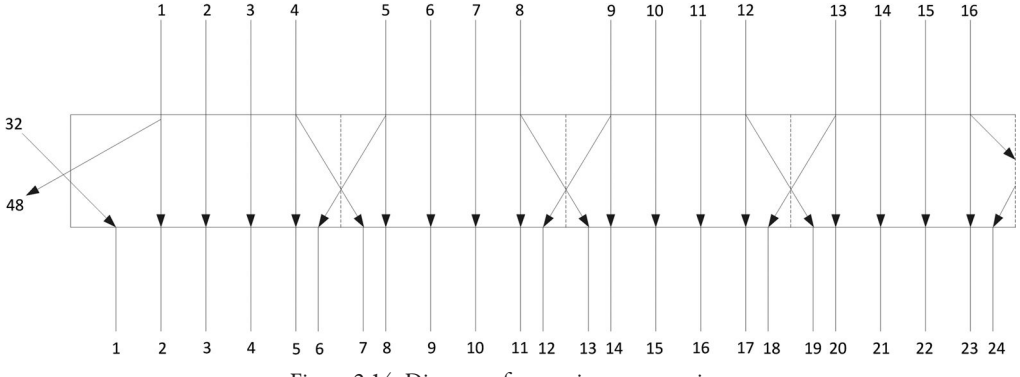


Figure 2.14 Diagram of expansion permutation.

Table 2.15 Permutation expansion.

32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1

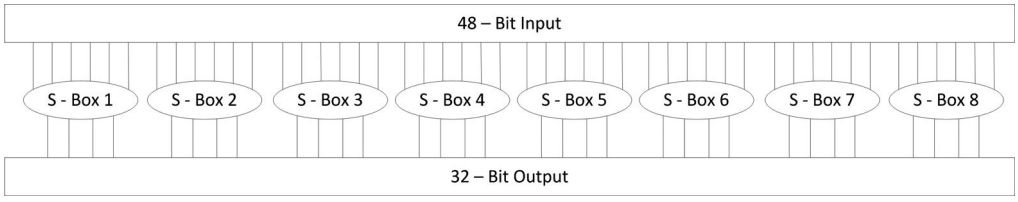


Figure 2.15 Replacement stage via S-box.

Each S-box represents a table with four rows and 16 columns and each input in the box is represented by a 4-bit number. The six input bits of the S-boxes indicate the row and column under which positioning should occur in order to read the relative output. The completed tables of the S-box are indicated in Table 2.16.

The replacements made by the S-boxes are the most critical phase and, at the same time, the strength of the DES, involving typically non-linear operations.

The result of this replacement via S-boxes is a series of 4-bit blocks which are recombined into a single 32-bit block that is sent to the next stage called permutation by P-box.

The P-box stage of permutation maps each input bit into an outlet position without using the same bit twice and in any case using all the bits. The permutation map is shown in Table 2.17.

The XOR operation is executed at the end of this permutation operation with the left half of the initial block of 64 bits. After this the right and left halves are switched and a new round begins.

At the end of the round cycles, the final permutation indicated in Table 2.18 is performed.

The great advantage of DES is that, despite the large number of operations performed, the operation of decryption takes place in the same manner, returning the original message in plaintext.

Another great advantage is the option of encrypting the message several times using several keys. To decrypt it, the same keys must be used in reverse order, from the last to the first used during encryption.

2.4.2 Security of DES

DES has been studied extensively and a number of points are still unclear such as the S-box. In this sense, just consider that these contain a hidden trap door to facilitate the operation of decryption by the national authorities, without necessarily having to know the key.

Let us now analyse the keys in more detail. First, it is essential to emphasise those keys that are called weak keys, due to the process by which the initial key is modified to obtain a sub-key for each round that composes the algorithm. In fact, the initial key is divided into two halves and each half is shifted independently: if all the bits of each half are 1 or 0, the same sub-key for each round will be generated. This may happen if the key is composed of all 1s or 0s or simply if the two halves are composed of all 1s or all 0s.

In addition, some pairs of keys may be able to encrypt plaintext in an identical ciphertext: in practice, a key of the pair can decrypt the encrypted message with the other pair. These keys are called semi-weak.

Table 2.16 S-boxes.

S-box 1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	1	11	9	5	3	8
4	1	4	8	13	6	2	11	15	12	9	7	3	10	5	0
15	2	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S-box 2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S-box 3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S-box 4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S-box 5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S-box 6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S-box 7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S-box 8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Table 2.17 P-box permutation.

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

Table 2.18 Final permutation.

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

In addition, some keys only produce four sub-keys, each used four times for the algorithm. These keys are called possibly weak keys.

The total number of non-optimum keys, considering the weak, the semi-weak and possibly weak ones, is 64, an extremely small number compared to the 2^{56} keys available, using 56 bits of key and the probability of generating them by a random process is extremely low.

Now let us look at the concept of complement keys. When running the complement bit to bit of a key and that complement key is used to encrypt the complement of the message in plaintext, the DES will return the complement of the encrypted message. This property is called complementation and is due precisely to the manner in which the operation of encryption within the algorithm is carried out. This also means that if a chosen-plaintext attack is carried out against the DES, the number of keys to be tried drops from 2^{56} to 2^{55}.

Now let us look at the reason behind the choice of 16 rounds. In fact, after five rounds, each bit of the encrypted text becomes a function of every bit of the input text and every bit of the key while after eight rounds the ciphertext becomes a totally random function of each bit of the plaintext and each bit of the key. With time, the variants of the DES with a small number of rounds have been attacked with success and it has been proved that the DES with a number of rounds less than 16 can be violated with a known-plaintext attack more efficiently than a brute-force attack.

2.4.3 Differential and linear analysis

In 1990, an attack technique was developed called differential cryptoanalysis and it was proved that a chosen-plaintext attack performed against the DES is more efficient compared to a brute-force attack.

To do this, appropriate pairs of encrypted text attack are used whose plaintext presents well-determined differences, observing the evolution of these differences while the plaintext is processed by the several rounds of the DES when using the same key. The two texts may be chosen in a random manner as long as they meet certain differences. Using such differences in ciphertext, what emerges is that different probability can be assigned to different keys and analysing a large number of pairs of keys those most likely will tend to be identified.

Figure 2.16 shows a diagram of a DES round.

Suppose that two texts X and X^* are characterised by a certain difference ΔX. These texts correspond to the encrypted texts Y and Y^*, which are characterised by the difference between Y and ΔY. Since both the expansion permutation and the P-box are known, ΔA and ΔC are also known. B and B^* are also known and their difference is equal to ΔA. For each value of ΔA and not all the values of ΔC are equally probable and the combination of ΔA and ΔC suggests bits for ΔA XOR K_i and ΔA^* XOR K_i. Since A and A^* are known, this provides information about K_i.

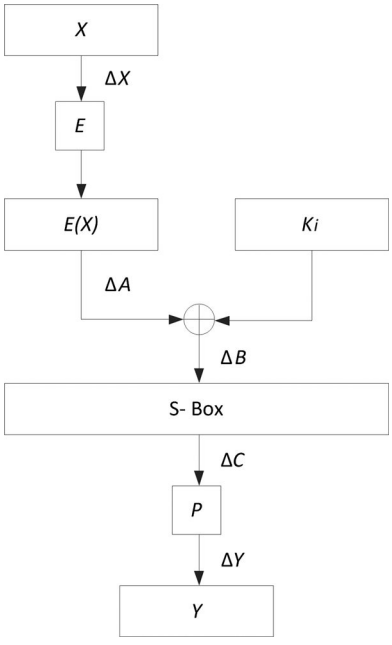


Figure 2.16 Diagram of a DES round.

If we are able to identify the key K_{16} of the last round of DES, then the 48 bits of the key are obtained: the other 8 bits can be obtained with brute force.

Since certain differences in the pairs of plaintext have a high probability of generating certain specific differences in the corresponding encrypted texts, these differences are called characteristics. These characteristics extend through a certain number of rounds and in practice define a path through them. There are, therefore, differences in input, differences at each round and the differences in output, all characterised by a specific probability.

It is possible to find these characteristics by generating a table where the rows represent the possible XOR of the inputs, the columns represent the possible output XOR and the values represent the number of times that a particular value of XOR is for a given XOR input. Such table can be generated for each of the 8 S-boxes of DES.

A pair of plaintexts that meets the characteristics is called right text while, if not is called wrong text: a right pair suggests the correct round key while a wrong pair suggests a random round key.

To find the correct round key, all that is needed is to operate a reasonable number of attempts in such a way as to be able to verify that a particular key occurs with greater frequency than others.

The differential attack, in its basic version, at a given round of DES allows tracing to the sub 48-bit keys of this round, while the remaining 8 bits must be obtained by brute force.

Differential cryptoanalysis operates correctly against the DES and other similar algorithms that use the S-boxes, depending on the outcome of the attack from the structure of the S-boxes.

The resistance of the DES can be improved by increasing the round number and differential cryptoanalysis requires the same time as a brute-force attack for a system characterised by 17 to 18 rounds. For 19 rounds this becomes impractical because more than 2^{64} plaintexts are required.

It is important to emphasise that this type of attack is more theoretical than practical, since the high amount of time and necessary data are beyond the reach of most attackers.

Another type of attack, called related key, focuses on the differences between the keys rather than on the differences between plaintexts. In this case, we choose a relationship between pairs of keys

without knowing the keys themselves and the data are encrypted with both keys. After this, a known-plaintext or a chosen-plaintext attack is performed.

Another type of attack is linear cryptoanalysis, which uses the linear approximations for modelling the operation of a block cipher like the DES.

Over the years time advanced techniques have been developed that use together the differential with the linear techniques and for this reason are called differential – linear cryptoanalysis.

After differential cryptoanalysis entered the public domain, IBM published the criteria that underlie generation of the S-boxes.

2.4.4 DES variants

A variant of DES is called triple DES. It uses three steps of DES and, in this case, a brute-force attack requires the trying of 2^{112} keys rather than 2^{56} keys Figure 2.17.

Another variation makes use of a different sub-key for each round instead of a general single 56-bit key. The resulting algorithm for this is called DES with independent sub-keys. Because there are 16 rounds, each using a 48-bit key, the total length of the key should be 768 bits and this fact brings the attempts of a brute-force attack to 2^{768} attempts even if a man-in-the-middle attack could reduce the number of attempts to 2^{384}, in any case an extremely high number.

Another variant of DES is the so-called DESX. DESX uses a technique called whitening to hide the inputs and outputs from DES. In addition to the 56 bits of the DES, it uses a further 64-bit whitening key that is used to perform an XOR operation with plaintext before commencing the real DES sequence. A further XOR operation is performed on the ciphertext after the last round, using a suitable hash function of the total 120-bit key. DESX is much more resistant than DES against brute-force attacks; if there are n known texts, the number of operations to conduct an attack is equal to $(2^{120})/n$. In addition, it is more resistant to attacks by differential and linear cryptoanalysis.

Another variant is DES generalised, or GDES, designed to increase the calculation speed and robustness. The schema is shown in Figure 2.18.

GDES operates on blocks of variable length in plaintext. Encryption is performed by dividing the blocks into q sub-blocks of 32 bits, where q, in general, is equal to the number of blocks divided by 32. Function f is calculated once for each round on the block farthest to the right. After this, an XOR operation is performed with all the other parts that are then rotated to the right. This algorithm is characterised by a variable number n of rounds.

Two other variants are DES with alternated S-boxes and DES with S-box key dependents. In the latter case, the S-boxes, varying with the key, make any differential or linear type cryptoanalytic attack extremely difficult.

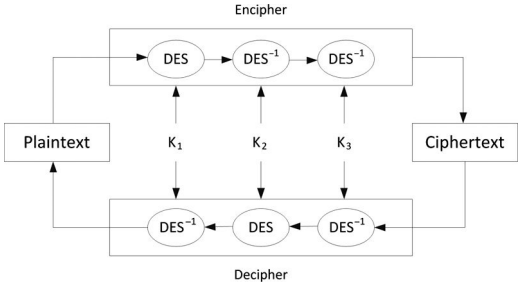


Figure 2.17 Diagram of triple DES.

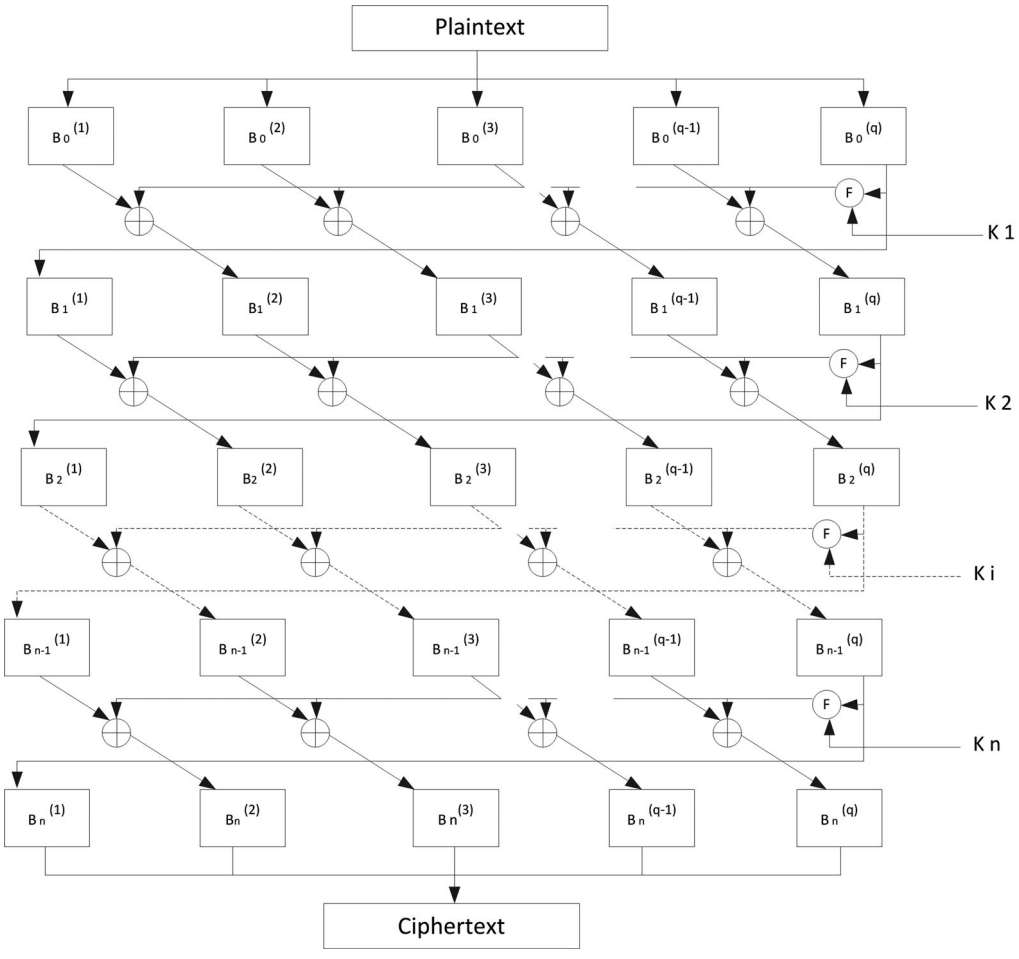


Figure 2.18 GDES operating schema.

2.5 Other block ciphers

In the current state, there are many other block ciphers that are not described in this section for reasons of space, but for more details the reader can refer to the bibliography section provided at the end of this book. The main block ciphers are: Lucifer, Madryga, NewDES, FEAL, REDOC, LOKI, Khufu, Khafre, RC2, IDEA, MMB, CA-1.1, Skipjack, Gost, CAST, Blowfish, Safer, 3-way, Crab, SXAL8/MBAL and RC5.

2.6 Cipher combination

Block ciphers can be combined in many ways to give rise to new safer algorithms without having develop new systems for the purpose.

Multiple encryption represents one possible technique in which an encryption algorithm is used several times to encrypt the same text each time using a different key.

Cascading is a technique similar to multiple encryption but, in this case, different algorithms are used in cascade instead of the same algorithm, always with different keys.

It is very important, for the purposes of reducing the possibility of attack by brute force, to use different keys for each encryption operation in cascade.

2.6.1 Double encryption

Double encryption in fact involves encryption of text twice, each time using a different key. The first time we encrypt with a key and the second time we encrypt with a second key. To decrypt, first the second key is used and then the first key. If n is the length of the key in bits, a brute-force attack would require 2^{2n} key attempts rather than 2^n keys.

2.6.2 Triple encryption

Triple encryption operates in three steps using only two keys. The first time we encrypt with a key, the second time with the second key and the third time with the first key.

A variant of this method is to encrypt with the first key, decipher with the second key and then encrypt again with the first key. For decryption a reverse sequence is used. This technique is also called encrypt-decrypt-encrypt (EDE) mode.

To increase the level of security, three keys are usually required.

2.6.3 Whitening

Whitening is referred to all the times that we perform an XOR operation between a given key and plaintext, before the latter is sent at an encryption stage. This technique is used, as we have seen, in DESX. This technique prevents a cryptoanalyst from obtaining valid pairs in plaintext/ciphertext forcing the same to guess not only the key from the algorithm but also the whitening data.

2.6.4 Cascading

The technique of cascading involves the use of multiple encryption algorithms in cascade. It is very important that the keys used for the various stages of the cascade are different.

During a chosen-plaintext type attack, a cascade is at least as difficult to crack because it is difficult to violate each stage that composes it.

Cascading is very useful when two parties do not trust the algorithm used by the other person. In this case, both of these algorithms can be used in cascade, with different keys, to reassure both parties.

2.7 Pseudo-random sequence generators and flow ciphers

The importance of random and pseudo-random sequence generators for key generation and for use with flow ciphers has already been discussed. This topic will be explored in greater depth in this section.

2.7.1 Congruent linear generators

A congruent linear generator is a pseudo-random sequence generator that can be expressed by the following formula:

$$X_n = (a\,X_{n.1} + b) \bmod m \tag{2.15}$$

where X_n is the nth number in the sequence, Xn_{-1} is the $(n - 1)$-th number in the sequence and a, b and m are constants. The constant a is also caller multiplier, b the increase and m the modulo. The key, also called seed, is represented by the initial value X_0.

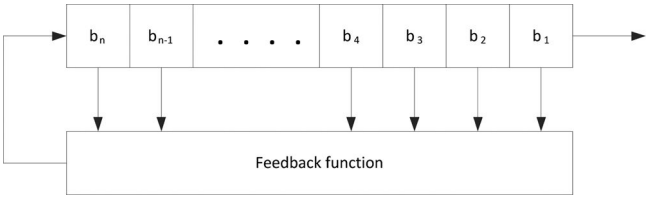


Figure 2.19 Scheme of a feedback shift register.

This generator is characterised by a period of not more than m. If a, b and m are suitably selected, then the generator will be characterised by a maximum period.

The advantage of congruent linear generators is their speed, requiring few operations per bit. The disadvantage is their predictability that makes them unsuitable for cryptographic applications. However, they remain suitable for simulations, showing a good statistical behaviour compared to most of the empirical tests.

It was also attempted to combine between them various congruent linear generators, obtaining longer periods and better results in statistical tests. However, they proved unfit for cryptographic applications.

2.7.2 Linear shift records with feedback

Shift registers are used intensively in cryptographic applications and in encoding theory. Flow ciphers based on shift registers have always been used intensively by military cryptography.

Feedback shift register is composed of a shift register and a feedback function, as shown in Figure 2.19.

A shift register is no more than a sequence of bits: each time a bit is required, all the bits of the register are shifted to the right and the necessary bit is extracted, while a new bit is inserted to the left calculating according to all the other bits in the register. The register period is defined as the maximum not periodic sequence that can be generated from this.

In stream ciphers, intensive use is made of shift registers because they can be easily implemented via hardware.

The simplest type of shift register is linear feedback shift register (LFSR), shown in Figure 2.20.

In this register, the feedback function is a simple XOR of a number of bits of the LFSR. The set of these bits is called *tap* sequence. Owing to the simplicity of the considered structure, a great deal of mathematical theory can be used to analyse it.

An n bit LFSR can be found in one of its possible $2^n - 1$ states which means that it can, if well designed, generate a sequence of $2^n - 1$ bits without repetition. In order for an LFSR to be able to generate sequences according to its maximum theoretical period without repetition, the polynomial generated using the *tap* sequence plus 1 must be a primitive polynomial modulo 2. The level of the polynomial is equal to the length of the shift register.

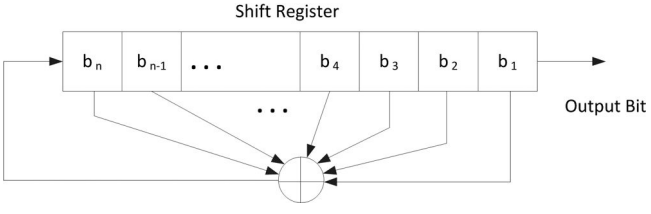


Figure 2.20 Diagram of a linear feedback shift register.

In general, there is an algorithm for generating a primitive polynomial modulo 2 of a certain level. The easiest way to do this is to select a polynomial at random and check if it is primitive.

If it concerns a polynomial which does not use all the coefficients, this is known as sparse polynomial. Sparse polynomials are considered weak and increase the level of vulnerability of the algorithm that uses them. For this reason, the so-called dense polynomials are preferred which are polynomials that use many coefficients.

LFSR is a class of pseudo-random sequence generators but has, however, non-random characteristics. Among other features, it must be pointed out that the long sequences of numbers generated are highly related and in some cases not entirely random. Despite this, LFSR are often used in encryption algorithms.

LFSR can also be implemented via software, even if this makes them slower.

2.7.3 Design and analysis of stream ciphers

Almost all of the stream ciphers base their operation on LFSR. Stream cipher analysis is usually easier to use than block ciphers, and since the first are based on LFSRS, the latter represents an object of analysis. A very common parameter for their analysis is represented by linear complexity, defined as the length n of the shorter LFSR that behaves like the register object of analysis. Each sequence generated by a finite state machine of a finite field is characterised by a completed linear complexity. Linear complexity is very important, since there are algorithms that can be traced to the scheme of operation of an LFSR after having observed only $2n$ bits of the output stream, and can easily violate the stream cipher that uses it.

It should be stressed that a very high non-linear complexity does not ensure a high level of security but can in any case be confirmed that a low linear complexity produces a low level of security.

To obtain a high non-linear complexity usually requires several generators, combining their respective outputs in a non-linear manner. The risk of such a solution is the possibility that the output sequence of one or more generators that make up this system may be related and the tools of linear algebra can be used to conduct an attack that, for this reason, is defined correlation attack.

2.7.4 Stream ciphers based on LFSR

It has already been stated that stream ciphers base their operation on LFSR. One or more LFSR are usually used, combined preferably in a non-linear manner, characterised by different lengths and polynomials of different feedback. If all the lengths are relatively prime between each other and all the polynomials are primitive, then the length of the overall generator will be the maximum. The algorithm key is used as the initial state of the various LFSR. Each output bit is a function of some or all of the bits of the LFSR: this function is called combining function and the entire generator is called generator combination.

There are many types of LFSRS that will not be discussed in this section for reasons of space, referring to the bibliography section provided at the end of this book for further information. The main types are Geffe generator, generalised Geffe generator, Jennings generator, stop-and-go generator, alternating stop-and-go generator, stop-and-go bilateral generator, threshold generator, auto-decimated generator, inner product multi-speed generator, sum generator, random dynamic sequence generator, Gollmann cascade, squeeze generator and auto-squeeze generator.

2.7.5 A5 stream cipher

The A5 stream cipher is used to encrypt radio mobile communications performed through the Global System for Mobile (GSM) system. It is used to encrypt the connection between the mobile terminal

and the radio base-station to which the same is linked. The rest of the connection is not encrypted, making the interception of calls on this section possible for phone companies or any attacker.

A5 is composed of three LFSR, whose length is equal to 19, 22 and 23 and whose polynomials are sparse. The output of the overall generator is the XOR operation of the three outputs.

An attack against this system that requires 2^{40} ciphers has been developed.

This algorithm is, however, of moderate quality, being very efficient and having passed all the major statistical tests. Its only weakness is the brevity of extension of the shift registers on which it is based, which makes it possible to have an exhaustive search of the keys.

2.7.6 Additive generators

Additives generators are very efficient because they produce random words and non-random bits. They can be used as components of complex generators.

The initial state of the generator is represented by an array $X_1, X_2, \ldots, X_m$ of n bit words that represents the initial state of the key. The ith word is calculated by the generator as:

$$X_i = (X_{i-a} + X_{i-b} + \cdots + X_{i-m}) \bmod 2 \tag{2.16}$$

where the coefficients $a, b, \ldots\ldots\ldots\ldots, m$ are appropriately selected to ensure that the period is the maximum possible and equal to $2^n - 1$.

The main additive generators, not described for reasons of space, are: Fish, Pike and Mush.

2.7.7 PKZIP

Even the well-known encryption program PKZIP is equipped with an encryption program that is substantially a stream cipher that encrypts a bit at a time. Without going into the details of implementation, it can be argued that this cipher does not guarantee a high degree of security, being required from 40 to 200 bytes of plaintext to conduct an attack characterised by a time complexity equal to 2^{27}.

2.7.8 Design of stream ciphers

The design of stream ciphers is very similar to that of block ciphers.

There are basically four different approaches for the design of a flow cipher:

1. System-theoretical approach that tries to ensure that the designed algorithm generates a difficult and unknown problem for any cryptoanalyst, using a set of rules and design criteria.
2. Information-theoretical approach that tries to keep the cryptoanalyst in the dark about the plaintext. Apart from the work done, the cryptoanalyst will never be able to find a single solution.
3. Complexity-theoretic approach that seeks to base the algorithm on known unresolved or difficult to solve mathematical problems, such as the factorisation of prime numbers or discrete logarithms.
4. Random approach that seeks to create a big problem that obliges the cryptoanalyst to examine a huge amount of useless data during his/her attempt to attack.

2.7.9 Generation of multiple streams from a single pseudo-random generator

When it is necessary to have multiple streams at the same time, such as when multiple channels of communication are involved, it is necessary to resort to several generators. This approach leads to an intensive use of hardware and the need to synchronise, between them, the various generators.

In some cases, we may have recourse to only one generator as in the diagram shown in Figure 2.21.

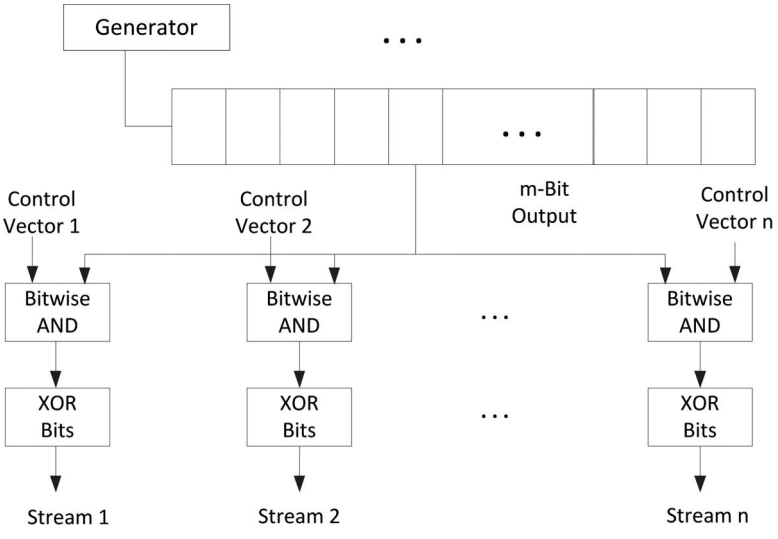


Figure 2.21 Multistream generator.

In such a layout, the output stream of a generator of proven security is used in input, which is sent in an *m* bit shift register. At each clock impulse, the register performs a shift to the right. Then, for each stream of desired output, it performs the AND operation between the m bits of the register and the m bits of a control vector that represents the identifier of the output channel. After this operation, the XOR of all the output bits is performed, thereby obtaining a single bit that is the outlet for that channel.

2.8 Real random sequence generators

The generators of pseudo-random sequences, although of high security, may not be sufficient for cryptographic applications.

Even when attempting to generate random sequences by means of pseudo-random generators, it is sufficient for an attacker that comes into possession of a copy of the generator and the relative key to be able to generate the same sequence, thereby violating the system.

On the contrary, a random generator may not be in any way reproduced in the operation. It is able to reproduce a bit sequence with the same statistical characteristics of random bits and it is not reproducible.

The main systems used to create random generators are random noise, computer clock, latency of beating on the keyboard, polarisation and correlation, and distillation of randomness.

2.8.1 Random noise

The best way to have a random source is to draw on the real world. In any case, to do this, dedicated hardware is required: sometimes a simple computer can be used.

We can, for example, think of a real phenomenon that happens on a regular basis but in a random manner. We can then measure the time that elapses between the first event and the second event, between the second event and the third event and so on. Subsequently, if the first measured interval is greater than the second one, an output bit equal to 1 is produced, and vice versa an output bit is

produced. If this operation is performed for all measured events, a perfectly random sequence will be obtained in output.

With regard to hardware devices, devices based on capacity are built with metals and insulators on the frequency instability of a free oscillator, on the extent of radioactive decay, on the thermal noise of a diode junction and so on.

A random number generator has also been developed that uses as a base mechanism the time required by a computer disc drive to read a block of data on the disc itself, using the differences in the time of access as source. Implementation of this system has led to a generator capable of producing 100 bits per minute.

2.8.2 Computer clock

If a limited number of random bits is required, computer clock register can be used.

The generation of bits with this system must not, however, be exceeded because the same would, if in excessive number, not be perfectly random. To do this, a sub-routine can be created that, at regular intervals, accesses the clock register on the computer and removes the least significant bit. This mode of operating is very sensitive to the randomness of system interruptions and granularity of the clock.

2.8.3 Keyboard latency typing

When we strike the keys of a keyboard of a computer, the sequence that follows is both random and non-random. By measuring the time between successive strikes and taking the least significant bit of this measurement, a fairly random bit sequence is obtained.

This technique is fairly limited given the limited speed with which the keys on a keyboard are struck.

2.8.4 Polarisation and correlation

Even if using a random process for the generation of random bits, given that instrumentation is used due to its size, a polarisation error is always introduced that influences the randomness of the generated sequence.

To eliminate this polarisation, an XOR operation can be performed on several bits together. If a random bit is polarised biased towards 0 by a factor e less than 1, its probability $P(0)$ of being 0 can be written as:

$$P(0) = 0.5 + e \tag{2.17}$$

If running the XOR operation between two of these bits, we get:

$$P(0) = (0.5 + e)^2 + (0.5 - e)^2 = 0.5 + 2e^2 \tag{2.18}$$

If running the XOR operation between four of these bits, we get:

$$P(0) = 0.5 + 8e^2 \tag{2.19}$$

It can be seen how, by increasing the number of bits, the polarisation factor decreases, e being less than 1. If the polarisation factor e is known, it is possible to calculate the number of bits that must be used to perform the XOR operation in order to reduce the polarisation itself to negligible values.

To reduce, in any case, any residual correlation of flow in output, it is also possible to use two or more random sources and to perform the operation of XOR between the output bits of each of the processes measured.

2.8.5 Distillation of randomness

The best way to have a perfectly random generator is by taking several random, or almost random, sources and distilling their size in output to concentrate their inherent randomness. This randomness can be registered to be used appropriately at a later date. A very useful function to perform this distillation is the hash function. This function can be applied to a myriad of events that take place on a computer, such as typing on the keyboard; the movement of the mouse; the sector number, time and latency of every disc operation; the current number of scan lines of the monitor; the content of the image being currently displayed; the contents of the FAT and the kernel tabs; the time of accessing devices; the CPU load; the microphone input and so on.

2.9 One-way hash functions

A one-way hash function $H(M)$ is a function that operates on a message M of arbitrary length, returning a message h of length predetermined m, called pre-image. Mathematically, this is expressed as:

$$h = H(M) \tag{2.20}$$

There are many functions able to do this but only hash functions have the following characteristics:

1. Given M, it is easy to calculate h.
2. Given h, it is difficult to calculate M in such a manner that $H(M) = h$.
3. Given M, it is difficult to find another message M^* such that $H(M) = H(M^*)$.

In practice, a hash function calculates a unique fingerprint of a given message M.

If Alice signs her message M using a DSA on $H(M)$ and Bob is able to generate a message M^*, which is different from M, such that $H(M) = H(M^*)$, then Bob could declare that Alice has signed M^*.

In some applications, the fact of being one way is not a sufficient characteristic and a further characteristic called collision resistance is required that is the difficulty of finding two random messages M and M^* such that $H(M) = H(M^*)$.

A typical attack against this ownership is the birthday attack that has already been described above.

The following protocol illustrates how, if the collision resistance property is not satisfied, Alice can use the birthday attack to deceive Bob. This protocol operates according to the following steps:

1. Alice prepares two versions of a contract, one favourable and the other unfavourable to Bob.
2. Alice performs a series of changes to every document and calculates its hash. These changes are mild in nature and, for example, act on the punctuation of the document itself.
3. Alice compares the hash values of each pair of documents in search of those that have the same hash value.
4. Alice has Bob sign the version of the contract that is favourable to the latter using a protocol that requires only the signature of the hash value.
5. In the future, Alice replaces Bob's original contract with the varied version, convincing an external judge, where required, that the second is the original contract.

There are many other protocols of attack, most of which are based on the birthday attack.

Of great importance also is the key length. If only 64 bits are used, there is a real possibility that a birthday attack can be successfully carried out. For this reason, the majority of hash functions produce 128-bit outputs, forcing the attacker to try 2^{64} random documents before finding the pair that provides the same hash value. The 128 value is considered not excessively high for lasting security and for this reason new concept algorithms use 160 bits.

A good method for generating a secure hash value is as follows:

1. The hash value of a message is generated using any secure algorithm.
2. The hash value found is added to the message.
3. The hash value of the whole "calculated-message hash" is generated.
4. A hash value consisting of the hash value calculated in step 1 concatenated with that calculated in step 3 is generated.
5. The operations referred to in steps 1 and 3 are repeated the desired number of times.

Since the hash functions operate on messages of arbitrary length always returning strings of predetermined length, they must operate a sort of compression on the incoming message. Usually these operate on blocks of data of the input message, calculating, from time to time, the hash value h_i of block M_i. In the subsequent steps, compression is performed, providing in input at this stage M_i and the value h_{i-1}.

Following this process, the hash of the entire message is represented by the hash of the last input block. The output value, the so-called pre-image, is a binary representation of the incoming message.

There are many hashing algorithms that will not be discussed in this section for reasons of space, referring to the bibliography section provided at the end of this book for further information. They are: Snefru, N-Hash, MD2, MD4, MD5, Secure Hash Algorithm (SHA), RIPE-MD and Haval.

2.9.1 Use of the symmetric block algorithms for generation of one-way hash functions

Symmetric block algorithms can be efficiently used for the generation of one-way hash functions: if the algorithm is secure, then the corresponding hash function is also secure.

The simplest method is to encrypt the message using a certain key, a given initialisation vector, and using the last block encrypted as hash value.

A more advanced approach is to use the block message as the key, the previous hash value as input and the current value of the hash as output.

A very useful parameter for assessing the performance of a hash function based on block ciphers is the so-called hash rate, or number of n bit block messages: the higher the rate, the more rapid the algorithm.

There are many hashing algorithms that based their operation on block algorithms. They are not discussed in the following description, for reasons of space, referring to the bibliography section provided at the end of this book for further information. They are: Davies–Meyer amended, Preneel–Bosselaers–Govaerts–Vanderwalle, Quisquater–Girault, Loki double-block, Parallel Davies–Meyer, Tandem and Abreast Davies–Meyer, MDC-2, MDC-4, AR and GOST.

2.9.2 Use of public-key algorithms for the generation of one-way hash functions

Symmetric block algorithms can be efficiently used for the generation of one-way hash functions when they are used in block chaining mode. If at the end of the operation the private key is destroyed, breach of the hash value is just as difficult as reading the message without possessing the related private key.

The following example is based on the RSA algorithm which is illustrated in detail in the following. Given a message M of which we want to calculate the hash value; given a number n obtained by the product of two primes p and q and given that a number is very large and relatively prime to $(p - 1)(q - 1)$, then the hash function is calculated as $H(M) = M^e \bmod n$. The solution to this problem is as difficult as finding the discrete logarithm of e. The only contraindication of this algorithm is its extreme slowness compared to other algorithms listed above.

2.9.3 Message authentication code

A message authentication code (MAC) is a one-way hash function that depends on a key. The MAC, in practice, enjoys the same properties of one-way hash functions, the only difference being that it uses a key allowing only those in possession of the latter to be able to verify the hash value. The MAC can be used to authenticate the files exchanged between different subjects. It can also be used to verify the integrity of an exchanged file that may have been altered, for example, by a virus. In this sense, a user can calculate the MAC of a file and store its value in a table: if we used a simple hash function, an intelligent virus could alter the file, calculate the hash and replace it with the one stored in the tables. This is not possible when using the MAC.

A very simple way to use a hash function such as MAC is to encrypt the hash value with a symmetric algorithm. Every MAC can be used as a one-way hash function making its key public.

There are many MAC algorithms that will not be discussed in this section for reasons of space, referring to the bibliography section provided at the end of this book for further information. They are: CBC-MAC, message authentication algorithm (MAA), bidirectional MAC method, Jueneman, RIPE-MAC, IBC-Hash, one-way hash function MAC and MAC stream cipher.

2.10 Advanced Encryption Standard

This section explains the algorithm called AES.

2.10.1 Introduction to AES

The Advanced Encryption Standard (AES) is a block encryption algorithm used as encryption standard by the US Government. Given its security and its specifications, it is expected that in the near future it will be used throughout the world as was the case with its predecessor, the DES.

For many years, in fact, the DES represent the standard for encryption and authentication of documents. The NIST chose it as an encryption standard in 1977 with 5-year validity. This standard was reaffirmed until December 1998.

It has already been said that from the outset the DES was the subject of much criticism that debated its cryptographic strength, such as:

1. in reference to the use of S-boxes in encryption and decryption procedures, it was suspected that these structures concealed trap doors;
2. half of the 256 keys necessary for a brutal attack permitted its violation. This led to the consideration that the key length was too short.

In view of the date of expiry of the DES validity and due to these criticisms, the NIST decided to choose new encryption standards that offered a higher level of security. To this end, on 12 September 1997, the NIST proclaimed a public competition for the appointment of a new and more advanced standard: the AES.

On 2 October 2000, the NIST selected “Rijndael” as the algorithm to be proposed for the AES. The algorithm was developed by two Belgian cryptographers: Joan Daemen and Vincent Rijmen, hence the name of “Rijndael”, derived in fact from the names of the inventors.

The algorithm was finally adopted by the NIST and by US FIPS PUB in November 2001 after years of studies and standardisation efforts.

Unlike DES, Rijndael is a replacement and permutation network, not a Feistel network. AES is fast whether developed in software or if developed in hardware and is relatively easy to implement and

requires little memory. The new encryption standard is replacing previous standards and its diffusion continues to increase.

2.10.2 Preliminary concepts

2.10.2.1 Input and output

The input and output of the algorithm consists of a sequence of 128 bits. We often refer to this sequence with the term "block" and the number of bits that compose it with the term "block length". An encryption key is a sequence of 128, 192 or 256 bits. The bits of these sequences are numbered starting from 0, therefore the value linked to each bit is an index that varies in the range $0 < i < 128$, $0 < i < 192$ or $0 < i < 256$, depending on the length of the block and key.

2.10.2.2 Bytes

The basic unit in the computation of Rijndael is the byte, a sequence of 8 bits treated as a single entity. The input, output and encryption key are processed as the byte arrays, obtained by dividing the sequences into groups of eight contiguous bits. The byte sequences are referred to using the array in which they are stored, for example $a[n]$, where the value n may vary in the range: length of the key = 128 bits, with $0 < n < 16$; length of key = 192 bits, with $0 < n < 24$; length of key = 256 bits, with $0 < n < 32$; and length of the block = 128 bits, with $0 < n < 16$.

It is useful to represent the values of the bytes using hexadecimal notation in which the groups of 8 bits are divided into two groups of four and each of the two represents a character between 0, 9, $a, \ldots, f$, as shown in Figure 2.22.

For example, the item {0 1 1 0 0 0 1 1} can be represented as {6 3}, where the four most significant bits form the value 6, while the remaining bits form a value 3.

2.10.2.3 Byte array

Byte arrays are shown in the following form: $a0$ $a1$ $a2$ . . . . . $a15$.

The order of the bits compared to byte is defined by the sequence of 128 bits of input0: input1, input2, ... , input126, input127 (input sequence 128 bits as follows:

$a0$ = {input0, input1, ... , input7};
$a1$ = {input8, input9, ... , input15}; (order of the bits between the bytes)
. . . . . .
. . . . . .
. . . . . .
$a15$ = {input120, input121, ... , input127}.

Figure 2.23 shows how the bits are numbered in each byte.

Bit Pattern	Character
0000	0
0001	1
0010	2
0011	3

Bit Pattern	Character
0100	4
0101	5
0110	6
0111	7

Bit Pattern	Character
1000	8
1001	9
1010	a
1011	b

Bit Pattern	Character
1100	c
1101	d
1110	e
1111	f

Figure 2.22 Hexadecimal representation.

Input bit sequence	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	.....
Byte number	0								1								2								.....
Bit numbers in byte	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	.....

Figure 2.23 Numbering of bytes.

2.10.2.4 State

Internally, the algorithm operations are performed on a two-dimensional array of bytes called States. The State consists of four rows of bytes, each containing Nb bytes, where Nb is the block length divided by 32. In the state array, denoted by the symbol s, each byte has two indices: the index row r varies in the range $0 < r < 4$ and the column index c varies in the range $0 < c < Nb$ (for the standard $Nb = 4$, so c varies in the range $0 < c < 4$).

At the beginning of the encryption or decryption, as we will see, the input, stored in the array[ ], is copied into the state array. All the operations are carried out on this array and thus its final value is copied into the output byte array, out[ ] Figures 2.24 to 2.30.

At the beginning of encryption or decryption, the input array in[ ] is copied into the state array in agreement with the following scheme:

$s[r, c] = \text{in}[r + 4c]$, with r between $0 < r < 4$ and c between $0 < c < Nb$.

At the end of encryption or decryption, the state array is copied into the output array out[ ] as follows:

$\text{out}[r + 4c] = s[r, c]$, with r between $0 < r < 4$ and c between $0 < c < Nb$.

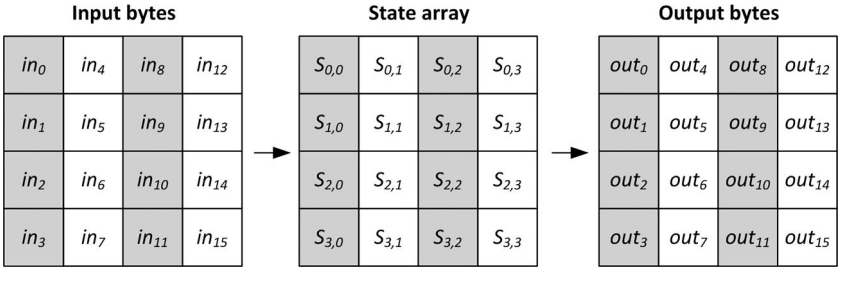


Figure 2.24 Input, status and output array.

c_0	c_4	a_8	a_{12}
c_1	c_5	a_9	a_{13}
c_2	c_6	a_{10}	a_{14}
c_3	c_7	a_{11}	a_{15}

Figure 2.25 State array.

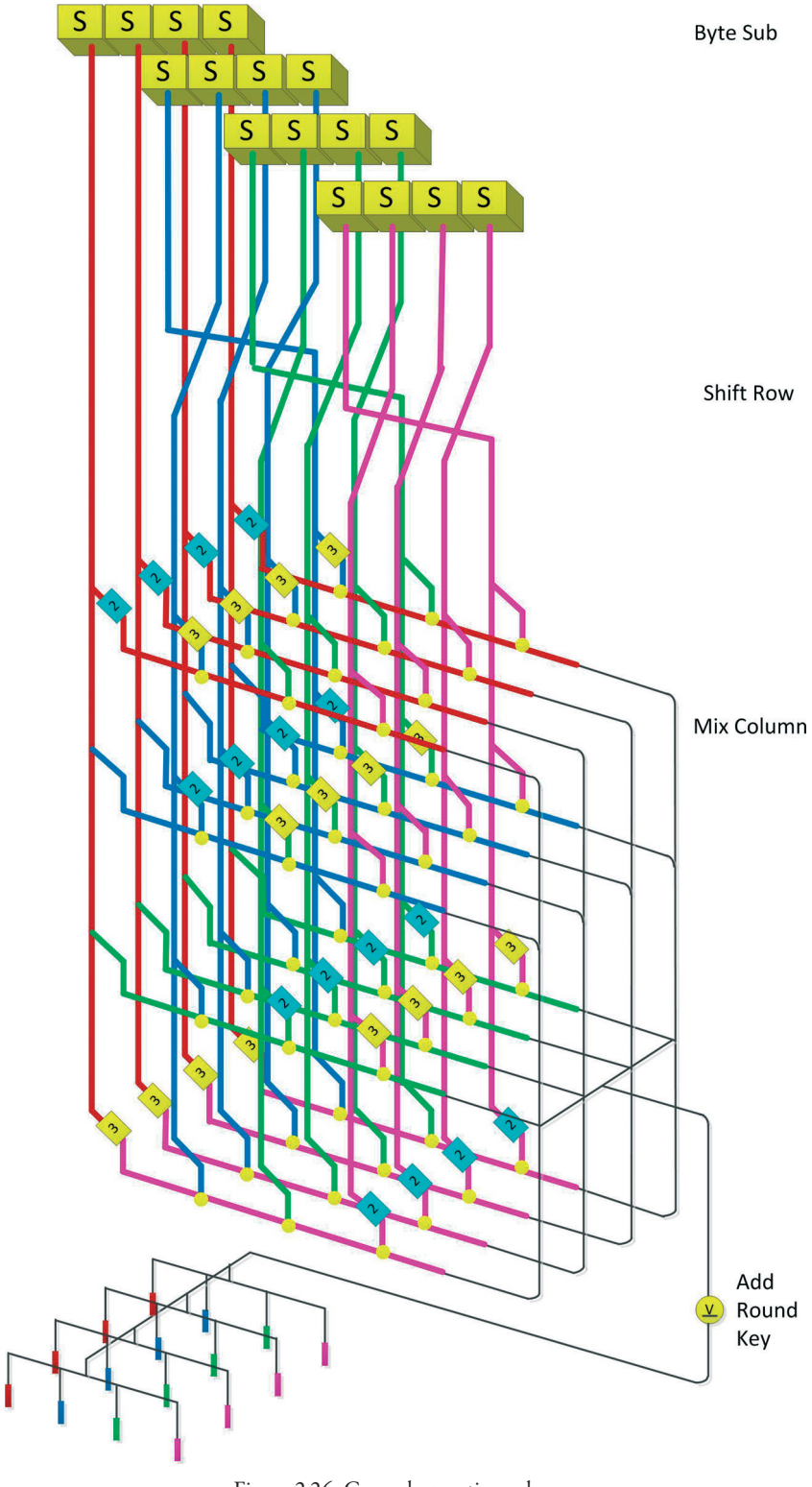


Figure 2.26 General operation schema.

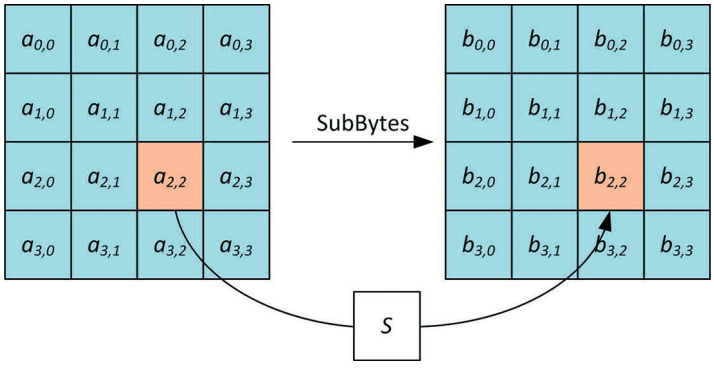


Figure 2.27 SubBytes transformation schema.

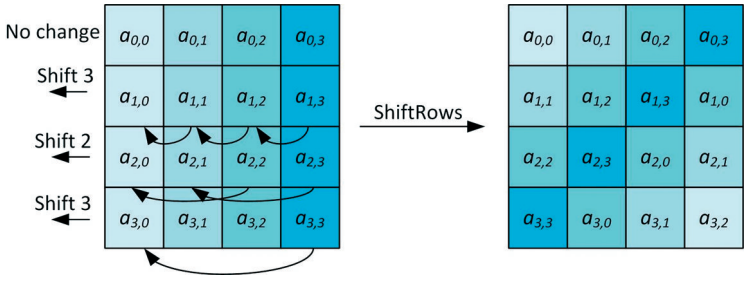


Figure 2.28 ShiftRows transformation schema.

It may be noted that 4 bytes in the columns of the state array form words of 32 bits, where the number of row *r* establishes an index for 4 bytes between each word.

2.10.2.5 State as column array

We have said that 4 bytes in each column of the state array form words of 32 bits, where the number of *r* rows establishes an index for 4 bytes between each word. The State can also be interpreted as a one-dimensional array of words of 32 bits, where the number of column *c* establishes an index in this array. For example, the State can be considered as an array of four words, as follows:

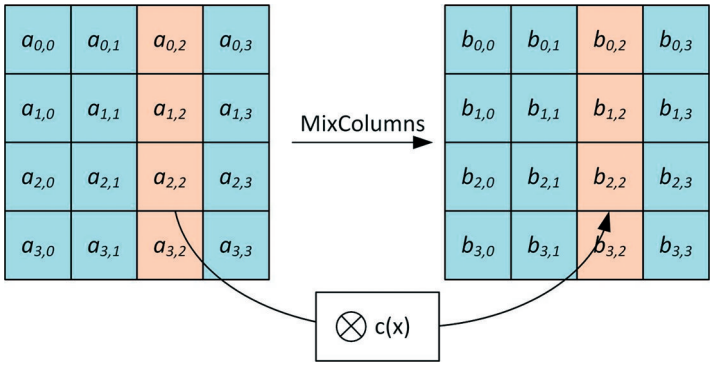


Figure 2.29 MixColumns transformation schema.

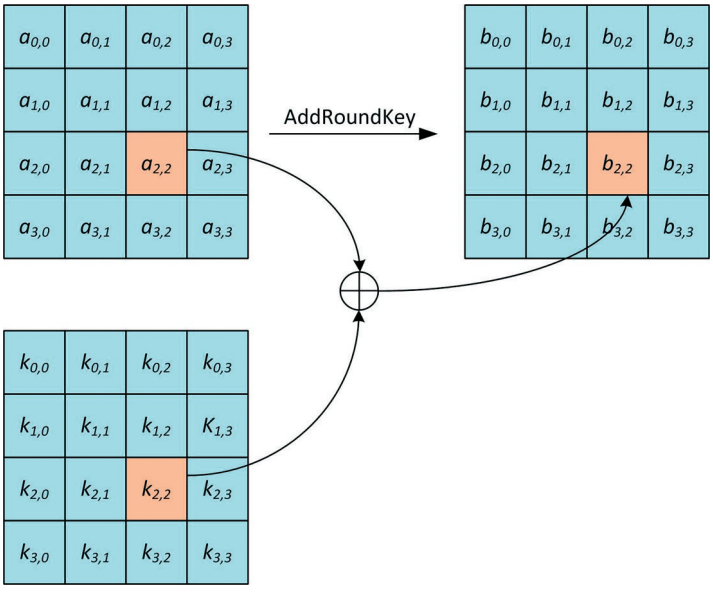


Figure 2.30 AddRoundKey transformation schema.

w0 = a0 a1 a2 a3 w1 = a4 a5 a6 a7 w2 = a8 a9 a10 a11 w3 = a12 a13 a14 a15

where we considered the input block consists of the following bytes a0 a1 a2 ... ... a15 and is therefore mapped in the state array.

2.10.2.6 Mathematical preliminaries

All bytes in the algorithm are interpreted as elements of a "finite field". The elements of a finite field can be represented in different ways, in this case polynomial representation was chosen.

A byte b, formed by the bits b7 b6 b5 b4 b3 b2 b1 b0, is considered as a polynomial with coefficients in {0, 1}: b7 $x7$ + b6 $x6$ + b5 $x5$ + b4 $x4$ + b3 $x3$ + b2 $x2$ + b1 x + b0.

For example, the byte with a binary value 01010111 corresponds to the polynomial $x6 + x4 + x2 + x + 1$.

The elements of a finite field can be added and multiplied, but these operations are different from those used for numbers.

2.10.3 Description of the algorithm

Rijndael is a block cipher with a variable length of the block and key.

AES provides a block length of 128 bits and a key length of 128 bits, thus defining Rijndael as the new AES-128 standard (128 indicates the length of the key). However, another two lengths of the key are also taken into account, 192 bits and 256 bits, thus obtaining versions AES-192 and AES-256.

Rijndael was designed to be used with additional lengths of the block and key, however these are not taken into consideration in the standard. For this reason, formally, AES is not equivalent to the Rijndael, since in AES the block is of a fixed size (128 bits) and the key can be 128, 192 or 256 bits, while the Rijndael specifies only that the block and the key must be a multiple of 32 bits with 128 bits as a minimum and 256 bits as maximum. The Rijndael can then handle different block and key sizes.

AES operates using arrays of 4 × 4 bytes called states. When the algorithm has blocks of 128 bits as input, the State matrix has four rows and four columns; if the number of blocks in input becomes

32 bits longer, a column is added to the State and so on up to 256 bits. In practice, the number of bits of the block in input can be divided by 32 and the quotient specifies the number of columns.

2.10.4 Rational schema

The Rijndael cipher uses simple operations by employing an encryption key. The various operations are applied to the bytes of the block in input in several rounds, the number of rounds is variable and depends on the length of the key and the block. Each round involves four basic operations for both encryption and decryption, with which substitutions, mixtures and movements of the byte of input are made by creating in this way a non-linearity of the data which means greater security.

The rational schema describes what happens during the encryption algorithm. In fact, the four fundamental operations that characterise each round are:

1. The first is a byte replacement transformation, SubBytes, which uses a substitution table (S-box).
2. The second is a row shift transformation with different offsets, ShiftRows.
3. The third is a data mix transformation between each column of the state array, MixColumns.
4. The fourth is a transformation that provides for the addition of the round key to the state array, AddRoundKey.

The transformations are described in the following:

1. SubBytes: this operation computes a replacement of bytes using a replacement table known as S-box.

   In the SubBytes step, each byte of the matrix is changed via the S-box to 8 bits. This operation provides non-linearity to the algorithm. The S-box used is derived from an inverse function in the finite field GF(28), known to have excellent properties of non-linearity. To avoid a potential attack based on algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box was carefully chosen such that it would not possess fixed points or opposite fixed points.
2. ShiftRows: this operation accomplishes a cyclic displacement of the rows of the State that contains the bytes of the input data.

   The ShiftRows step moves the rows of the matrix of a parameter depending on the number of row. In AES the first row remains unchanged; the second is moved to a place to the left; the third by two places and the fourth by three. In this way the last column of the data in input will form the diagonal of the matrix in output. (Rijndael uses a slightly different design due to length matrices that are not fixed.)

   All of the operations are carried out using the index of the modulo column.
3. MixColumns: this operation produces a mixture of bytes in the State columns.

   The MixColumns step takes the four bytes of each column and combines them using an invertible linear transformation. Used in conjunction, ShiftRows and MixColumns ensure that the criterion of confusion and diffusion in the algorithm (Shannon theory) are complied with. Each column is treated as a polynomial in GF(28) and modulo $x4 + 1$ is multiplied by a fixed polynomial $c(x) = 3 \times 3 + x2 + x + 2$.
4. AddRoundKey: this operation accomplishes the addition of a round key to the bytes of the data.

   The AddRoundKey step combines, via one XOR, the session key with the matrix obtained from previous steps (State). A session key is obtained from the primary key for each round (with the generally simple steps, e.g. a shift of the bit position) thanks to the key scheduler.

The operation of replacing the bytes imposes, through the application of S-boxes, a certain non-linearity in the data, while the subsequent operations of ShiftRows and MixColumns realise a mixture of data that reminds us of the famous Rubik's cube in which the colours of the cubes can be mixed with

movements on the columns and rows. In this way it has been created, this cipher realises the step for decryption of a block cipher by simply reversing some of the above-mentioned operations. These inverse operations are then applied in a number of rounds similar to that counted in encryption.

2.10.5 Encryption

At the beginning of encryption, the input, stored in the array in[ ], is copied into the state array. After the initial addition of a round key, the state array is transformed with a round transformation cycle of 10, 12 or 14 rounds. The number of rounds depends on the length of the key and also the last round differs from the first $Nr - 1$. The transformation cycle completed, the final State is copied into the output, out[ ] array. The encryption also uses the scheduled key that consists of a one-dimensional array of 4-byte words, denoted with w[ ], derived from the expansion function of the key.

2.10.6 Key expansion function

Rijndael uses an encryption key K and implements a routine of key expansion to generate a scheduled key. The expansion algorithm of the key generates a total of Nb^* $(Nr + 1)$ words: the encryption algorithm requires an initial set of Nb words, and each one of the Nr round requires Nb words of the key. The resulting scheduled key consists of a linear array of 4-byte words, denoted by $w[i]$, with which it varies between $0 < i < Nb$ $(Nr + 1)$. The expansion of the input key in the scheduled key can be described in pseudo-code, where SubWord () is a function that takes in input a word of 4 bytes and applies the S-box for each of the 4 bytes to produce an output word. The function RotWord () takes a word [a0, a1, a2, a3] as input, performs a cyclic permutation and returns a word [a0, a1, a2, a3]. An array of constant words is also used, Rcon [i], that contains the data values from [$xi - 1$, { 0 0 }, { 0 0 }, { 0 0 }], with -1 being represents the $(i - 1)$th power of x in the field GF(28) (x is equal to { 0 2 }). It should also be noted that the variable i starts with the value of 1 instead of 0.

It can be noted that the first Nk words of the expanded key are obtained directly from the encryption key, while every subsequent word, $w[i]$, is equal to the XOR of the previous word, $w[i - 1]$, with the earlier Nk word positions. For the word with positions which are multiples of Nk, a transformation before the XOR, followed by an XOR with a round constant, Rcon[i] is applied to $w[i - 1]$. This transformation consists of a cyclic shift of bytes to a word (RotWord ()), followed by a SubWord () function that applies an S-box to all 4 bytes of the word. It is important to note that the expansion algorithm of the key for a 256-bit key ($Nk = 8$) behaves differently with respect to a 128- or 192-bit key. If $Nk = 8$, and $i-4$ is a multiple of Nk, SubWord () transformation is applied to $w[i - 1]$ before XOR.

The function of the expansion algorithm of the key is to provide resistance against the following types of attack:

1. Attacks in which part of the encryption of the key is known to the cryptoanalyst.
2. Attacks in which encryption of the key is known and the encryption is used as a function of compression of a function.
3. Related-Key attacks.

A necessary condition to resist Related-Key attacks is to never have two ciphers of the different keys that have a common round key set.

Expansion of the key also plays an important role in the elimination of symmetries:

1. Symmetry in round transformation: this treats all the bytes of a State in the same way. This symmetry can be removed by having constant rounds in scheduling of the key.
2. Symmetry between rounds: round transformation is the same for all rounds. This equality can be removed by having dependent and constant rounds in scheduling of the key.

Key expansion was chosen according to the following criteria:

1. use of invertible transformations. In fact, knowing all *Nk* consecutive words of the expanded key allows regeneration of the full table;
2. speed on a wide range of microprocessors;
3. the use of constant rounds to remove symmetries;
4. the dissemination of ciphers of the different keys in the round key;
5. knowledge of a part of the encryption of the key or pieces of round key does not allow calculation of many other parts of round key;
6. sufficient non-linearity to prohibit the full determination of different round keys from ciphers of different keys;
7. simplicity of description.

2.10.7 Decryption

Transformations used in encryption can be reversed and then implemented in reverse order thus producing the decryption algorithm.

In realisation of the decryption, it is essential that the non-linear single step (SubBytes) is the first transformation in a round and the lines are moved before MixColumns is applied. In the inverse of a round, the order of transformations is reversed; as a result, the non-linear step will be the last step of the inverse round and the rows will be moved after application of the inverse of MixColumns.

The structure of Rijndael is such that the sequence of transformations in the decryption is the same as that of the encryption, with the transformations being replaced with their inverse and a change in the scheduling of the key.

The transformations used in decryption, InvShiftRows, InvSubBytes, InvMixColumns and AddRoundKey, process the state array that contains the ciphertext and produce plaintext as a result.

2.10.8 Security

During 2004 there were no AES forcing. The NSA pointed out that all of the finalists of the standardisation process were equipped with sufficient security to become the AES, but Rijndael was selected due to its flexibility in dealing with different key lengths, its simple implementation in hardware and in software and its low memory requirements that also allows its implementation in devices with limited resources such as smart cards. AES can be used to protect classified information. For SECRET level, a 128-bit key is sufficient, while for the TOP SECRET level, 192- or 256-bit keys are recommended. This means that for the first time the public has access to a cryptographic technology that NSA considers adequate to protect TOP SECRET documents. The need to use long keys (192 or 256 bits) for TOP SECRET documents has been explained. Some believe that this would indicate that the NSA has identified a potential attack that could force a relatively short key (128 bits), while most experts believe that the recommendations of the NSA are based primarily on wanting to ensure a high margin of security for the next decades against a potential exhaustive attack.

Most of the cryptographic algorithms are forced by reducing the number of rounds. AES performs 10 rounds for the 128-bit key, 12 rounds for the 192-bit key and 14 rounds for the 256-bit key. The best attacks have managed to force the AES with 7 rounds and 128-bit key, 8 rounds and 192 bit key and 9 rounds and 256-bit key.

Some cryptographers have pointed out that the difference between the rounds performed by AES and the maximum rounds before the algorithm is no longer forceable is small (especially with short keys). They fear that improvements in analysis techniques could allow them to force the algorithm without checking all the keys. Currently, an exhaustive search is impractical: the 128-bit key produces

3, 4, . . ., 1,038 different combinations. One of the best brute-force attacks was carried out by the project "distributed.net" on a 64-bit key using the RC5 algorithm; the attack took almost 5 years, using the "free" time of thousands of CPU of volunteers scattered throughout the network. Even considering that the power of the computer increases over time, significant time would still be required before a 128-bit key could be attacked with the brute-force method.

Another doubt concerning the AES stems from its mathematical structure. Unlike most of the block algorithms, for AES there is a detailed mathematical description. Although it has never been used to conduct a tailored attack, this does not exclude the fact that in future this description may be used to conduct an attack based on its mathematical properties.

In 2002, the theoretical attack called eXtended Sparse Linearization (XSL) attack announced by Nicolas Courtois and Josef Pieprzyk showed a potential weak point of the AES (and of other ciphers). Although the attack is mathematically correct, in reality it is impracticable due to the huge amount of machine time required to implement it. Improvements in the attack have reduced the machine time required and thus, in future, this attack may become feasible.

However, currently, the AES is considered a fast algorithm, secure and attacks, up to now presented, have provided interesting theoretical studies but in practice are of little use.

2.11 Public-key algorithms

Public-key algorithms were discovered by Diffie and Hellman and independently by Merkle. The basic concept of these algorithms is represented by the possibility of generating key pairs (public and private, one for encryption and another for decryption), inextricably linked to each other. The first algorithm of this kind was presented by Diffie and Hellman in 1976.

Since the date, there have been many other public-key algorithms developed, many of which are not considered secure, and among those considered secure many of them are not useful or practical. In addition, many of these algorithms are characterised by keys that are too long for practical uses or by ciphertext that are much larger than plaintext.

Only a small number of algorithms are characterised by a high level of security and convenience. These algorithms are based on the mathematical problems described in the previous section. Some of these algorithms are extremely useful for public-key cryptography, others are suitable only for the distribution of keys while some others are only suitable for digital signature. Only three algorithms are generally suitable for all applications. They are RSA, Elgman and Rabin. The sole mechanism of the RSA is discussed in the following, for reasons of space, referring to the bibliography section provided at the end of this book for other algorithms. These algorithms are, unfortunately, extremely slow when compared with symmetric-key algorithms, as they based their operation on complex mathematical calculations. For this reason, they are used combined with symmetric algorithms, creating the so-called hybrid cryptosystems (already shown previously) where a symmetric algorithm with random session key is used to encrypt a message and a public-key algorithm is used to encrypt the random session key.

Public-key algorithms are designed to withstand chosen-plaintext type attacks because their security is based both on the difficulty that exists in deducting the private key from the public key and the difficulty that exists in deducting plaintext from ciphertext. These algorithms are not susceptible to chosen-ciphertext type attacks. Such an attack is difficult to prevent in systems where digital signature is performed in the reverse manner with respect to encryption, unless different keys for encryption and digital signature are used.

2.11.1 The RSA algorithm

The RSA algorithm is the easiest public-key system to understand and implement. It takes its name from its three inventors: Rivest, Shamir and Adleman, and has been tested for years by the most disparate cryptographic attacks.

RSA bases its security on the mathematical problem, as illustrated previously, of the factorisation of prime numbers and its keys are a function of a very large pair (at least 100 or 200 bits) of prime numbers. Deduction of plaintext starting from the relevant ciphertext and from the relevant public key used for encryption is equivalent to finding the factorisation of two prime numbers.

To generate the two keys, two very large prime numbers p and q are randomly chosen. For greater security, p and q are selected of the same length. After this, the product $n = pq$ is calculated. Subsequently, the encryption key is randomly chosen in such a way that e and $(p - 1)(q - 1)$ are relatively prime. Finally, using the extended Euclid algorithm, the decryption key d is calculated in such a way that $ed \equiv 1 \bmod (p - 1)(q - 1)$ which means $d \equiv e^{-1} \bmod (p - 1)(q - 1)$.

It is important to note that d and n are also relatively prime between each other. The e and n numbers represent the public key while the number d represents the private key. Once the key pair has been generated, the numbers p and q are no longer use and can be erased but never disclosed, failure to do so leads to the violation of the private key.

To encrypt a message m, it is divided into numeric blocks smaller than n: if binary data are used then the largest power of 2 less than n is chosen. If p and q are primes of at least 100 bits, then n will be a number of 200 bits, and each message block m_i will be less than 200 bits. The encrypted message c will be composed of message blocks c_i that are generally the same size. The encryption formula is $c_i = m_i^e \bmod n$. To decipher the message, a cipher block c_i is taken and the operation $m_i = c_i^d \bmod n$ is followed since $c_i^d = (m_i^e)^d = m_i^{ed} = m_i^{k(p-1)(q-1)+1} = m_i m_i^{k(p-1)(q-1)} = m_i{*}1 = m_i{*}1$ all modulo n.

The RSA operating mode is summarised in Table 2.19.

Let us examine, in the following, a practical example. Assuming $p = 47$ and $q = 71$ that give $n = pq = 3{,}337$. The encryption key e must not have factors in common with $(p - 1)(q - 1) = 46{*}70 = 3{,}220$. A random choice of e could be represented by number 79. In this case, using Euler's extended algorithm, $d = 79^{-1} \bmod 3{,}220 = 1{,}019$. At this point, we can publish e and n, keeping secret d and eliminating p and q. To encrypt a message of any length, it is divided into suitable blocks as indicated above and the various blocks are encrypted in sequence by performing the appropriate calculations. Decryption is performed in a similar way using the decryption key and working block by block in the same encryption sequence.

In hardware implementations, the RSA algorithm is about 1,000 times slower than DES, while in software implementations it is about 100 times slower.

RSA security is based entirely on the problem of the factorisation of large prime numbers. To attack RSA efficiently, a search algorithm of prime numbers should be put in place, not yet discovered. It is also possible to attack RSA by trying to randomly guess the value $(p - 1)(q - 1)$ but this type of attack is simpler than that relating to factorisation.

Table 2.19 RSA operating mode.

Public key	n: product of two prime numbers p and q, which must remain secret e: relatively prime with p and q
Private key	d: $e^{-1} \bmod (p - 1)(q - 1)$
Ciphering	$c = m^e \bmod n$
Deciphering	$m = c^d \bmod n$

Factorisation of n represents the most obvious type of attack. In this case, the attacker has the public key and modulo n: to find d he/she must factor n, addressing a mathematical problem which is not easy to solve. Of course, it is possible to try all the possible values of d using a brute-force attack but this type of attack is less efficient than factorisation.

Some types of attacks are concentrated on the implementation of the RSA rather than against the basic algorithm. Let us examine a few examples.

In the first type of attack, Eve listens to communications of Alice and intercepts a message c encrypted with her public key using RSA. Eve wants to be able to read this message, that is she wants to perform the following mathematical operation: $m = c^d$. To retrieve the message m, she chooses a random key r such that r is less than m. Subsequently, she retrieves the public key and calculates $x = r^e \bmod n$, $y = xc \bmod n$ and $t = r^{-1} \bmod n$. If $x = r^e \bmod n$ then $r = x^d \bmod n$. At this point, Eve tries to make Alice sign y with her private key, deciphering y. Alice then sends Eve $u = y^d \bmod n$. At this point, Eve calculates $u \bmod n = r^{-1} y^d \bmod n = r^{-1} x^d c^d \bmod n = c^d \bmod n = m$, obtaining, as final result, the desired message m.

In the second type of attack, Trent is the computer of a public notary. If Alice wants a document to be recorded, she sends it to Trent who signs it with the digital signature RSA and sends it back. Mallory at this point wants Trent to sign a message m^* that the same would never sign. First, Mallory chooses an arbitrary value x and computes $y = x^e \bmod n$ being able to find easily e, the latter being the public key of Trent. After this, Mallory calculates $m = ym^* \bmod n$ and sends m to Trent for him to sign that returns $m^{*d} \bmod n$. At this point, Mallory calculates $(m^d \bmod n)\, x^{-1} \bmod n$, which represents the signature of m^* that is the predetermined objective of Mallory.

In the third type of attack, Eve wants Alice to sign the message m_3. She generates, in this sense, two messages m_1 and m_2 in such a way that $m_3 = m_1\, m_2 \bmod n$. If Eve succeeds in having Alice sign messages m_1 and m_2, then she can calculate m_3 as $m_3^d = (m_1^d \bmod n)\, (m_2^d \bmod n)$. This leads to the recommendation of never signing documents using RSA when unsure of the origin.

Possible implementation of RSA could provide anyone with the same value of n but different values for the exponents e and d: this can be very risky. The most common problem is that if we encrypt the same message with different exponents and those exponents are relatively prime between each other, then the plaintext can also be recovered without the decryption exponent. If m is plaintext, e_1 and e_2 are the two encryption keys and n is the common form, then the two encrypted messages c_1 and c_2 will be, respectively, $c_1 = m^{e1} \bmod n$ and $c_2 = m^{e2} \bmod n$. The attacker knows n, e_1, e_2, c_1 and c_2. Since e_1 and e_2 are relatively prime between each other, using the Euler extended algorithm, it is possible to find r and s such that $re_1 + se_2 = 1$. Assuming that r is negative, then the Euler extended algorithm can be used again finding $(c_1^{-1})^{-r} c_2^s = m \bmod n$ which allows calculation of m. This leads to discouraging the use of a common value of n within a group of users.

Encryption and verification of the signature with RSA are very fast if a small value for e is used, but this makes the algorithm susceptible to attack.

Another type of attack is the possibility of recovering d when the same is less than a quarter of the size of n and e is less than n. This rarely happens if e and d are randomly chosen and does not happen at all if it is characterised by a small value of e.

From what we have seen, a series of points of weakness of the RSA can be listed:

1. Knowledge of a pair of encryption/decryption exponents for a given modulo makes it possible for a potential attacker to factor the modulo.
2. The knowledge of a pair of encryption/decryption exponents for a given modulo makes it possible for a potential attacker to calculate other pairs of encryption/decryption without necessarily having to factor n.
3. A common modulo in an RSA protocol on a communication network should not be used.
4. The messages should be mixed with random values to prevent attacks on reduced value exponents.
5. The decryption coefficient should be as large as possible.

We must always remember that it is not enough to have a secure encryption algorithm but both the cryptosystem and the cryptographic protocol must be secure.

In general, it is very important to sign messages before encrypting them even if, very often, this recommendation is not followed. This weakness may be easily attacked if using RSA. Let us suppose Alice wants to send a message to Bob. First, she encrypts it with Bob's public key and then encrypts it with her private key and the result is $(m^{eB} \bmod n_B)^{dA} \bmod n_a$. Bob may in this case declare that Alice sent him message m^* and not message m. In fact, since Bob knows the factorisation of n_B, as it is his modulo, he can calculate the discrete logarithm in relation to n_B and all that has to be done is to calculate $m^{*x} = m \bmod n_B$. Moreover, if he can publish xe_B as his new public exponent and keep n_B as his modulo, he can say that Alice sent him the message m^* encrypted with this new exponent.

RSA has, with time, become a standard throughout the world.

2.11.2 Elliptic curve cryptosystems

Elliptic curves have been studied for years, and as such their properties are very well known. In 1985, the first encryption schema based on elliptic curves was published. This schema, rather than being a new schema, was the adaptation of a public-key algorithm, such as Diffie–Hellman, using elliptic curves.

Elliptic curves are very interesting because they allow the construction of elements and rules of combination to produce appropriate groups that have very useful features for cryptographic applications but that, at the same time, make cryptoanalysis very difficult.

In particular, elliptic curves on the finite field GF(2^n) are extremely interesting.

For further in-depth studies on the subject, the reader can refer to the bibliography section provided at the end of this book.

2.11.3 Other public-key cryptosystems

In addition to RSA, very well known and used throughout the world, there are many other public-key algorithms that will not be shown in this book for reasons of space, referring to the bibliography section provided at the end of this book for further information. These are Knapsack, Pohlig – Hellman, Rabin, Elgamal, McEliece and Luc.

2.12 Public-key algorithms for digital signature

In this section, public-key algorithms for digital signature are illustrated.

2.12.1 Digital signature algorithm

In August 1991, the NIST published a proposal for a DSA for use in their DSS.

Before this proposal, there was great confusion between algorithm (DSA) and standard (DSS): the standard uses the algorithm and the algorithm is a part of the standard.

In 1992, NIST received a series of comments against DSA which are as follows:

1. DSA cannot be used for encryption and the distribution of keys.
2. DSA was developed by the NSA and thus there can be service ports within the algorithm.
3. DSA is slower than RSA.
4. RSA is a *de facto* standard.
5. The process of selecting DSA was not published and insufficient time for algorithm analysis has been given.

6. DSA may conflict with other patents.
7. The key size is too short.

In 1994, the standard was published.

DSA is a variant of the signature algorithm of Schnorr and Elgamal. It uses the following parameters:

1. p which is a long L bit prime number that varies between 512 and 1,024 and is a multiple of 64. Its limited length was the subject of much criticism and was subsequently changed by NIST.
2. Q which is a prime factor of $p - 1$ of 160 bits.
3. $g = h^{(p-1)/q} \bmod p$ where h is a number less than $p - 1$ in a manner such that $h^{(p-1)/q} \bmod p$ is greater than 1;
4. X, which is a number less than q;
5. $y = g^x \bmod p$.

The algorithm also uses a one-way $H(m)$ hash function where m is the message in question.

The first three parameters p, q and g are public and available on the network, the private key is represented by x and the public key is represented by y.

To sign a message m, the following tasks must be performed:

1. Alice generates a random number k lower than q.
2. Alice generates $r = (g^k \bmod p) \bmod q$ and $s = (k^{-1}(H(m) + xr) \bmod q$. The parameters r and s represent the signature that is sent to Bob.
3. Bob verifies the signature by calculating $w = s^{-1} \bmod q$, $u_1 = (H(m)w) \bmod q$, $u_2 = (rw) \bmod q$, $v = ((g^{u1} y^{u2}) \bmod p) \bmod q$. If $v = r$ then the signature is verified.

Table 2.20 shows the operative diagram of the DSA.

We have already said that 512 bits of digital signature may not be safe for long-term security while 1,024 bits certainly are.

Now let us look at the possible attacks against k. Each signature requires a new value of k and this value should be chosen at random. If Eve manages to find a k-value that Alice has used to sign a message, she can deduct some of the properties of the random number generator that has generated k and may retrieve Alice's private key. If Eve manages to get Alice to sign two messages using the same k, she can retrieve the private key x and with this private key can sign in place of Alice. For this reason, it is necessary that, in order to ensure maximum security, a good random number generator is used in DSA.

Table 2.20 DSA operating schema.

Public key	p prime number variable between 512 and 1,024 bits (can be shared in a group of users) q prime number of 160 bits factor of $p - 1$ (can be shared in a group of users) $g = h^{(p-1)/q} \bmod p$ where h is a number lesser than $p - 1$ so that $h^{(p-1)/q} \bmod p$ is greater than 1 (can be shared in a group of users) $y = g^x \bmod p$ (number of p bits)
Private key	$x < q$ (number of 160 bits)
Sign	k is chosen at random and it is lesser than q $r = (g^k \bmod p) \bmod q$ $s = (k^{-1}(H(m) + xr) \bmod q$
Sign verification	$w = s^{-1} \bmod q$ $u_1 = (H(m)w) \bmod q$ $u_2 = (rw) \bmod q$ $v = ((g^{u1} y^{u2}) \bmod p) \bmod q$ If $v = r$ the sign is verified

Table 2.21 Example of choice of the coefficients a, b, c ($r^* = r \bmod q$).

$\pm r^*$	$\pm s$	M
$\pm r^* m$	$\pm s$	1
$\pm r^* m$	$\pm ms$	1
$\pm mr^*$	$\pm r^* s$	1
$\pm ms$	$\pm r^* s$	1

Even if the DSS does not specify anything about the possible sharing of a common modulo between different users, some implementations do this. The use of a common modulo may represent a vulnerability, which is the subject of attack by cryptoanalysts.

2.12.2 Digital signature via discrete logarithms

Discrete logarithms can be used to provide an algorithm for digital signature that is shown below.

A very large prime number p and q, very large prime number and a factor of $p - 1$, are chosen. After this g is chosen, a variable prime number ranging between 1 and p in such a manner that $g^q = 1 \bmod p$. These numbers are public and can be shared by a group of users. The private key is x, number less than q, while the public key is $y = g^x \bmod p$.

To sign a message m, a random number k less than and relatively prime with q must be chosen. If q is also prime, then each value of k less than q is sufficient. Then $r = g^k \bmod p$ is calculated. The generalised signature equation becomes $ak = b + cx \bmod p$ where the coefficients a, b and c can be chosen from a variety of methods. Table 2.21 shows six possibilities.

To verify the signature, the recipient should check the following equation, called verification: $r^a = g^b y^c \bmod p$. Table 2.22 shows the list of signatures and relevant checks possible using only the first line of potential values for a, b and c and ignoring the sign $\pm$.

Table 2.22 shows only six possible schemas that climb up to 24 taking into account the sign $\pm$. If all the possible alternatives listed in Table 2.21 are used, a total of 120 signature schemas will be obtained.

2.12.3 Other algorithms for digital signature

In addition to DSA, there are other algorithms for digital signature that will not be shown in the following for reasons of space, for further information the reader can refer to the bibliography section provided at the end of this book. These are GOST, Ong–Schnorr–Shamir and ESIGN.

Table 2.22 Possible signature schemas for the algorithm based on discrete logarithms.

Sign equation	Verification equation
$r^* k = s + mx \bmod q$	$r^{r^*} = g^s y^m \bmod p$
$r^* k = s + sx \bmod q$	$r^{r^*} = g^m y^s \bmod p$
$sk = r^* + mx \bmod q$	$r^s = g^{r^*} y^m \bmod p$
$sk = m + r^* x \bmod q$	$r^s = g^m y^{r^*} \bmod p$
$mk = s + r^* x \bmod q$	$r^m = g^s y^{r^*} \bmod p$
$mk = r^* + sx \bmod q$	$r^m = g^{r^*} y^s \bmod p$

2.13 Algorithms for the exchange of keys

The algorithms for the exchange of the keys are very important to ensure the safety of a cryptosystem. In the following, only a few are described, referring, for more detail, to the bibliography section provided at the end of this book.

2.13.1 Diffie–Hellman

The first public-key algorithm discovered is presented by Diffie–Hellman, dating back to 1976. It bases its security on the ease with which the operation of exponentiation in a finite field can be performed and the relative difficulty with which the operation of discrete logarithm in the same finite field can be executed.

This algorithm can be used for the distribution of the keys, since Alice and Bob use it to exchange secret keys, but may not be used for the encryption and decryption of messages.

The mathematics underlying this algorithm are relatively simple. Initially, Alice and Bob reach an agreement on the use of two very large prime numbers n and g and in such a way that g is primitive modulo n. These integers must not be secret and Alice and Bob can also communicate them to each other on a channel that is not secure. These may also be shared by a group of users.

The protocol operates according to the following steps:

1. Alice chooses a very large random integer x and sends Bob the following number: $X = g^x \bmod n$.
2. Bob chooses a very large random integer y and sends Alice the following number: $Y = g^y \bmod n$.
3. Alice computes $k = Y^x \bmod n$.
4. Bob computes $k^* = X^y \bmod n$.

Using this procedure, two values k and k^* both equal to $g^{xy} \bmod n$ are obtained. No other person who is able to intercept communications on the channel is able to calculate this value. He/she is only able to know n, g, X and Y. The only way to obtain that value is able to calculate the discrete logarithm and retrieve x and y. In this way, k represents the secret key that both Alice and Bob have calculated separately.

It is clear that the correct choice of g and n is essential to ensure maximum security to the algorithm. In this sense, it can be demonstrated that the number $(n - 1)/2$ should be prime and n should be very large because the difficulty of the algorithm is based on the difficulty of factoring numbers of the same magnitude of n. With regard to g, it must be primitive modulo n and can also be chosen as arbitrarily small and not necessarily prime in that it only serves to generate a large subgroup of a multiplicative group modulo n.

The Diffie–Hellman algorithm can be easily extended to work with three or more users. Below is shown an operating procedure between three users: Alice, Bob and Carol. It operates according to the following steps:

1. Alice chooses a very large integer x and sends it to Bob $X = g^x \bmod n$.
2. Bob chooses a very large integer y and sends it to Carol $Y = g^y \bmod n$.
3. Carol chooses a very large integer z and sends it to Alice $Z = g^z \bmod n$.
4. Alice sends Bob $Z^* = Z^x \bmod n$.
5. Bob sends Carol $X^* = X^y \bmod n$.
6. Carol sends Alice $Y^* = Y^z \bmod n$.
7. Alice computes $k = Y^{*x} \bmod n$.
8. Bob computes $k = Z^{*y} \bmod n$.
9. Carol calculates $k = X^{*z} \bmod n$.

In this way, all the three subjects have calculated independently the key $k = g^{xyz} \bmod n$ without any eavesdropper on the channel being able to take possession of it. The protocol shown can be extended to any number of users by adding appropriate calculation steps.

A variant of Diffie–Hellman allows Alice to generate a key and to send it to Bob by performing the following steps:

1. Alice chooses a very large random integer x and calculates $k = g^x \bmod n$.
2. Bob chooses a very large random integer y and sends Alice $Y = g^y \bmod n$.
3. Alice sends Bob $X = Y^x \bmod n$.
4. Bob calculates $z = y^{-1}$ and $k^* = X^z \bmod n$.

Following these steps, we get $k^* = k$. The advantage of this protocol is the possibility of calculating k prior to any exchange of information between Alice and Bob, allowing Alice to encrypt a message using k. She can, in this way, send the message to different people and exchange the key at a later time.

If there are several users who want to communicate in a secure manner, each of them may publish their own public key $X = g^x \bmod n$ in a public database. If Alice wants to communicate with Bob, she needs only to take the public key of the latter and generate a key shared between the two that is used to encrypt the message. To decipher the message, Bob takes Alice's public key from the database and generates the same shared key. In this way, each user can generate a secure key of communication without having to exchange information first. It is clear that, in order to prevent possible attacks, public keys should be certified and changed with a certain periodicity.

2.13.2 Station–station protocol

The Diffie–Hellman protocol for key exchange is susceptible to man-in-the-middle attacks. To avoid this, Alice and Bob could sign their message in advance. The protocol that is shown in this section assumes that Alice possesses a public-key certificate of Bob and vice versa. These certificates must be issued by a third trusted party. The procedure is developed according to the following steps:

1. Alice generates a random number x and sends it to Bob.
2. Bob generates a random number y and, using the Diffie–Hellman key exchange protocol, calculates the shared key k based on x and y. After this, he signs x and y and encrypts the signature using k. Then he sends the encrypted message together with y to Alice.
3. Alice computes k and decrypts Bob's message, suitably verifying the signature. Then she sends Bob a signed message, containing x and y, encrypted using the shared key.
4. Bob decrypts the message and verifies Alice's signature.

2.13.3 Exchange of encrypted keys

Encrypted key exchange (EKE) is able to ensure security and authentication on a computer network using both symmetric algorithms and public-key algorithms in such a manner as to use a shared secret key to encrypt a public key generated at random. The protocol assumes that Alice and Bob share a common password P. This protocol allows the mutual authentication and generation of a shared session key K. The protocol operates according to the following steps:

1. Alice generates a random pair of private–public keys. She encrypts the public key K^*, using the symmetric algorithm and P as the key, and sends the message to Bob.
2. Bob, who knows P, deciphers the message received from Alice and obtains K^*. He subsequently generates a random session key K and the digit with the public key received from Alice, using P as the key, sending everything to Alice.

3. Alice decrypts the message to obtain K. She generates a random string R_A that she encrypts with the key K and sends to Bob.
4. Bob decrypts the message to obtain R_A. He generates another random string R_B, encrypts both strings with K and sends the result to Alice.
5. Alice decrypts the message to obtain R_A and R_B, encrypts R_B with K and sends it to Bob.
6. Bob decrypts the message to obtain R_B.

At this point, the protocol is terminated as both parties are able to communicate using the session key K.

In step 3, both Alice and Bob know K^* and K, where K is the session key that can be used to encrypt all the messages between them. Eve, in the middle, knows only $E_P(K^*)$ and $E_P(E_{K^*}(K))$ and certain messages encrypted with K. She could try with certain random values of P to decipher the messages but would not succeed because she would not be able to carry forward her attack without first violating the public-key algorithm: if K and K are chosen in a random manner, this attack becomes very difficult.

In the protocol used, steps 3 and 6 provide validation; steps 3 and 5 ensure that Alice and Bob know K and steps 4 and 6 reassure Bob that Alice knows K.

The EKE protocol can be implemented using different public-key algorithms such as Diffie–Hellman, RSA and ElGamal.

2.14 Quantum cryptography

Quantum cryptography uses the normal uncertainty present in the real world. Thanks to this, a channel of communication can be created in which it is impossible to conduct an operation of interception without disturbing transmission itself. Its security is based on certain physical laws of quantum mechanics which state that particles do not exist in a single state, but in different states at the same time all of which are characterised by a well-defined probability. If an outside observer tries to observe a particle, it collapses in only one of these aspects. It is therefore possible to measure an aspect of the particle but not all aspects at the same time. For example, it is not possible to measure the position and velocity of a particle at the same time. If measuring a quantity, the ability to measure the other quantity is inevitably lost. Such uncertainty is inherent to the quantum world and, therefore, to the world of particles and there is no way to sidestep this.

This uncertainty can be efficiently used to generate a secret key via the quantum properties of light photons. Photons, during their propagation, can oscillate along preference direction (high-low, right-left, up-and-down, etc.), and in this case photons are referred to as polarised along the considered direction. Sunlight is not normally polarised and emitted photons oscillate along all the directions. To polarise a non-polarised beam of photons, a polarising filter can be used that allows passage of all the photons characterised with the same filter of the and blocks all others. If a filter is horizontally polarised and the photons are vertically polarised, no photon will be able to pass. If this filter is rotated, it is seen that as the angle of inclination is gradually increased, the photons will begin to move, until all have passed when the angle of inclination becomes entirely parallel in the direction of oscillation of the photons. Such behaviour is contrary to intuition, in which it is expected that photons will pass only when the polariser has made the final rotation of 90°. But in quantum mechanics photons always have a certain probability of varying their polarisation: for very large angles of rotation, this possibility is practically zero, and the photons do not pass; for angles of inclination of 45°, this probability is equal to 1/2; for angles close to 90°, this probability increases until becoming unitary when the angle is 90°.

Polarisation can be measured using any basis. It may, for example, be vertical or horizontal linear, or right or left diagonal. If a polarised photon is measured with a determined basis, using the same basis, measurement will be effective and will measure the correct value; if the wrong basis is used, measurement will be random.

This property can be used to generate a secret key in the following manner:

1. Alice sends Bob a string of photonic impulses, each randomly polarised in one of the main directions of polarisation: vertical, horizontal, right diagonal and left diagonal.
2. Bob has a polarisation detector that can adjust to perform linear polarisation or diagonal polarisation measurements, it is impossible to perform both types of measurement at the same time as this is prohibited by quantum mechanics. If his detector is positioned to detect linear polarisations and the photon in arrival is linearly polarised, then it will be correctly detected. The same is true for diagonal polariser and diagonally polarised photon. If the detector is positioned to detect diagonal polarisation and the photon in input is diagonally polarised then the result of the measurement will be random. The same is true for diagonal polariser and linearly polarised photon. He is not, however, able to know the difference.
3. Bob tells Alice, on a channel that is not secure, what types of polarisation he has used for the various detections.
4. Alice tells Bob which bases used are correct.
5. Alice and Bob retain only the data related to polarisations correctly detected. Using an agreed code, they can translate these measured polarisations into bits. It could, for example, be considered that linear polarisations are equal to 0 and those diagonal equal to 1.

Following the exchange protocol illustrated, it is possible to communicate a certain number of bits according to the desired quantity. On average, Bob is able to correctly receive 50% of the bits sent by Alice: if the latter sends $2n$ bits, only n bits will on average be correctly received. They can use these bits for a symmetric key or even for a single use tab.

The great advantage is the impossibility, by Alice, of intercepting the transmitted bits. In fact, she must make the same random choice as Bob on detector polarisation to receive the various bits, always having a 50% chance of finding the correct detection polarisation. Since incorrect detections change the photon polarisation, she would inevitably introduce errors onto the channel during her attempt to intercept, ultimately causing Alice and Bob to possess different bit strings.

At the end of the exchange operation, Alice and Bob terminate in the following manner:

1. Alice and Bob compare certain bits of the string exchanged: if they find discrepancies, this means that they were intercepted while if there are none, then they eliminate the bits used for the comparison and use the others.

Using this system, the probability of interception of Eve is 50%. If n different levels of polarisation are used, then its probabilities of correct interception are only 1 to 2^n. If Eve tried to reconstruct all the bits, she would inevitably destroy them.

Several commercial systems for quantum cryptography are currently being developed that operate on optical fibres at distances of up to a few tens of kilometres.

2.15 Practical applications

A number of applications will be summarised later that are currently being used in cryptography. For reasons of space only a few applications will be explored, referring to the bibliography section provided at the end of this book for the additional numerous applications.

2.15.1 Management protocol of secret IBM keys

This protocol was developed at the end of the 1970s and provides a complete system of key management for communications over the network or for file encryption.

The protocol envisages three situations: secure communication between a server and different terminals, secure storage of data on a server and secure communication between servers. It is not able to ensure secure and direct communication between terminals but may be suitably modified to do this.

The details of operation of the protocol in question can be found in the bibliography section provided at the end of this book.

2.15.2 STU-III

Cryptography has also been used for the protection of telephone conversations using dedicated devices such as the Secure Telephone Unit (STU). It has the same size as a normal telephone but is resistant to interceptions.

2.15.3 Kerberos

Kerberos is an authentication protocol on the TCP/IP network that uses a trusted third party, acting as an arbitrator. It provides authentication on a secure network, allowing a given subject access to various computers on the network. It bases its operation on symmetric key cryptography, sharing a symmetric key with every subject that operates on the network and uses the knowledge of the key as an identifier.

To operate properly, Kerberos is equipped with an archive containing all the keys of the computer connected to the network, whether they are client or server. All network services, as well as the clients that use these services, are authenticated. Since Kerberos knows everyone's secret keys, it can generate messages to ensure the identity of a given subject in relation to another subject. Kerberos is also able to generate session keys that are used by two subjects for the secure exchange of information. These session keys, once used, are destroyed. The main algorithm for symmetric encryption used by Kerberos is DES.

Kerberos operates through tickets that are released, after checking the identity of the applicant in order to request a service. In this sense, it uses ticket granting. When a client request a service, it sends a request for issue of a ticket to Kerberos that, after having authenticated the client, proceeds to authorisation. Subsequently, the client submits such authorisation to the ticket handler that, if everything is in place, will issue the ticket required. At this point, the client sends this ticket to the desired server to use the required service. A diagram of the operation is shown in Figure 2.31.

Kerberos will be discussed in detail in the chapter 5 concerning the security of communications on a wired network.

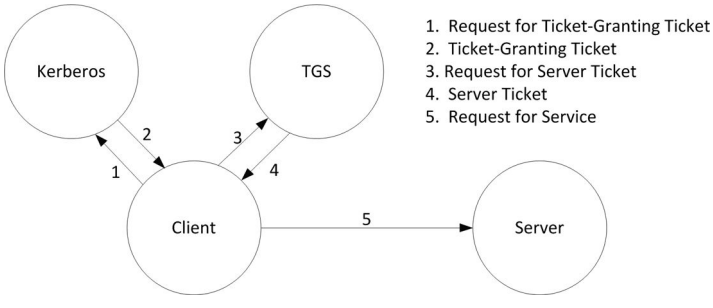


Figure 2.31 Diagram of Kerberos operation.

2.15.4 Kryptonight

Kryptonight represents a system of authentication and key distribution developed by IBM. It is able to provide the following services:

1. authentication of users;
2. authentication between two parties;
3. key distribution;
4. authentication of the origin and content of the data.

It is very similar to Kerberos but is characterised by certain differences as follows:

1. use of a hash function for authentication and encryption of tickets;
2. operation is not based on a synchronised clock;
3. if Alice wants to communicate with Bob, it may allow Alice to send a message to Bob to begin the key exchange protocol.

For further details on the Kryptonight operation, the reader can refer to the bibliography section provided at the end of this book.

2.15.5 SESAME

SESAME is an acronym which stands for Secure European System for Applications in a Multivendor Environment and is therefore a project of the European Community aimed at producing the technology for the authentication of users with distributed type access control.

It bases its operation on public-key cryptography, and instead of using a real encryption algorithm, it uses an XOR function with a 64-bit key.

For further details about the SESAME operation, the reader can refer to the bibliography section provided at the end of this book.

2.15.6 IBM common cryptographic architecture

Common Cryptographic Architecture (CCA) provides a service of cryptographic primitives for the confidentiality, integrity and management of keys and a service of personal identification number (PIN). The keys are organised by suitable control vectors. Each key is provided with a control vector with which an XOR operation is performed and is never separated from this vector if not within secure and dedicated hardware. The control vector bits are defined in such a way as to have a well-specified meaning for the use and management of the CCA keys. The control vectors are inserted, together with the encrypted keys, into appropriate facilities called key tokens. The internal tokens of keys are used locally and contain encrypted keys with a master key. The tokens of external keys are encrypted with a special encryption key of keys.

The length of the keys is specified using the control vector bits. Thus, there are keys of length 56 bits used for the functionality of privacy and message authentication. Also there are double length keys (112 bits) used for the management of keys, PINS and of other special services.

The control vector is controlled in a suitable secure processor and must be in accordance with the rules permitted by the CCA for each function that is implemented. When a new key is generated, the control vector specifies its use.

The CCA uses a combination of public-key cryptography and secret key cryptography. The KDC shares a primary key with all users and encrypts the session keys using this primary key. The main keys are distributed using public-key cryptography.

Systems based on CCA are designed to be interoperable.

For further details on the operation of CCA, the reader can refer to the bibliography section provided at the end of this book.

2.15.7 ISO Authentication

ISO also provides a specific protocol, called X.509, for the use of public cryptography. It is capable of providing authentication on a network. Even if particular algorithms are not specified, in any case the specifications recommend the use of RSA. X.509 was originally published in 1988 and, following public review and relevant comments, was revised in 1993 in order to correct security issues.

One of the most important components of X.509 is the management structure of the public-key certificates. In this sense, a different name is assigned to each user and a trusted CA provides a unique name for each user and a signed certificate containing the name of the same user and the relevant public key. The structure of an X.509 certificate is shown in Figure 2.32.

This certificate contains the following parameters:

1. version, which identifies the format of the certificate;
2. serial number, which identifies a unique number within CA;
3. identification of the algorithm, which specifies the algorithm used to sign the certificate and the relevant parameters used;
4. producer, which specifies the name of the person who produced the certificate;
5. period of validity, which represents the date of issue and expiry date;

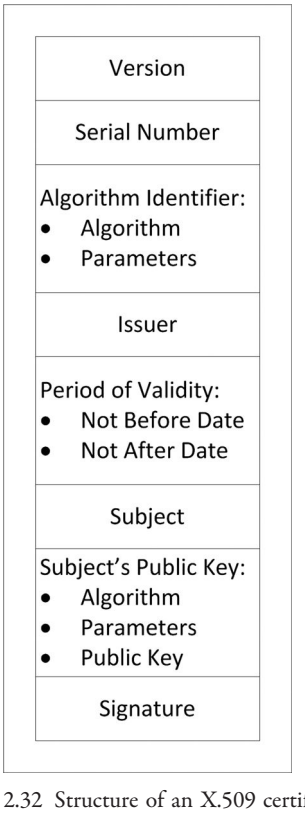


Figure 2.32 Structure of an X.509 certificate.

6. subject, which represents the name of the user;
7. public key, which in fact indicates the information about the public key used, such as the name of the algorithm, its parameters and the corresponding public key;
8. signature, which represents the signature of the CA.

If Alice wants to communicate with Bob, she must retrieve the certificate from CA and verify its authenticity. If both use the same CA, this is very easy since Alice needs only verify the signature of the CA on Bob's certificate. If both use two different CAs then the process becomes more difficult. In this case, it is necessary to resort to a cascade process, with various CAs placed in pyramid order in which each higher level CA certifies the CAs at lower levels, as shown in Figure 2.33.

In the case of Figure 2.33, Alice is certified by CA_A and Bob is certified by CA_B. Alice knows the public key of CA_A. CA_C has a certificate signed by CA_A and in this way Alice can prove it. In addition, CA_D possesses a certificate signed by CA_C, CA_B has a certificate signed by CA_D and Bob possesses a certificate signed by CA_B. Moving along the hierarchy shown up to a common point, in this case CA_D, Alice can verify the validity of Bob's certificate.

Certificates can be stored in suitable archives scattered about on the network in order to facilitate the operations of retrieval. Users can exchange them with each other without difficulty. In the event of expiry of a certificate, the same should be removed from each network archive, keeping only one copy of the CA for any future disputes.

Certificates can also be revoked in the event of key or CA compromise or upon unavailability of the CA to certify a given user. Each CA should maintain an archive of all those keys revoked but not yet expired. If Alice receives a new certificate, she should check to see if the same has been revoked, consulting the archive of the keys revoked on a network that can be found on the network or that have already been previously downloaded onto her computer. The key revocation mechanism represents the most vulnerable element of the system.

If Alice wants to communicate with Bob, the first thing she must do is to recover Bob's public key from a trusted network file, which is called a path certificate between Alice and Bob. Then she can start a one-way, two-way or three-way authentication protocol. The one-way protocol represents a single communication from Alice to Bob that is able to ensure the identity of Alice and Bob and the integrity of their communications and is able to protect itself from attacks. The two-way protocol also adds Bob's response and ensures that Bob, and not an attacker, has sent the requested

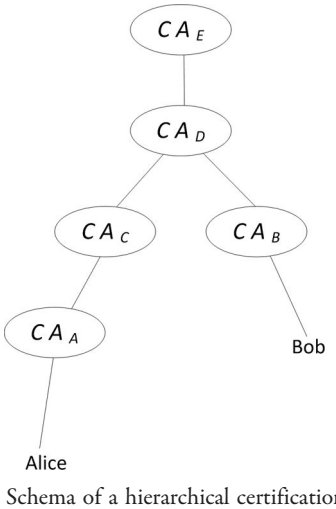


Figure 2.33 Schema of a hierarchical certification structure.

response. This protocol also ensures the security of communications and resistance to any possible malicious attacks. Both the one-way and two-way protocols use the stamping service. The three-way protocol adds a further message from Alice to Bob and eliminates the need to resort to the service of stamping.

The one-way protocol works according to the following steps:

1. Alice generates a random number R_A.
2. Alice generates a message $M = (T_A, R_A, I_B, d)$, where T_A is Alice's stamp, I_B Bob's identity and d arbitrary data that can be encrypted with Bob's public key E_B for security.
3. Alice sends $(C_A, D_A(M))$ to Bob, where C_A is the certificate of Alice and D_A her private key.
4. Bob verifies C_A and obtains Alice's public key E_A, ensuring that it is still valid.
5. Bob uses E_A to decipher $D_A(M)$, verifying the signature of Alice and the integrity of the information signed.
6. Bob checks the accuracy of I_B in M.
7. Bob verifies T_A in M and confirms that the message is valid.
8. Bob may optionally check R_A in M in an archive of old numbers to make sure that the message is not replicated.

The two-way protocol is in practice composed of a one-way protocol from Alice to Bob and a one-way protocol from Bob to Alice.

The three-way protocol operates like a two-way protocol but without T_A and T_B identifications.

2.15.8 Privacy Enhanced Mail

Privacy Enhanced Mail (PEM) is the standard adopted by the Internet Architecture Board (IAB) to provide a secure email service on the Internet. It was initially designed by the Internet Research Task Force (IRTF) Privacy and Security Research Group (PSRG).

The PEM protocol provides the services of encryption, authentication, message integrity and key management.

It is a so-called inclusive protocol, that is it is compatible with a wide variety of systems for managing keys, including public key and secret key schemas for the encryption of data encryption keys. Symmetric cryptography is used for encryption of text messages while hash type algorithms are used for verification of the integrity of messages.

This protocol provides three advanced services of privacy which are confidentiality, authentication and message integrity; not requiring any email management system processing specification.

2.15.9 TIS/PEM

Subsequent to PEM a variant called Trusted Information System (TIS)/PEM was developed, suitable for UNIX, DOS and Windows-based operating systems.

It supports the existence of a multiple certification hierarchy on the Internet even though it recommends a single certification hierarchy.

2.15.10 Message Security Protocol

The Message Security Protocol (MSP) represents the equivalent of PEM in the military field. It was developed by NSA in the late 1980s and is aimed at signing and encryption of messages.

MSP, as well as PEM, was designed to implement many cryptographic algorithms to provide services of signature, hashing and encryption, and is able to run on dedicated chips.

2.15.11 Pretty Good Privacy

Pretty Good Privacy (PGP) is a freeware program for email security that uses IDEA for data encryption, RSA with keys up to 2,047 bits for the management of keys and digital signature and MD5 for the calculation of one-way hash functions.

The random generation of public keys uses as seed the already seen keyboard latency typing while the generation of IDEA random keys is made using the method shown in ANSI X9.17.

The PGP encrypts the private keys of users using the hash function of a passphrase instead of a password.

It uses layered type security. The only knowledge that a possible attacker can acquire with regard to an encrypted message is that of membership of the message itself as long as he/she knows the relevant identification of the key container.

PGP is very efficient in key distribution. It does not use an external certifier but a trusted Web system. Each user generates and distributes their own public key that is signed by other users, generating an interconnected system of PGP users In this way, a trust management policy is not specified and every user is free to decide which other users they will trust. Each user has a set of signed public keys which they store in a file called public keyring. Each key in the keyring is marked with a field in which is written the degree of confidence entrusted to the respective owner. There is also another field, called trust in signature, which indicates that the user trusts certification of the public key of another user. Finally, there is another field, named trust in owner, which indicates that the user trusts the owner of a key in relation to signing the public keys of other users. PGP also however gives the user the freedom of using keys that are not trusted advising the same appropriately.

The weak point of the certification system is the withdrawal of keys in that it is not possible to know if a key is compromised. In fact, if a user's key is stolen, the same user issues a certificate of withdrawal of the key which is distributed to all other users. Given the mechanism of connection among the various users, it is not the case that this certificate reaches them all. In addition, since Alice must sign her own certificate with her own private key, if in case the private key has been stolen, a significant problem may arise.

The latest version of PGP also implements other encryption algorithms, such as the triple-DES, SHA and others. It also implements additional functionality such as the separation between encryption keys and keys of digital signature and advanced procedures for the withdrawal of keys and management of the same.

PGP is virtually available for all operating systems and is also available for a fee with advanced functionality.

If we trust the security of the IDEA algorithm, then PGP is a commercial program that ensures the same level of security of encryption of military level.

2.15.12 Smart card

A smart card is essentially a plastic card the size of a credit card that contains an electronic chip and these days has an application across all levels.

The electronic chip on the smart card can contain a microprocessor, a RAM-type memory, a type of ROM memory and possibly a EPROM or EEPROM-type memory.

These cards are characterised by their own operating system, their own programs and their own data and are characterised by a high level of intrinsic security, given that they can accompany us at all times and thus they can be stored appropriately.

They are immune to sabotage, suitably protecting the subject emitter of the card.

Given their characteristics, smart cards are widely used in the field of cryptography.

Further insights can be found in the bibliography section provided at the end of this book.

2.15.13 Public-key cryptographic standards

Public-Key Cryptographic Standards (PKCS) represent an attempt by RSA Data Security (RSADS) to provide an industry standard for public-key cryptography. Typically, this type of approach is used by ANSI, given its characteristics. But, given the current situation in the field of cryptography, PKCS has considered it appropriate to develop this product for its own purposes.

PKCS standards are conveniently numbered and a brief description is given, taking into account that PKCS 2 and PKCS 4 were incorporated within PKCS 1:

1. PKCS 1 illustrates a method for encryption and decryption using RSA, aimed at digital signature and digital envelope described in PKCS #7. With regard to the digital signature, the hash function of the message is performed and the relevant result is encrypted with the private key of the user. Both the message and the hash are shown together as shown in PKCS 7. With regard to the digital envelope, the message is first encrypted with a symmetric algorithm and subsequently the message key is encrypted with the public key of the recipient. Both the encrypted message and the encrypted key are represented according to that shown in PKCS #7. Both methods are compatible with PEM.
2. PKCS 3 illustrates a method for the implementation of the Diffie–Hellman key exchange.
3. PKCS 5 illustrates a method for encrypting messages with a secret key that is derived from a password. This method uses both MD2 and MD5 to calculate the key from the password and encrypts using DES. It is designed to encrypt private keys for transfer between computers but can also be used for encrypting messages.
4. PKCS 6 illustrates a syntax for public-key certificates which is nothing other than a super-set of X.509.
5. PKCS 7 shows a general syntax for data that can be encrypted or signed. This syntax also allows other attributes such as stamping. It is compatible with PEM and the signed and encrypted message can be converted to PEM format and vice versa.
6. PKCS 8 illustrates a syntax for information concerning private keys. It is used to encrypt such information.
7. PKCS 9 shows the types of attributes of certificates extended to PKCS 6.
8. PKCS 10 shows a standard syntax for certificate request. A certificate includes a name, a public key and a set of attributes, all signed by the person requesting certification. Certificate requests are sent to a CA that transforms the requests into X509 public-key certificates or PKCS certificates.
9. PKCS 11 illustrates the characteristics of an interface for devices for laptop encryption of any kind.
10. PKCS 12 illustrates a syntax for programming using software of user public keys, protected private keys, certificates and other cryptographic information. Its purpose is that the standardisation of a single key file be used in multiple applications.

In substance PKCS provides a standard for data transfer based on public-key cryptography.

2.15.14 CLIPPER

CLIPPER is an electronic chip designed by NSA, immune to sabotage, which is used for the encryption of voice conversations that is also used by the US Federal Government.

In this sense, it allows a government interceptor to recover the session key and then to decrypt the conversation in progress generating many disputes regarding privacy.

2.15.15 CAPSTONE

CAPSTONE represents the other electronic chip designed by NSA and used by the federal government. It implements the following features: a skipjack algorithm, an algorithm for the exchange

of public keys, an algorithm for digital signature, a hashing algorithm, an algorithm for general exponentiation and a random number generator that uses a pure noise source.

It provides functionality suitable for secure electronic commerce and other applications via computer.

2.15.16 Other systems

The applications described so far represent only the initial historical group, since, with time, many others were created that will not be illustrated for reasons of space, with reference to the bibliography section provided at the end of this book for further information.

This page intentionally left blank

CHAPTER 3

STEGANOGRAPHY

3.1 Introduction

Steganography is a type of hidden communication. The term derives from the ancient Greek steganos (hidden) and grafein (writing). While cryptography is intended to protect the contents of messages, generating incomprehensible content, steganography hides the message itself, without leaving a hint of its existence.

While steganography hides a message within another message, leaving the normal image, video or music file almost unaltered and in any case making changes to the original files imperceptible, cryptography encrypts the message, making a set of incomprehensible symbols.

While a set of graphic images, video or music files within which the message is hidden does not arouse suspicion, a file of incomprehensible characters does.

While steganography requires attention when reusing images or music files, cryptography requires attention when reusing keys.

While there is no restriction in using steganography, there are certain restrictions using certain forms of cryptography.

3.2 History of steganography

Steganography has a very ancient history. A number of historical examples are illustrated in this chapter.

3.2.1 The Egyptians

The Egyptians, thanks to the use of hieroglyphics, are considered to be among the first users of steganography. Even though hieroglyphics are a form of writing, in some cases, they were designed in a way as to make their meaning comprehensible only to those who could read and write, as they could understand the code.

3.2.2 The Greeks

The Greeks used various methods of steganography. For example, during the war against the Persians, they used wooden tablets on which they wrote secret messages to be sent, and then covered the tablet

with wax in such a way as to hide the message and to ensure the safety of the messenger who was travelling by horse over long distances.

Another method is the one described by the historian Herodotus in which the message was written on the shaven head of the messenger, waiting for the next regrowth of hair. It was not a fast method of communication but it ensured that the messenger could pass enemy controls and barriers without the message being intercepted. After arriving at the destination, the messenger's head was shaved and the message was successfully read.

3.2.3 The Chinese

The Chinese always used messengers like the Greeks, but the message was written on silk fabric which, subsequently, was enclosed in a ball of wax, easy to carry anywhere and everywhere.

3.2.4 Gaspar Schott

Gaspar Schott, in his book *ScholaSteganographica*, described a method to hide secret information in a musical score, by associating letters with notes. The resulting music was certainly not pleasant to listen to, but hid secret messages effectively.

3.2.5 Johannes Trithemius

The monk Johannes Trithemius is considered to be one of the founders of modern cryptography. He wrote the work *Steganography* in three volumes around 1,500, suggesting various methods to hide information. It is only recently that the mysterious code illustrated in the third volume has been deciphered.

3.2.6 Giovanni Porta

The Italian scientist Giovanni Porta, born in 1535, contributed significantly to both steganography and cryptography. He described a technique for writing on the inside of a boiled egg using a suitable mixture as ink. By writing the message on the shell, the ink penetrates within and is imprinted on the cooked egg white. The only way to read the message is to remove the outer shell of the boiled egg.

In his book *De FurtivisLiteramNotis*, he identifies three types of cryptographic systems: transposition, substitution with a symbol and substitution with another letter. His studies were profitably used by his successors.

3.2.7 GirolamoCardano

GirolamoCardano was an able physician, astrologer and mathematician. He wrote 131 books on topics as diverse as: mathematics, astronomy, physics, poisons, air, water, dreams, wisdom, morality and music. His main contribution to steganography is the Cardano grid, for which he is still remembered.

The system consists of applying a grid over a perceptible message. The grid is perforated above the letters that are part of the hidden message. In this way, by applying the grid over the basic message, it is possible to directly read the hidden message. The key to access the message is the position of the holes on the grid.

3.2.8 Blaise de Vigenere

Blaise de Vigenere is a famous name in the history of cryptography through developments introduced in the replacement polyalphabetic cipher. He studied the texts of Trithemius, Cardano and Porta and created a cipher, forgotten until its rediscovery in the nineteenth century.

3.2.9 Auguste Kerckhoffs

Auguste Kerckhoffs contributed significantly to cryptography rather than to steganography. His most famous book is *La CryptographieMilitaire*, its objective being an exploration of modern cryptography systems. As such, he put in place a system to encrypt communications via telegraph, which was the most common communication system used in his time. He proposed a series of principles that are still valid and are listed as follows:

1. A system should be, if not theoretically unbreakable, at least unbreakable in practice.
2. The compromise of a system should not cause problems for the correspondent.
3. The key should be easy to remember without annotations, and should be easy to change.
4. The cryptogram should be transmitted via telegraph.
5. The apparatus or documents should be transportable and usable by a single person.
6. The system should be simple and not require specific training or a long series of instructions.

Kerckhoffs even proposed a principle that is still valid, which states that if an encryption method is known by an adversary the security of the system must be entirely based on key security.

3.2.10 Bishop John Wilkins

In 1641, Bishop John Wilkins published a book anonymously, entitled *Mercury* or *The Secret and Swift Messenger*. In this book, in addition to describing several methods of cryptography, he also described a method for writing with fluorescent ink, obtained from natural products, only visible under exposure to ultraviolet rays.

3.2.11 Mary Queen of Scots

Mary Queen of Scots, a Catholic, used cryptography and steganography to communicate during the period in which the Catholic lords wanted to replace her with Queen Elizabeth, a Protestant.

Unfortunately for Mary, Queen Elizabeth's First Secretary, a certain Francis Walsingham, was also head of the secret service, and managed to accumulate sufficient evidence to impeach Queen Mary, as organiser of the plot against Queen Elizabeth.

The problem was that the communication system used was not sufficiently secure and ciphers were breached and, as such, the full meaning could be read.

3.2.12 George Washington

Steganography played a crucial role during the United States War of Revolution and was very useful to George Washington on many occasions. Numerous methods were employed and are not reported for reasons of space.

3.2.13 Air mail by pigeons in Paris in 1870

During the Franco-Prussian war of 1870 to 1871, Paris was under complete siege and communications with the rest of the country were completely interrupted. Various attempts were made by the French to restore communications, but the most efficient proved to be the use of pigeons for sending air mail. This method was very vulnerable because pigeons could be attacked by hunters, predators etc., but they still managed to ensure the delivery of more than 95,000 private messages.

In the beginning, the message was rolled up and tied to a pigeon's foot, but this method made it easy for the message to be read in case the pigeon is captured. Subsequently, a system was devised to compress the message using micro-photographic techniques.

The use of pigeons became an integral part of all European armies and became obsolete when wireless communication systems (wireless) were developed.

3.2.14 The First World War

During the First World War, there were different situations in which steganography was used. A widely used method was the Turning grid, which was a progression of the Cardano grid. It consists of a normal grid divided into cells in which certain cells are perforated. To use the Turning grid, the encryptor writes the first sequence of letters in the grid. Once the spaces are completed, the grid is rotated 90° and the other letters are written again and so on, continuing to rotate the grid until the message to be written is accomplished.

The Germans equipped their troops with different grids, capable of adapting to messages of different length. The French were able to develop a method of attack against this encryption system and the grids lasted only 4 months.

During the First World War, with the introduction of radio communications, the need for security became paramount and both cryptography and steganography improved in leaps and bounds. The use of invisible ink proved to be very useful, because it is detectable only on exposure to heat sources.

3.2.15 The Second World War

Steganography went through a period of great development during the Second World War. After the Japanese attack on Pearl Harbour against the Americans, the Americans introduced a fierce campaign of censure against everything that could be a source of hidden steganographic messages, such as chess games by post, crosswords, drawings and blank sheets written on with invisible ink.

Censorship legislation outlawed the sending of any type of text that was not clear and understandable, or that was not written in English, French, Spanish or Portuguese. Even the mass media were censored and were not allowed to send certain stories via telephone or telegraph.

Personal announcements such as those for lost dogs were also censured. Street interviews were also no longer permitted, avoiding the risk of such persons being able to send secret messages. Children's Christmas songs were also censored. All of this was to avoid the risk of exchanging steganographic messages.

The German Nazis, for their part, used photographic microdots. This technique was to reduce the message to be sent to a single point, using photographic techniques, which was affixed to the inside of an innocuous message, such as a normal punctuation mark.

They also used acrostics that consisted of intentionally composed texts so that the first letter of each paragraph formed a message.

3.2.16 The Vietnam War

Steganography was also used during the Vietnam War. For example, an officer, when he/she was captured and was put in front of the media, sent the message "torture" using Morse code, blinking his eyes appropriately. Many other techniques were used to communicate.

3.2.17 Margaret Thatcher

Steganography was also used by the British Prime Minister Margaret Thatcher in the 1980s, by using a *digital watermarking* system (which will be discussed in the chapter 4). She, tired of the continuous leaking out of confidential documents, ordered that all documents should be encoded with the identity of the writer, in such a manner as to be traceable to the writer in the event of the theft of the same document.

3.3 Principles of steganography

Steganography is always applied when a hidden communication is necessary. Typically, two subjects, Alice and Bob, intend to exchange a message secretly without a third party, who is able to intercept and alter the message, being aware of it.

If cryptography attempts to hide the content of the message, leaving the presence of the message visible, steganography attempts to completely hide the existence of an exchanged message. In this situation, Alice and Bob are aware that there could be a third party who can conduct passive, active or malicious attacks.

3.3.1 The background to secret communication

Most steganographic systems are represented by the diagram shown in Figure 3.1.

As can be seen in Figure 3.1, Alice wants to send a secret message m to Bob, choosing randomly, via a private random source r, an innocuous message c, called *cover-object*, which can be transmitted to Bob without arousing suspicion, within which the secret message m is hidden, optionally using a key *k*, called *stego-key*. This operation must be carried out very carefully in order to generate a final message *s*,

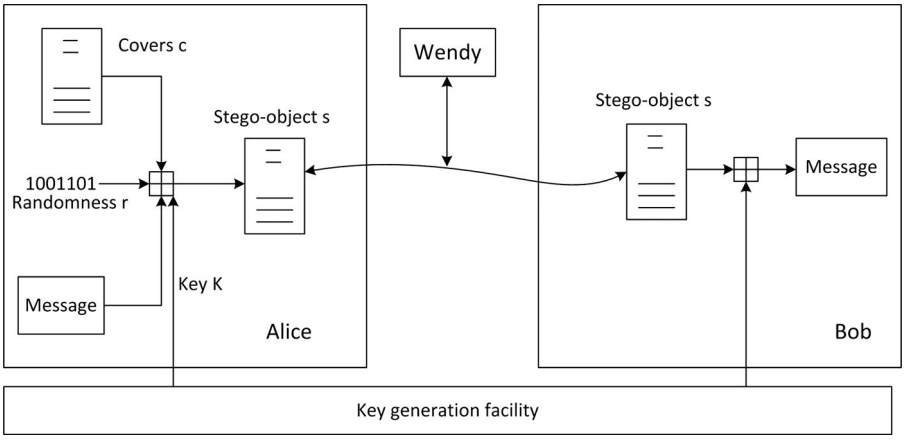


Figure 3.1 Typical diagram of a steganographic system.

called *stego-object*, which does not arouse any suspicion and does not allow the detection of the existence of a secret message m, hidden within.

In a perfect system, a normal *cover* should not be distinguishable from a *stego-object*, either by a human or by a computer that looks for statistical connections. From a theoretical point of view, a *cover* is an image file, a digital audio file, a video file, a text file, etc.

At this point Alice transmits s to Bob over an insecure channel, hoping that the third party (Wendy) does not notice the hidden message in s. Bob can retrieve the message m because he knows the steganographic method used and the key k used in the concealment process. This extraction process should be possible without having the original *cover* c.

A third party who intercepts the communication should not be able to decide if the sender is of the active kind, and if he is, that is, capable of sending a *cover* containing secret messages rather than *covers* containing no message, in this sense, the security of hidden communication is based on the inability to distinguish the *cover-object* from the *stego-object*.

In practice, not all types of files can be used as a *cover* to carry out secret communications, as the changes effected by the steganographic process should not be visible to anyone else, except the person involved in the secret communication process. It is very important, therefore, that the file used as a *cover* contains a sufficient amount of superfluous data that can be replaced, without being noticed, with secret information. For example, it is known that any data that originates from a physical process is affected by a certain amount of noise that is random in nature. This noise can be used to hide secret information.

In any case, a *cover* should never be used twice, because a hacker who has access to the two versions could easily detect and possibly reconstruct the secret message. It is strongly recommended that multiple uses of a *cover* are avoided and that both the sender and the recipient destroy them after their use.

Steganography can be divided into steganographic models, the principle ones being:

1. injective steganography;
2. substitutive steganography;
3. generative steganography;
4. selective steganography;
5. constructive steganography.

With injective steganography, the secret message is literally injected into the *cover*. In most cases, this results in an increase in the memory occupation of the original cover and this may be a clue for a potential hacker to the presence of a secret message within the *cover*. As such, the larger the size of the *cover* is, the more information can be inserted, and the lesser the probability of detection of the secret message.

In substitutive steganography, part of the information of the *cover* is appropriately replaced with the secret message information, reducing as much as possible perceptibility of the alteration. This technique results in no increase to the size of the *cover* and as such is very effective. It is one of the most commonly used techniques. A typical example is the replacement of the noise in the communication channel with the secret message.

In generative steganography, the *cover* is literally generated to reproduce, in a suitable manner, the secret message using an appropriate algorithm. This technique is very effective in that it is very difficult to detect the presence of a secret message within the *cover*. If the *cover* is an image, it is still very difficult to generate a realistic image.

In selective steganography, only files that already have a property are selected, from all those possible files, and this property is used to hide the secret message. As a result, this technique is very time consuming and as such is rarely used even though it is very resistant to attacks.

In constructive steganography, the operation is very similar to that of the replacement steganography. In general, it is exploited by using the channel noise, where an appropriate model is first

constructed and then replaced the noise with the secret message, always following the model created. Even in this case, it is extremely difficult to intercept the secret message, but its weakness is in creating a valid noise model.

The three basic types of steganographic protocols are:

1. pure steganography;
2. secret key steganography;
3. public key steganography.

3.3.1.1 Pure steganography

When a steganographic system does not require a prior exchange of secret information, such as a *stego-key*, this is called pure steganography. As a formula, the concealment process can be described as an E mapping: $C \times M \rightarrow C$, where C is the set of possible *covers* and M is the set of possible messages. The extraction process consists of a D mapping: $C \rightarrow M$, extracting the message outside of the *cover*. It is obviously necessary that $|C| \geq |M|$. Both the sender and the recipient must have access to concealment and extraction algorithms, but the algorithm should not be public.

A formula definition of pure steganography is given by: the quadruple $S = \langle C, M, D, E \rangle$, where C is the set of possible *covers*, M the set of possible messages being $|C| \geq |M|$, E: $C \times M \rightarrow C$ the concealment function and D: $C \rightarrow M$ the extraction function, the property being valid $D(E(c, m)) = m$, for all the $m \in M$ and for all the $c \in C$, is defined as a pure steganographic system.

In most steganographic systems, the set C is selected in such a way that it is represented by messages with a certain and seemingly innocuous meaning that two subjects exchange without arousing suspicion. The concealment process is defined in such a way that the *cover* and the corresponding *stego-object* are perceptually similar. As a formula, the perceptual similarity can be defined by the similarity function as: both sets of C are not empty. A sim function: $C^{\dagger}2 \rightarrow (\vdash -\infty,1]$ is defined as a similarity function on C, if for each $u, v \in C$ is $\text{sim}(u, v) = 1$ if $u = v$ and $\text{sim}(u, v) < 1$ if $u \neq v$.

In the case of digital images or audio, the similarity function is virtually represented by the correlation function. Given the definition of the similarity function, it can be said that the majority of steganographic systems try to observe the sim condition $(c, E(c, m)) \approx 1$ for all the $m \in M$ and for all the $c \in C$.

Covers that have never been used before should not be accessible to a potential hacker. For each communication process, the *cover* should be chosen randomly, perhaps by choosing those that are not altered by the concealment process. This selection can be performed via the similarity function. Thus, during the concealment phase, the sender should choose a *cover* c in a way that $c = \max_{u \in C} sim(u, E(u, m))$.

If the *cover* is generated by a scanning process, the original can be scanned at will, as the noise in the scanning process will always produce slightly different *covers* and the sender can choose the one most suitable for the concealment process. This technique is also called the invisibility selection method.

In some cases, you may have access to a *cover*'s public database. When a hacker, who has access to the original version of the *cover*, is easily able to detect hidden information, the sender selects an element of the database and applies certain changes to obtain a new *cover*, that is used for the secret communication. This method is however not without risks. In fact, if a hacker knows the alteration method of the *cover*, he/she can regenerate the original cover, that is the one not containing the secret message, and use the original to revert to the secret message even without knowing the steganographic method used, but by simply making a comparison.

Certain steganographic techniques use both cryptography and steganography, by encrypting the message before it is hidden in the *cover*. As such, it is more difficult for a hacker to reveal the concealed encrypted text. In any case, efficient steganographic techniques do not require prior encryption.

3.3.1.2 Secret key steganography

Pure steganography does not require any prior information, apart from the knowledge of the functions E and D, and the security of the system is entirely dependent on its secrecy. This violates the Kerckhoff principle and, as such, is not considered very secure.

It is therefore very important to have access to *a stego-key*, known only to the sender and to the recipient. Obviously the *cover* and the *stego-object* must be perceptually similar.

A formula definition is given by: the quintuple $S = \langle C, M, K, D, E \rangle$, where C is the set of possible *covers*, M the set of possible messages being $|C| \geq |M|$, K the set of possible keys, E_K: $C \times M \times K \rightarrow C$ the concealment function and D_K: $C \times K \rightarrow M$ the extraction function, the property being valid $D_K(E_K(c, m, k), k) = m$, for all the $m \in M$, for all the $c \in C$ and for all the $k \in K$, is defined as a secret key steganographic system.

Secret key steganography of course requires a prior exchange of keys, even if this type of communication compromises the objective of performing a secret communication. This exchange can be simplified by using certain characteristics of the *cover* and a secure hash function, in such a way that the key k can be obtained by applying the hash function to the characteristic of the *cover*. If the steganographic process does not alter the characteristic of the identified *cover*, the recipient can easily calculate the steganographic key. Obviously, the selected characteristic must be heavily dependent on the *cover*, to ensure an adequate level of security, even if the secrecy requested of the hash function violates the Kerckhoff principle. If the *cover* is a digital image, it is advisable to take all the most important colour bits of the same *cover* as characteristic.

Certain algorithms require knowledge of the original *cover* when decrypting. Such systems are of limited interest as they require transmission of the original cover: this transaction is a similar problem to the exchange of encryption keys.

3.3.1.3 Public key steganography

Public key steganography, in a manner similar to public key cryptography, does not require a prior exchange of keys. Public key steganography requires the use of two keys, one public and one private, where the public key is stored in a public database. The public key is used in the concealment process, while the private key is used to extract the secret message.

One possible way to create a public key steganographic system is via the use of a public key cryptographic system. A possible system consists of exploiting the quasi-randomness of an encrypted message. Here, Alice encrypts the secret message with Bob's public key to obtain a quasi-random file and hides it in a channel accessible to Bob, by replacing the natural randomness of the channel with the encrypted message and its quasi-randomness. It is assumed that both the encryption algorithm and the concealment algorithm are known. Bob, who does not know when the message is sent, will try to decipher what is received with his private key, by reconstructing the secret message. A possible interceptor, who knows both the encryption algorithm and the concealment algorithm, may try to extract the message, but if he/she does not have Bob's private key, he/she will not know whether the received random stream of bits is the encrypted message or simply normal channel noise.

In most applications, pure steganography is preferred, as the exchange of keys between the sender and the recipient is not necessary, although pure steganographic protocols do not guarantee security if the hacker knows the concealment method. By implementing a key exchange protocol using public key steganography, the sender and the recipient can exchange a secret key k that may be subsequently used in a secret key steganographic system. As there is no *stego-key* in this system, apart from the exchanged encryption public key, it can be viewed as a pure steganographic process, even if its definition is not completely in-line with the formula provided previously. In this protocol, Alice randomly generates a pair of public – private keys for use with any public key encryption system. Then she hides the key in a known channel but only visible to Bob. Neither Bob nor a possible hacker can know whether the

channel contains the key or random bits. In any case, Bob is advised that a *stego-object* sent by Alice contains the public key of the latter and tries to extract it. Once the key is removed, he hides a key k, chosen at random, with a short confirmation message, both encrypted with Alice's public key in a *cover*, and sends everything to Alice. At this point, even if a hacker attempts to retrieve the hidden information, he/she sees only the quasi-random bits and nothing else, as the hidden message is suitably encrypted. Alice, waiting for the message from Bob, extracts the secret information and decrypts it with her private key. At this point Alice and Bob share their own *stego-key*.

The protocol is susceptible to a "man-in-the-middle" attack. If there is an intruder in the middle, he/she can take the first *stego-object* sent by Alice and replace her public key with his/her own. In this way, Bob will encrypt the secret key K with the hacker's public key instead of Alice's public key. At this point, the hacker knows the secret key K and can send it to Alice, encrypted in advance with the latter's public key, and hiding it in a *cover*. Having done this, Alice receives Bob's correct secret key, but does not know that this key is also in the possession of the hacker.

3.3.2 Steganographic security systems

The compromise of a steganographic system is divided into three phases:

1. disclosure;
2. extraction;
3. disabling the hidden information.

In any case, a system becomes insecure when a hacker is able to establish the existence of a hidden message. When a formal security model for steganography needs to be developed, it must be assumed that a potential hacker has infinite computing power available and is able to develop numerous hacking techniques. If it is not possible to confirm the presence of a secret message within a *cover*, then the system is theoretically secure.

3.3.2.1 Perfect security

The theoretical and formal definition for steganographic security systems was formulated by Cachin. The main idea is to refer to a selection of *covers* according to a statistical variable C with a probability distribution P_C. The concealment of a secret message can be seen as a function defined in C, where P_S is the probability distribution of $E_K(c, m, k)$, that is the set of all of the *stego-objects* generated by the steganographic system. If a *cover* is never used as a *stego-object*, then $P_S(c) = 0$. In order to calculate P_S, a probability distribution must be set on K and M. Using the definition of relative entropy $D(P_1 \| P_2)$ between the two distributions P_1 and P_2 defined on the set Q:

$$D\left(\frac{P_1}{P_2}\right) = \sum_{q \in Q} P_1(q) \log_2 \frac{P_1(q)}{P_{2(q)}} \tag{3.1}$$

that measures the inefficiency while assuming that the distribution is P_2 when the true distribution is P_1, then the impact of the concealment process on the distribution P_C can be measured. In detail, it is possible to define the security of a steganographic system in terms of $D(P_C \| P_S)$ as: S is a steganographic system, P_S the distribution probability of the *stego-cover* sent over the channel and P_C the distribution probability of C. S is thus ε-secure against a passive attack if $\mathrm{D}(P_C \| P_S) \leq \varepsilon$ and is called perfectly secure if $\varepsilon = 0$.

There is a theorem (which is omitted from the demonstration for the sake of brevity) that states that there are perfect steganographic systems.

3.3.2.2 The detection of secrets messages

During an attack, a passive hacker must be able to decide whether a *cover* c sent from a sender to a recipient contains or does not contain a concealed message. This operation can be formulated as a hypothesis-testing statistic. Here, the hacker defines a test function f: $C \rightarrow \{0, 1\}$ as:

$$f(c) = \begin{cases} if\ c\ contains\ a\ secret\ message \\ 0\ otherwise \end{cases} \tag{3.2}$$

that the hacker uses to classify the *covers* while passing through an insecure channel. In some cases, the hacker is able to correctly classify the *cover*, while in other cases he/she is unable to detect a hidden message, making a so-called type II error. It is also possible that the hacker detects a hidden message in a *cover* that does not contain not even a single message. In this case, he/she commits a type I error. Ideal steganographic systems attempt to maximise the probability β that a passive hacker makes a type II error. An ideal system should have a value β equal to 1. In the following, it is shown that a perfectly secure steganographic system possesses this feature, provided that the hacker makes type I errors with a zero probability.

For steganographic systems ε-certain, the probabilities α and β that a passive hacker makes type I and type II errors, respectively, are subject to the following theorem, also called the Cachin theorem, from the discoverer's name: let S be a steganographic system that is of the type ε-secure against passive hackers, the probability β that a hacker does not reveal a hidden message and the probability α that a hacker may falsely reveal a hidden message satisfy the following inequality:

$$d(\alpha, \beta) \leq \varepsilon \tag{3.3}$$

where $d\ (\alpha, \beta)$ is the relative entropy binary defined as:

$$d(\alpha, \beta) = \alpha \log_2 \frac{\alpha}{1-\beta} + (1-\alpha) \log_2 \frac{1-\alpha}{\beta} \tag{3.4}$$

In detail, if $\beta = 0$ then $\beta \geq 2^{-\varepsilon}$. It can therefore be said that for steganographic systems ε-secure in which $\alpha = 0$, if $\varepsilon \rightarrow 0$ then the probability $\beta \rightarrow 1$. If ε is very small, a potential hacker will almost certainly fail to detect hidden messages.

3.3.3 The concealment of information in data noise

It has already been said that steganography mostly uses the presence of redundant information in the communication process. Images, video or digital audio contain some redundancy in the form of noise. In this section, it is assumed, without loss of generality, that the *cover* c can be represented in the form of a sequence of binary digits. In the case of digital sound, this sequence is represented by the sequence of samples over time, whereas for digital images a sequence is represented, for example, by a vectorisation of the image. Let $L(c)$ be the number of elements in the sequence, m the secret message and $L(m)$ the message length in bits.

The general principle of steganography consists of putting the secret message in the noise component of the signal. If it is possible to encode the information in a way that it can be indistinguishable from real random noise, then a hacker has no way of revealing the hidden message.

The easiest way to hide information in a sequence of binary digits is to replace the Least Significant Bit (LSB) with a bit from the message m. In floating-point arithmetic, the LSB of the mantissa can be used. As the size of the hidden message is normally much smaller than the number of bits available to hide the information, that is $L(m) \ll L(c)$, the final part of the LSBs may remain unchanged. As a change in the LSB of a byte is in the addition or subtraction of a small amount, the difference lies in the range of noise and is usually unnoticed. This technique does not, however, ensure a high level of

security and a potential hacker can simply try to decode the *cover* as if he/she was the recipient. In addition, this algorithm significantly changes the statistical properties of the *cover*, even if the message consists of actual random bits.

This technique can be improved by selecting, according to a key sequence, the bits to be exchanged rather than using them all sequentially. The key sequence can be created using a pseudo-random number generator. This last technique can be applied to the *cover* flow, that is a *cover* in which the sender does not have access to the entire sequence, as in the case of a digital file during its recording. There are also random access *covers* that allow the sender to access any part of them and to perform the concealment process.

3.3.4 Adaptive and non-adaptive algorithms

The methods shown in the previous section are such that the least significant components of the message, such as noise, are replaced by components of the secret message. Although these components do not have specific statistical properties, the concealment process does not take into account such properties and significantly alters the statistical profile of the *cover*. This variation can be used by a passive hacker to detect the presence of a secret message and to extract it.

3.3.4.1 Laplace filtering

If the steganographic technique hides 1 bit of information in the LSB of a *cover*, represented by an image p in grey scale (which can be represented by a matrix (X,Y)), it is possible to use the discrete Laplace operator defined as:

$$V^2 p(x,y) = p(x+1, y) + p(x-1, y) + p(x, y+1) + p(x, y-1) - 4p(x,y). \qquad (3.5)$$

to detect the presence of a secret message.

By applying the above-mentioned operator to every point of the image p, the second Laplace filtered image is obtained. Since it is expected that adjacent pixels have the same colour, it follows that the histogram of the filtered image has a very sharp cusp curve, centred around zero. The concealment process adds noise to the image that is statistically quite different from the true random noise; if we consider the histogram of the relative filtered image, the result is a curve that is always centred around zero but with a different profile from the sharp cusp.

Laplace filtering is not a detector of the presence of a secret message but is able to provide strong evidence that the image has been modified.

3.3.4.2 Use of *cover* models

To avoid the above-mentioned attacks, it is possible to resort to *cover* modelling according to appropriate characteristics in such a way as to generate adaptive steganographic algorithms that are not always easy to achieve. For example, it is possible to encode secret information in a way that it is statistically indistinguishable from the background noise. Subsequently, secret information is hidden in the parts of the *cover* where the background noise is higher. To do this, the exact *cover* model used must be known, even if a possible hacker, equipped with considerable resources, could develop better models with which to extract the hidden information.

3.3.5 Active and malicious hackers

When designing a steganographic system, it is necessary to pay special attention to the possible presence of active and malicious hackers. Active hackers can alter the *cover* during the communication process. It is generally assumed that a potential hacker cannot alter the full *cover* and its contents but

can only make minor changes. A hacker is deemed malicious if he/she constructs messages and initiates a steganographic communication with a recipient in the name of an original sender.

3.3.5.1 Active hackers and resilient steganography

Steganographic systems are extremely sensitive to changes of *cover*s, such as with images, processing techniques such as filtering, transformations etc., and in the case of audio, such as filtering. They are also sensitive to compression techniques with loss that reduce the amount of information by eliminating the components that cannot be detected by the signal, inevitably deleting the hidden messages.

An active hacker, who is not able to detect or remove a secret message, can simply add random noise to a *cover* to destroy the hidden information. With digital images, the hacker could use image processing techniques or convert the image to another format. These techniques are extremely dangerous and aggressive for steganography. Therefore, an extremely important feature for steganographic systems is resilience. A system is deemed resilient if the hidden information cannot be altered without visibly altering the *stego-object*.

The defined formula for resilience is the following: let S be a steganographic system and T a class of mapping $C \rightarrow C$. S is T-resilient if, for all $p \in T$, there is:

$$D_K(p(E_K(c,m,k)),k) = D_K(E_K(c,m,k),k) = m \tag{3.6}$$

for a secret key steganographic system and

$$D(p(E(c,m))) = D(E(c,m)) = m \tag{3.7}$$

for a pure steganographic system, regardless of the choice of $m \in M$, $c \in C$, and $k \in K$.

Obviously, there is a compromise between resilience and security, as the more resilient a system against *cover* changes is, the less secure it is, because resilience can be achieved by encoding redundant information that visibly degrades the cover and almost certainly alters the probability distribution P_S.

Many steganographic systems are designed to be resilient with respect to a specific class of mapping.

A resilient algorithm must place the information to be hidden in parts of the signal that are more significantly perceptual as the information encoded in noise can be easily removed.

3.3.5.2 Supraliminal channels

If it is assumed that an active hacker can make minor changes to a *stego-object*, then each *cover* that contains significant perceptual information cannot be removed without changing the entire meaning of the *cover*. If the information to be hidden is encoded in a way as to represent a significant part of the *cover*, then the information can be transmitted between two subjects with a high degree of integrity. This type of communication is also called a supraliminal channel.

Cover used for secret communication may be described by the so-called *cover-plots*, which are a formal description of the perceptually significant parts of *covers*. Let S be the set of all the *cover-plots* and f a function f: $S \rightarrow \{0,1\}^N$, called *cover-plot* function. To hide a bit string bit $x \in \{0,1\}^N$ in a supraliminal channel, the sender selects an element $s \in f^{-1}(x)$ and sends a *cover* consistent with *cover-plot* s via an insecure channel. The hacker, suspecting that a supraliminal channel is being used, slightly alters the *cover* in an attempt to remove the secret message hidden in the noise component, but cannot alter the *cover-plot*. Bob extracts the *cover-plot* s from the *cover* that he has received and applies the function f to obtain x.

In practical terms, it must be possible to create a *cover* from every *cover-plot* in such a way that, if an active hacker makes minor changes to the *cover*, the hidden string bit does not change. Furthermore, it must be possible to deduce exactly a *cover-plot* from a *cover* that can be used for secret communication; the function f should be in the public domain and the functions f and f^{-1} should be easily computable.

A *cover* that does not contain any secret message should map to a random string. Under these conditions, the only difference between a *stego-object* and a simple *cover* is that the string $f(s)$ has a specific meaning.

To communicate through a supraliminal channel, the sender cannot explicitly hide a string from the clear meaning, as the hacker, having access to the public function f, can retrieve the information. If the secret message is a random key or an encrypted message, seemingly random, then the hacker thinks that the transmitted message is a random string.

A supraliminal channel can be used to implement a key exchange system. The sender generates a public key E and a private key D, selecting a *cover-plot* $s \in f^{-1}(E)$, and sends a cover consistent with s to the recipient. The recipient retrieves E calculating the *cover-plot* by applying f. A random key k is selected, it is encrypted using E, a *cover* consistent with *cover-plot* s is selected $\in f^{-1}(E(k))$ and it is sent to the sender who can regenerate $E(k)$ in a similar way. The sender decrypts $E(k)$ using their private key D. At this point, the hacker does not suspect that the hidden message in the supraliminal channel has a meaning and that a key exchange is occurring. Although the protocol is susceptible to an attack from "man-in-the-middle" attacks, a non-malicious hacker cannot obtain k from the information exchanged between the sender and the recipient.

The great advantage of the system, in addition to its reduced passing band due to the strength of the encryption, is its flexibility even if it is still not yet evident if a cover-plot function f can be implemented efficiently.

3.3.5.3 Malicious hackers and secure steganography

If a malicious hacker is in evidence, security is no longer a sufficient requirement. If the concealment process is not dependent on the secret information shared between the sender and the recipient, a hacker can create a message because it is not possible to verify the true identity of the sender. To avoid this kind of attack, the algorithm must be robust and secure. A steganographic algorithm can be defined as secure if it satisfies the following four requirements:

1. Hidden messages that use a public algorithm and a secret key where the secret key must be able to exclusively identify the sender.
2. Only those who hold the correct key can detect, extract and prove the existence of a hidden message. No one else should be able to find statistical evidence for the existence of the message.
3. Even if a hacker knows the content of a secret message, he/she should not be able to detect the others.
4. It is computationally complex to detect hidden messages.

3.3.6 Concealment of information within written text

Written text, as opposed to noise, does not contain much redundant information that can be used to effect secret communications. Steganographic methods may attempt to encode information directly within the text or within the text formatting.

There are several techniques designed to hide directly inside messages. For example, we can introduce typos, punctuation omissions and replacing words with their synonyms. Most of these methods are ineffective as they visibly degrade the original text. Furthermore, these methods require interaction with the user and are therefore not automatic.

However, it is possible in any case to create a text message with the sole purpose of it being a *cover* for secret communication. Here, the information is not contained in a *cover*, but is the *cover* itself.

If the text *cover* is transmitted in a particular format, then the secret information can be hidden in the format rather than in the text itself. For example, we might think that if the space between two lines is less than a predetermined threshold, then it is encoded 0; otherwise a 1 is encoded. A similar

system can be used to transmit information in a text, in ASCII format, by appropriately adding the space character.

Other methods can be used such as encoding the information in a way in which the word processor selects the end of the line.

Even if the concealment of information in written text is a valid solution, it is still not clear if this method is secure and resilient, as simple text formatting results in the total destruction of the hidden message.

3.3.7 Examples of invisible communication

In this section, some of the techniques used for concealing information will be illustrated.

3.3.7.1 Steganography in operating systems

There are different techniques that facilitate hidden communication if both parties have access to the same computer.

In particular, it is possible to exploit some of the properties of the ISO/OSI model, such as unused parts of a data frame at the data link layer, the stamping of an IP packet (logic 1 if the time increment is even, logic 0 if the time increment is odd), the detection of collisions at the Ethernet physical level and control of Internet messages.

3.3.7.2 Steganography in video communications systems

Steganography can be used to hide secret messages in a video stream. For example, it can hide messages within a video compression system based on Discrete Cosine Transformation (DCT), or within Integrated Services Digital Network (ISDN) communication system, or even within a cellular telephone communication. The amount of communication concealable depends, of course, on the video stream used as a *cover*.

3.3.7.3 Steganography in executable files

Executable files usually contain a large amount of redundancy, useful for hiding information. Code obfuscation techniques are usually used, developed primarily to protect the re-engineering of software. These techniques attempt to transform a program into an equivalent working program: using a sequence of appropriate transformation, the information can be hidden.

3.4 The principal steganographic techniques

Currently, there are numerous techniques to conceal information, most of which use the substitution method, which consists of replacing redundant parts of information with the secret message. The main disadvantage is the relative fragility in modifying a *cover*. The development of new resilient watermarking techniques (which will be discussed in the chapter 4) has facilitated the development of resilient and secure steganography techniques.

There are numerous ways by which steganographic systems can be classified. A possible way is by the classification of the type of the *cover* being used. Another way is by the type of change applied to the *cover* during the concealment process. If using the latter method, the majority of steganographic systems can be divided into the following six groups:

1. replacement method, replacing the redundant parts of the *cover* with parts of the secret message;

2. domain transformation methods, which hide the information in a signal processing area, such as the frequency;
3. spread spectrum methods, which use the principles of spread spectrum communication;
4. statistical methods, which encode the secret message by changing the statistical properties of the *cover*;
5. distortion methods, using the distortion of the signal to hide the message and measure the deviation from the original *cover* to recover the same;
6. methods for generating the *cover*, which hide the information by creating a special *cover*.

3.4.1 Preliminary definitions

In the following, it will be assumed, in general, that the cover can be represented by a sequence of numbers c_i of the length $L(c)$, being $1 \leq i \leq L(c)$. Digital sound is the sequence of samples taken over time, while images can be the vector representation of the image. It can be assumed that the values of c_i are equal to 0 or 1 in binary files, or between 0 and a maximum (e.g. equal to 256 in the case of 8-bit representation) for multilevel images or audio files. The *stego-object* is usually referred to by s that is a sequence of numbers s_i of the length $L(s)$. Sometimes it is necessary to index the c_i elements of the cover: in this case, the index j is used. If the index itself is to be indexed, then the notation j_i is used. The jth element of a cover will be indicated as c_{ji}. The *stego-key* will be denoted by k, the message by m, its length by $L(m)$ and the bits that compose it with m_i, where $1 \leq i \leq L(m)$. Unless otherwise indicated, it is assumed that m_i can assume the value 0 or 1.

A colour value is normally composed of three components belonging to a colour space. A space often used is the RGB (red, green, blue). As the red, green and blue colours are additive primary colours, each colour can be shown as the weighted sum of these colours. An RGB vector describes the intensity of its components. Another widely used colour space is called YC_bC_r, which makes a distinction between the so-called luminance Y and the chrominance components C_b and C_r. The Y component is responsible for the brilliance of a colour while the components C_b and C_r are responsible for the degree of colour. A colour in the RGB space can be converted into a colour space YC_bC_r by applying the following transformation:

$$Y = 0.299R + 0.587G + 0.114B \tag{3.8a}$$

$$C_b = 0.5 + (B - Y)/2 \tag{3.8b}$$

$$C_r = 0.5 + (R - Y)/1.6 \tag{3.8c}$$

An image C is a discrete function that assigns a colour vector $c(x, y)$ for each pixel (x, y) of the image itself.

3.4.2 Substitution methods

There are a number of substitution methods for various types of files. These methods range from the substitution of the LSBs to the modification of the properties of the image luminance. The basic substitution attempts to encode the secret information by replacing the insignificant parts of the *cover* with the bits of the secret message. The recipient can extract the information if he/she knows where the secret message has been hidden. As during the concealment process only small changes are made, the sender assumes that these changes may not be noticed by a passive hacker.

3.4.2.1 Replacement of the least significant bits

Replacement methods are widely used in the field of steganography and are relatively easy to apply to images and audio. Thanks to these methods, it is possible to conceal a large amount of information at the expense of a small perceived variation on the *cover*.

There are many free software distributions, which base their operation principle on this method.

The method to replace the LSB can be applied to all uncompressed images that can be easily manipulated. This method is very weak because a small transformation applied to the image results in the destruction of the hidden message.

The concealment process consists of choosing a subset $\{j_1, j_2, \ldots \ldots, j_{L(m)}\}$ of the *cover* element and the replacement of c_{ji} with m_i that exchanges the LSB of c_{ji} with m_i which can be equal to 1 or 0. With this method, a large number of gradually increasing weighted bits can also be substituted, resulting in greater variations of the perceptual quality of the original image. In the extraction process, the LSBs are extracted and aligned to regenerate the original message.

To recover the secret message, the recipient must have access to the sequence of indexed items used in the concealment process. Simply, the sender uses all of the *cover* elements, starting from the first element. In general, as the secret message has a number of bits less than the length $L(c)$ of the *cover*, the concealment process usually terminates before the end of the same *cover* file. This situation can result in security problems because in the part of *the cover* where the message is hidden, there will be a certain statistical flow, entirely different from the final part, where the concealed message is not present. To avoid this, there are a number of algorithms that conveniently lengthen bits with the appropriate message to be concealed, in a way that $L(c) = L(m)$, and have the same statistical flow on the *stego-object* obtained.

A more complex approach is the use of a pseudo-random number generator, to randomly distribute the message within the *cover*. Here, both the sender and the receiver must share a *stego-key k* that acts like a seed, for generating the concealment sequence and the recovery of the same sequence. Here the distance between two concealed bits is determined randomly. This technique is especially used in flow *cover*.

3.4.2.2 Pseudo-random permutation

If all the *cover* bits can be accessed in the concealment process, the secret message can be distributed throughout the *cover*. This technique further increases the complexity for a hacker because it is not absolutely guaranteed that the sequential bits of the message will be hidden in the same order.

In a first attempt, the sender could create a sequence $j_1, j_2, \ldots, j_{L(m)-1}, j_{L(m)}$ of the indexed elements and store the kth bit of the message in the indexed element j_k. It must be emphasised that an index can appear more than once in a sequence, because no restriction has been assigned to the pseudo-random number generator. In this case, reference is made to collision. If a collision occurs, the sender can try to insert more than 1 bit of the message in the *cover* elements, corrupting some. If the message is relatively short compared to the number of *cover* elements, the sender can expect a small likelihood of a collision occurring and the possibility of the reconstruction of bits, using an error correction code. This is of course only valid for short secret messages. The probability p that at least one collision will occur can be expressed as:

$$p \approx 1 - e\frac{L(m)[L(m) - 1]}{2L(C)} \tag{3.9}$$

If $L(c)$ is constant, then p converges rapidly towards 1 when $L(m)$ increases. For this reason, the probability of a collision is relatively low only for short messages.

3.4.2.3 *Cover* areas and parity bits

For our purposes, we define the *cover* area as a subset of $C_1, C_2, \ldots \ldots, C_{L(C)-1}, C_{L(C)}$. By dividing a *cover* into different disjointed areas, we can store 1 bit of information in an entire *cover* area rather than in a single element. A parity bit of an area R can be calculated as:

$$p(R) = \sum_{j \in R} LSB(c_j) \bmod 2 \tag{3.10}$$

In the concealment process $L(m)$, R_i disjointed areas are selected where $1 \leq i \leq L(m)$, each of which encodes a secret bit m_i in the parity bit $p(R_i)$. If the parity bit of a cover area R_i does not agree with the secret bit m_i for concealment, then an LSB of R_i bytes is variable, by varying the parity bit of the R_i *cover* area in such a way that $p(R_i) = m_i$. In extracting a hidden message, parities are calculated for all selected areas and properly aligned to reconstruct the secret message. The *cover* areas can also be selected during concealment and extraction in a pseudo-random way using a *stego-key* as a seed.

This method is not very resilient compared to the simple replacement of bits but is much more powerful in many cases. First, the sender can choose which item to change in the *cover* area and this can be done by varying as little as possible the flows of the statistical *cover*. Furthermore, the probability p_0 that the parity bit of a *cover* area consisting of N bits randomly chosen is 0 is approximately 1/2, and roughly independent of the probability p_0 that the probability of an LSB randomly selected in a *cover* element is equal to 0 because:

$$p'_o = \sum_{i=0}^{\frac{N}{2}} \binom{N}{2i} (1-p_0)^{2i} p_0^{N-2i} = \frac{p_0^N}{2} \left[\left(1 + \frac{1-p_0}{p_0}\right)^N + \left(1 - \frac{1-p_0}{p_0}\right)^N \right]$$

$$= \frac{1}{2}[1 + (2p_0 - 1)^N] \tag{3.11}$$

Because $(2p_0 - 1)^N \rightarrow 0$ if $0 \leq p_0 \leq 1$ and N is sufficiently large, it can be concluded that p_0 tends to increase by 1/2 of N regardless of the value of p_0. This means that the effect of the concealment process on a *cover* can be suitably reduced by increasing N.

3.4.2.4 Palette-based images

With palette-based images, only a subset of colours belonging to a specific colour space may be used. Every palette-based image format is composed of two parts: a palette that specifies the N colours as a list of indexed pairs $(i, \mathbf{c}_i)$ that assign a colour vector $\mathbf{c}_i$ to each index i and the current data of the images that assign an index of the palette to each pixel rather than the colour itself. If just a small number of colours in the image are used, this reduces the size of the image file. Two quintessential formats widely used are Graphics Interchange Format (GIF) and bitmap (BMP), although they are used less frequently due to the arrival of compressed formats.

In general, there are two principle methods to hide information within a palette-based image: palette manipulation and image manipulation.

With palette manipulation, it is possible to use the LSBs of the colour vectors to conceal information, using, for example, the method shown previously. Furthermore, because the palette does not need to follow a predefined structure, the information can be hidden using the colour order. Where N is the colours, it is possible to order the same in $N!$ in different ways, ensuring a discrete memory capacity to conceal messages of limited size. This method is extremely resilient as a hacker needs only re-order the colour sequence to destroy the related message represented by this sequence.

With image manipulation, care must be taken because changing the LSB of a pixel changes one colour palette to a similar colour, which is not necessarily perceptually similar, resulting in striking changes to the image. Therefore, it is necessary to pre-order the palette colours in such a way that

perceptually similar colours are close within the same palette. For example, colours can be stored according to the Euclidean distance d in the RGB space, which is represented by:

$$d = \sqrt{R^2 + G^2 + B^2} \tag{3.12}$$

Because the human eye is more sensitive to changes in luminance rather than chrominance, another approach is to organise the palettes according to their luminance component rather than their chrominance. Once the palette has been restructured, it is possible to alter the LSBs of the colour index without causing any perceptible change.

3.4.2.5 Quantisation and dithering

Dithering (in the numerical processing of signals) is a form of noise with a suitable distribution, which is deliberately added to samples to minimise distortion introduced by truncation, where the samples are re-quantised. Dithering is routinely used in the processing of sampled and quantised video and audio signals. Dithering has nothing to do with jitter and the quantisation of digital images can be used for steganography to hide information.

There are several methods to exploit them as for example by predictive encoding. With predictive encoding, the intensity of each pixel is predicted by using the values of the pixels that are found around it, based on both linear and non-linear functions. In its simplest form, the difference e_i between two adjacent pixels x_i and x_{i+1} is calculated and supplied to a quantiser, which generates a discrete approximation Δ_i of the difference in the signal $x_i - x_{i-1}$. When performing this operation, an error occurs at each quantisation step. If the signals are very much correlated, the error tends to 0 and perhaps an entropy encoder would be very useful. The difference in the signal on the receiver side is de-quantised and added to the last signal sample to generate an estimate of the sequence x_i.

For steganographic purposes, the quantisation error can be conveniently utilised, by adjusting the signal difference Δ_i to make it transmit additional information. Here, the *stego-key* is a table that associates a value 0/1 for each possible value of Δ_i.

To hide the information of the ith bit of the message, the difference Δ_i is first calculated. Subsequently, if it does not match (according to the table of association of the *stego-key*) the value of the ith bit of the secret message, it is substituted with the closest value Δ_j that corresponds to the value of the ith bit. The resulting Δ_i values are then sent to an entropy encoder.

There are other methods that insert hidden information within the signal during the dithering process.

3.4.2.6 Hiding information in binary images

Binary images, such as those produced by fax, have a series of redundancies in the distribution of white and blacks pixels. In such cases, it is obviously possible to use substitution schemes that are still prone to transmission errors and thus are not very resilient systems from this point of view.

There are numerous schemes to implement steganography onto binary images, and merely a few will be illustrated in this section, for reasons of space.

One scheme uses the number of black pixels to encode the hidden information. A binary image is divided into rectangular blocks B_i. Let $P_B(B_i)$ be the percentage of black pixels in block B_i and $P_W(B_i)$ the percentage of white pixels in block B_i. Essentially, a block B_i is represented by a logic 1 of the message to be hidden, if $P_B(B_i) > 50\%$, and a logic zero of the message to be hidden, if $P_W(B_i) > 50\%$. If in the concealment process, the requirement for a certain value by the message to be concealed and the number of pixel colours (black/white) do not agree, then the number of the pixel colours is varied appropriately. The best way to change the pixel colour without noticeably altering the image is to do so at the point of black/white and white/black transitions. To make the whole system resilient against transmission errors and other possible changes to the images, it is necessary to make adjustments to the

concealment process. It is in fact possible that certain pixels change colour during transmission and it may happen that $P_B(B_i)$ falls below 50%, destroying the information hidden. For this reason, two threshold values are defined: $T_B > 50\%$ and $T_W < 50\%$, and a resilience factor F that indicates the percentage of pixels can change colour during transmission. The sender ensures that during the concealment process $P_B(B_i) \in [T_B, T_B + F]$ and $P_W(B_i) \in [T_{W-F}, T_W]$ instead of $P_B(B_i) > 50\%$ and $P_W(B_i) > 50\%$ occurs. If too many pixels have to be changed to achieve this goal into a block, they are marked invalid.

3.4.2.7 Reserved or unused space on computers

The use of reserved or unused space can be a good way to hide information without effecting perceptible changes. For example, operating systems generate a certain amount of unused space when allocating files, using typical well-defined cluster sizes, regardless of the actual length of the file. As such, the space that is not used by clusters can be used to hide information without it being visible in the disc directory. In addition, space that is not used by the image or audio *header* files can be used to hide information.

Another method to hide information is to create a hidden partition that is not visible when the system is started normally but is visible only if disc configuration utilities are started.

The ISO/OSI protocol models also have characteristics that facilitate the concealment of information. The same Transmission Control Protocol / Internet Protocol (TCP/IP) packets have a certain amount of space not used in the *header*, which can be useful for concealing information, even if limited. Considering, however, the large number of packages needed for the transmission of a medium-sized file, it is easy to imagine how this number, although limited, multiplied by all relevant packages, is a considerable amount of available space.

3.4.3 Methods for domain transformation

We have already seen that LSB techniques are a very simple way to hide information. These however are very vulnerable to even slight variations of the *cover*. A hacker could simply apply a technique to completely destroy the hidden information or a minimum compression with loss.

It was recently discovered that steganographic techniques that operate in the domain frequency of a signal are generally more robust than techniques that operate in the time domain. The more advanced techniques operate in a transform domain, are more resistant to compression, cutting and to signal processing and have the advantage of being barely perceptible. There are many processing techniques such as DCT (already described substantially in Chapter 1) or wavelet transform. They may be applied to the entire image or parts thereof. However, there is a compromise between the amount of information that can be hidden and the resilient of the concealment method. Many transformation domains are independent of the image format and are able to remain unchanged when passing from one format without loss to the other with loss.

Before illustrating steganographic methods in the transformation domain, there follows a brief illustration of the formulas for Discrete Fourier Transform (DFT) and discrete cosine transform.

The DFT $S(k)$ of a sequence s with a length N is defined as:

$$S(k) = F(s)\sum_{n=0}^{N-1} s(n) \exp\left(-\frac{2in\pi k}{N}\right) \tag{3.13}$$

where $i = \sqrt{-1}$ is the imaginary unit. The inverse DFT is given by:

$$S(k) = F^{-1}(s)\sum_{n=0}^{N-1} s(n) \exp\left(\frac{2in\pi k}{N}\right) \tag{3.14}$$

The DCT is instead calculated as:

$$S(k) = D(s) = \frac{C(k)}{2} \sum_{j=0}^{N} s(j) \cos\left(\frac{(2j+1)\pi k}{2N}\right) \tag{3.15}$$

while the inverse transform is calculated as:

$$S(k) = D^{-1}(s) = \sum_{j=0}^{N} \frac{C(j)}{2} s(j) \cos\left(\frac{(2j+1)\pi k}{2N}\right) \tag{3.16}$$

where

$$C(u) = \frac{1}{\sqrt{2}}$$

if $u = 0$ and $C(u) = 1$ otherwise.

DCT has the advantage of providing a true output sequence if the input sequence is true. In image processing, the two-dimensional DCT is used, which is calculated as:

$$S(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} s(x,y) \cos\left(\frac{(2x+1)\pi u}{2N}\right) s(j) \cos\left(\frac{(2y+1)\pi u}{2N}\right) \tag{3.17}$$

while the inverse transform is calculated as:

$$S(x,y) = \frac{2}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(u)C(v)S(u,v) \cos\left(\frac{(2x+1)\pi u}{2N}\right) s(j) \cos\left(\frac{(2y+1)\pi u}{2N}\right). \tag{3.18}$$

We have already seen in Chapter 1, to be referred to for context, that the two-dimensional DCT is the basis upon which the compression algorithm with Joint Picture Expert Group (JPEG) loss can be effectively used for steganography.

3.4.3.1 Steganography in the DCT domain

A technique widely used to hide information is to modulate the relative size of two or more DCT coefficients within an image block.

In fact, during the encoding process, the image is divided into 8×8 pixel blocks and each block is used to hide 1 bit of information. The concealment process begins with the choice of a random block b_i which is used to encode the ith bit of the secret message. Let $B_i = D(b_i)$ be the DCT-transform image block. Before commencement of the communication, the sender and the receiver must agree on the position of the DCT coefficients that will be used in the concealment process. These coefficients are identified by two indices: (u_1, v_1) and (u_2, v_2). The two coefficients should correspond to medium frequency cosine functions in such a way that they are a significant part of the signal and are not deleted during the encoding process. It is also assumed that the concealment process does not significantly alter the *cover*, as the medium frequency DCT coefficients have similar amplitudes. Because the system being illustrated should be resilient with respect to JPEG compression, the DCT coefficients are selected so that the quantised values associated with each other in the JPEG compression algorithm are equal. Therefore, the valid coefficients are represented by the pairs (4, 1) and (3, 2) or by (1, 2) and (3, 0).

Let us now see how the concealment process works. Thus, a block encodes a one logic if $B_i(u_1, v_1) > B_i(u_2, v_2)$ otherwise the same encodes a zero logic. In the concealment process, the two coefficients are exchanged if their values do not correspond to the bit to be encoded. As JPEG quantisation can alter the relative size of the coefficients, the algorithm ensures that $|B_i(u_1, v_1) - B_i(u_2, v_2)| > \in$ for $\in > 0$, adding random noise to both coefficients. The higher the value of $\in$, the better the

resilience of the JPEG compression algorithm. This, of course, is at the expense of the image quality. The sender subsequently carries out an inverse DCT to map the coefficients in the space domain. To decode the image, all available blocks are DCT-transform, comparing suitably the two selected coefficients of each block. If the constant $\in$ and the position of the DCT coefficients are selected appropriately, the concealment process will not generate perceptible alteration in the image. It is expected that the illustrated method is robust with respect to JPEG compression, as in the quantisation process both selected coefficients are divided by the same quantised values.

3.4.3.2 Digital sound steganography

The concealment of information in digital sound is usually more difficult to perform than with digital images, as the human ear is more sensitive than the human eye and the disruption of a music file can be detected at the level of 1 ppm (part per million). Although the limit of perceptible noise increases with the *cover* noise level, the maximum acceptable noise level, however, remains relatively small. It is also true that the human ear is less sensitive to the phase component of sound and this property has led to the realisation of numerous audio compression algorithms.

In the encoding phase, digital data are depicted by a phase shift in the signal phase spectrum. As such, the signal s is divided into a series of N short sequences $s_i(n)$ of length $L(m)$ and a DFT transform is applied, obtaining a matrix of $\varphi_i(k)$ phases and amplitude A_i Fourier transforms. In this regard, it should be noted that:

$$A_i(k) = \sqrt{\mathrm{Re}\,[F\{s_i\}(k)]^2 + Im[F\{s_i\}(k)]^2} \tag{3.19}$$

$$\varphi_i(k) = \frac{\arctan Im[F\{s_i\}(k)]}{\mathrm{Re}\,[F\{s_i\}(k)]} \tag{3.20}$$

As the phase shift between two consecutive signal segments can be easily detected, their phase difference must be maintained over the *stego* signal. The concealment process inserts the secret message only in the phase vector of the first segment signal:

$$\varphi_0^* = \begin{cases} \frac{\pi}{2}\ if\ m_k = 0 \\ -\frac{\pi}{2}\ if\ m_k = 1 \end{cases} \tag{3.21}$$

and creates a new phase matrix using the differences of the original phases:

$$\begin{gathered} \varphi_1^*(k) = \varphi_0^*(k) + [\varphi_1(k) - \varphi_0(k)] \\ \vdots \\ \varphi_N^*(k) = \varphi_N^*(k) + [\varphi_N(k) - \varphi_{N-1}(k)] \end{gathered} \tag{3.22}$$

The sender uses the new phase matrix and the original amplitude matrix to build the *stego* signal using the inverse Fourier transform. Because $\varphi_0(k)$ has been changed, the absolute difference in the phase of all the following segments is changed while their differences are unchanged. As the secret information can be recovered, synchronisation is necessary. The receiver, knowing the length $L(m)$ of the sequence, is able to calculate the DFT and detect the phase $\varphi_0(k)$.

In addition to the encoding phase, the secret information can be concealed within the echo of an audio signal. Here, the aim is to conceal the information within a discrete signal $s(t)$ by introducing an

echo $s(t - \Delta t)$ in the *stego* signal $h(t)$:

$$h(t) = s(t) + \delta s(t - \Delta t) \tag{3.23}$$

where δ is a suitable encoding parameter.

The information is encoded within the signal by changing the delay Δt between the signal and the echo. In the encoding phase, the sender chooses between a delay Δt and a delay Δt^*, and uses them to encode a one logic or a zero logic of the message to be hidden. These delays are carefully selected so as not to be audible.

The basic scheme can only cover 1 bit at a time: as such, a *cover* signal is divided into $L(m)$ blocks before the encoding process. The consecutive blocks should be separated by a random number of samples so as to make detection and extraction of the secret message bits difficult.

Before extraction of the secret message, a sort of synchronisation must be performed because the receiver must be able to reconstruct the $L(m)$ signal blocks used by the sender. Each signal segment can be decoded through an auto-correlation function of the signal cepstrum (in signal theory, the cepstrum is the result of the Fourier transform applied to the signal in decibels of a signal. Its name derives from the inversion of the first four letters of the word "spectrum").

From what has been seen so far, it can be deduced that the encoding phase is resilient with respect to the re-sampling of the signal but is characterised by a low rate of transmission of the hidden message because the latter can only be encoded in the first segment of the signal. The use of echo, on the contrary, provides better performance in most cases.

3.4.3.3 Concealing information and data compression

In many cases, steganographic algorithms conceal information in data compression systems. A typical case may be depicted by a video conferencing system that facilitates the concealment of information within the compressed audio/video stream.

There are many suggested algorithms for performing video or compressed image steganography. An example previously illustrated is the DCT coefficients in representation via JPEG format. A similar approach can be applied to video streams in formats that use compressed frames in JPEG format.

3.4.4 Spread spectrum methods

Spread spectrum systems or SS (Spread Spectrum) were developed around 1950 as methods with a low probability of interception and disturbance. It has already been seen that these techniques can be defined as transmission methods in which the signal occupies a bandwidth in excess of the minimum required to send information. Band expansion takes place through a code that does not depend on data and is synchronised by the receiver for demodulation and the recovery of the original signal. Even if the power of the signal to be transmitted is high, the signal-to-noise ratio in each frequency is small. Even when performing suppression of the spread signal on certain frequency bands, enough information stay in the remaining bandwidth to recover the original signal. Subsequently, spread spectrum techniques make it very difficult to detect and/or delete the signal. Spread spectrum techniques are very similar to steganographic techniques, which attempt to distribute information within a *cover*, making it difficult to see changes that inevitably ensue. As such, spread spectrum techniques are very useful for steganography.

With spread spectrum transmission, it has already been seen that two basic techniques are used: Direct Sequence (DS) and Frequency Hopping (FH). With direct sequencing, the signal to be transmitted is expanded by means of a constant which is called chip rate, modulated by a pseudo-random signal and added to the carrier. With frequency hopping, the carrier signal frequency jumps from one frequency to another via a hopping sequence known only to the sender and the recipient. Direct sequencing is frequently used in *digital watermarking*, which will be discussed in Chapter 4.

The following will illustrate the basic theory that facilitates spread spectrum techniques for use with steganography.

In the general base model, it is assumed that there is a *cover* image with the dimensions N × M in grey scale: this technique can be used in all those fields that can be defined as a scalar product. It is also assumed that the sender and the receiver share a set of $L(m)$ orthogonal images with dimensions of N × M called φ_i, which is the *stego-key*. The receiver generates a *stego* message $E(x, y)$ by executing the following weighted sum:

$$E(x,y) = \sum_i m_i \varphi_i(x,y) \tag{3.24}$$

where m_i is the appropriate weighted coefficients.

As the images are mutually orthogonal, the following relation applies:

$$\langle \varphi_i, \varphi_j \rangle = \sum_{x=1}^{N} \sum_{y=1}^{M} \varphi_i(x,y)\varphi_j(x,y) = G_i \delta_{i,j} \tag{3.25}$$

where

$$G_i = \sum_{x=1}^{N} \sum_{y=1}^{M} \varphi_i(x,y)\varphi_j(x,y)$$

and $\delta_{i,j}$ are the Kronecker delta function (a function that is equal to 1 only when $i = j$).

The sender conceals the hidden message E in a *cover* C, obtaining the *stego-cover* S, by performing the following:

$$S(x,y) = C(x,y) + E(x,y) \tag{3.26}$$

Under ideal conditions, C is orthogonal to all the images φ_i, that is $<\mathrm{C}, \varphi_i> \; = 0$ and the recipient can extract the ith bit m_i of the message by projecting the *stego* image S on the ith base image φ_i as:

$$\langle S, \varphi_i \rangle = \langle C, \varphi_i \rangle + \langle \sum_j m_j \varphi_j, \varphi_i \rangle = \sum_j m_j \langle \varphi_j, \varphi_i \rangle = G_i m_i \tag{3.27}$$

It is therefore evident that the secret message can be recovered by calculating $m_i = \langle S, \varphi_i \rangle / G_i$.

Since in real conditions, C will never be orthogonal to all the base images φ_i, there is an error term $\langle C, \varphi_i \rangle = \Delta C_i$ and the above equation becomes:

$$\langle S, \varphi_i \rangle = \Delta C_i + G_i m_i \tag{3.28}$$

It can be shown that, under appropriate conditions, the expected value of ΔC_i is equal to 0. Therefore, C and φ_i are two-dimensional NM independent random variables. Assuming that all the base images are generated using a random process with a mean zero and that they are independent of the message that must be transmitted, then it follows that the expected value **E** of ΔC_i is equal to:

$$E[\Delta C_i] = \sum_{i=1}^{N} \sum_{J=1}^{M} E[C(x,y)]E[\varphi_i(x,y)] = 0 \tag{3.29}$$

It can therefore be seen how the expected value of the error term equals zero under the assumed conditions.

The decoding operation consists of reconstructing the secret message by projecting the *stego* image S on all the functions φ_i, obtaining an approximate value given by:

$$s_i = \langle S, \varphi_i \rangle = \Delta C_i + G_i m_i \tag{3.30}$$

Under the above conditions, the expected value of ΔC_i is equal to 0, whereby $s_i \approx G_i m_i$. The ultimate goal is to obtain m_i from s_i. If the secret message is encrypted as a string of values equal to 1

and −1 instead of simple binary strings, the values of m_i can be reconstructed using the function "sign", provided that $G_i \gg 0$, defined as:

$$m_i = sign(s_i) = \begin{cases} -1 \, if \, s_i < 0 \\ 0 \, if \, s_i = 0 \\ +1 \, if \, s_i > 0 \end{cases} \tag{3.31}$$

In the case where $m_i = 0$, the information is lost. In some critical cases, ΔC_i could become very large and the recovery of a bit would be impossible. Fortunately this does not often occur.

The major advantage in using spread spectrum techniques in steganography is its resilience with regard to image changes as the concealed information is expanded in a broader frequency band and it is very difficult to completely delete this information without destroying the *cover*. In practical cases, any change in the *stego-cover* results in an increase in the value of ΔC_i, but these changes are not significant for recovering the hidden message until $|\Delta C_i| < |G_i m_i|$.

3.4.5 Statistical methods

Statistical methods change the *cover* in a way that alters the statistical characteristics if one logic of the secret message must be transmitted, otherwise the *cover* is left unaltered. As such, the receiver must know if the *cover* has been altered in order to recover the secret message.

To build a steganographic message of $L(m)$ bits, it is necessary to divide the cover into $L(m)$ distinct blocks $B_1, B_2, \ldots, B_{L(m)}$. A secret bit m_i is inserted into the ith block by altering the statistical properties of block B_i if the bit is equal to 1 or by leaving them unchanged if the bit is equal to 0. The detection of a specific bit is performed through a test function $T(B_i)$ that is able to distinguish whether block B_i has been altered, returning to one logic, or if the block is unaltered, by returning to zero logic. The function T can be interpreted as a hypothesis testing function. The set of systems referred to above are called statistical steganography.

The main problem is in the construction of the function T that must be expressed in a closed form where possible. There are numerous proposals for statistical systems that are not shown here for reasons of space.

3.4.6 Distortion methods

Distortion methods, in contrast to substitution methods, require knowledge of the original *cover*, in order to perform the extraction of the hidden message. Here, the sender applies a modification sequence of the *cover* to obtain a *stego-object* by appropriately selecting the alteration sequence in such a way that it is relative to the message to be hidden. The receiver measures the difference between the original *cover* and the modified *cover* in order to reconstruct the secret message.

In many applications, these methods are not very useful because the recipient must have easy access to the original *cover* and this is not always possible because, with the same ease, a potential hacker could access and directly measure the difference between the two *covers*, reconstructing the secret message.

A simple approach of this kind is by hiding a message within a text, for example by modulating the position of lines and words or by adding spaces and invisible characters. For this, Hypertext Markup Language (HTML) files are very useful.

3.4.6.1 Methods for steganographic distortion in written texts

Numerous distortion techniques are available to hide a secret message within text. For example, it is possible to utilise the spaces between lines, varying them in an imperceptible manner at the top to

represent a one logic, and at the bottom to represent a zero logic. Some lines are left unchanged to ensure a minimum of synchronisation. To decode a secret message, the so-called centroid detection can be used, where centroid means the centre of mass of a line along the horizontal axis.

Another technique that can be used is using the spaces between words, modulating them appropriately, depending on the need to represent a one or zero logic.

3.4.6.2 Steganographic distortion methods in digital images

Distortion methods can also be applied to digital images, using a similar approach to replacement methods in which the sender selects the $L(m)$ *cover* pixels required to conceal the message. This selection can be made using a pseudo-random number generator. To hide a zero in a pixel, the sender leaves it intact, while in hiding a one in a pixel, a random value Δn is added to the pixel colour. This technique appears to be very similar to the replacement technique or LSB. With the LSB technique, the value of the selected colour is not necessarily equal to the bits of the secret message. Furthermore, Δn can be selected in such a way that preserves the statistical properties of the image. The receiver compares all the selected $L(m)$ pixels with the corresponding original *cover* pixels: if the ith pixel differs, then that pixel is equal to 1. There are many variations of this method not shown for reasons of space. For further information, the reader can refer to the bibliography section provided at the end of this book.

3.5 Steganalysis

All steganographic and *digital watermarking* techniques (which will be discussed in Chapter 4) have the following property: given a threshold of human perceptibility, Q is the amount of information that can be manipulated without introducing perceptible distortions. Let P be the part of the *cover* that, if manipulated, produces a perceptible distortion. The potential of transportation C for hidden information is equal to $C = P + Q$. The size of Q is available to both users of the steganographic system and to potential hackers.

The concealment of information, more visible in areas, renders the hidden information, more resilient, but it may also become visible in making alterations that will reveal its existence.

Steganographic tools operate according to different techniques and if these are unknown, or if the *stego-key* is unknown, if used, it is practically impossible to extract hidden information.

The purpose of steganography is to avoid revealing the presence of hidden messages, while the purpose of steganalysis is to reveal the presence of such hidden messages and to eventually destroy them.

Attacks and analysis can be effected in different ways: detection, extraction, confusion and disabling of information.

Every *cover* can be manipulated with the aim of disabling or destroying any hidden information contained in it.

In the field of steganalysis, terminology similar to that of cryptanalysis is applied, where a steganalyst is one who uses steganalysis to reveal hidden messages and a cryptanalyst is one who uses cryptanalysis to decrypt encrypted messages. The focus of steganalysis is towards the *stego-object*, while the focus of cryptanalysis is towards ciphertext.

In steganalysis there are different attack techniques as follows:

1. *stego-only attack*: here only the *stego-object* is available for analysis;
2. *known cover attack*: in this case, both the *stego-object* and the original *cover* are available;
3. *known message attack*: here it is assumed that the hidden message is available to a potential hacker. The hacker compares the differences in the *stego-object* with the hidden message to highlight

possible relationships that can be used for a future attack. This type of attack is, however, very difficult and complex and presents similar difficulties to a *stego-only attack*;

4. *chosen stego attack*: in this case, the *stego-object* and the steganographic algorithm are known;
5. *chosen message attack*: here the steganalyst generates a *stego-object* from a given message using a steganographic algorithm. The purpose of this type of attack is to determine matches in the *stego-object* that can provide additional information on the steganographic algorithm;
6. *known stego attack*: in this case, the steganographic algorithm, the original and the *stego-object* are known.

To verify the presence of information, it is necessary to detect the so-called signatures. Here, it must be remembered that non-standard patterns are used to hide information. For example, typical patterns in a document have spaces between the words or spaces between the lines. Any changes to the original text are not visible when the document is displayed on the screen but become visible if the document is opened and viewed with a word processor.

Even unused areas on a disc may be useful for concealing information. To detect such information, it is necessary to use appropriate programs that display the usage status of the discs.

Suitable filters can be used to capture TCP/IP packets that contain hidden or invalid information within them, usually hidden in reserved or unused areas.

Multimedia files are valid backups to conceal information, even if resulting in distortion, significant or insignificant. This distortion, or perceptible noise, can reveal the existence of a hidden message. With images, if the original file is available, it can be compared with the received file, highlighting the differences to reveal the existence of a hidden message.

Steganographic tools typically hide information blocks, trying not to introduce distortion or excessive noise in the *cover* file.

To analyse an image in searching for hidden information, it is necessary to define the concept of a standard image or standard media. To do this, it is necessary to compare their numerous original images and *stego* in terms of colour composition, luminance and the relationship between pixels, attempting to deduce all those common characteristics that define the standard image. With palette



Figure 3.2 An image hidden in the cover. It is the portrait of Paolina Bonaparte as "Venus Victrix" (Marble sculpture. Galleria Borghese – Rome).

images, the colours are usually ordered from the most used to the least used in order to reduce access time to the colour table. The change in colour values can be gradual, but is rarely more than 1 bit. Instead, images in grey scale change in increments greater than 1 bit. Certain images, such as handmade designs, fractals and clip art can have significant fluctuations in the value of the bits of adjacent pixels. In any case, the existence of a large number of these fluctuations can be a detection tool for the presence of a hidden message. In 8-bit images, small changes in the value of the pixels result in a drastic colour change, revealing the presence of a hidden message. As such, we could increase pixel resolution, but this results in a considerable increase in the memory usage of the original image. As 8-bit images have only 256 colours, the amount of concealable information using LSB techniques is relatively small. Subsequently, if any information hidden in a 24-bit colour image is to be destroyed, it can be reduced to 8 bits, trying to ensure that the colour yield remains the same.

A method widely used to reveal the existence of hidden messages in a *stego* image is to look for the presence of obvious and repetitive patterns. Here, distortions are easily visible to the human eye without having to resort to computer analysis. However, in many cases, there are images without distortions that are used to hide information. Images of 8-bit colour and grey scale using palettes are fairly easy to analyse to search for any hidden images. A method for detecting the presence of a hidden message is to create a vector of unique pixel values within the image that are ordered according to the luminance value: the presence of unusual patterns can be used as a valid indicator.

3.6 Practical examples

In the following are illustrated, by way of example, the original images and relative images, which have undergone the different steganographic techniques described in this section, using the programs on the enclosed CD.

These images are accompanied by the relevant histogram (distribution of the number of image pixels depending on the colour tones) for the three primary colours (red, green and blue) for the purposes of a more quantitative rather than qualitative comparison which can be carried out using the human eye.

The image inserted in the cover image is illustrated in Figure 3.2. It is an image of 336 × 252 pixels, with a 24-bit colour depth.

The cover image however has dimensions of 500 × 334 pixels.

3.6.1 Cryptapix

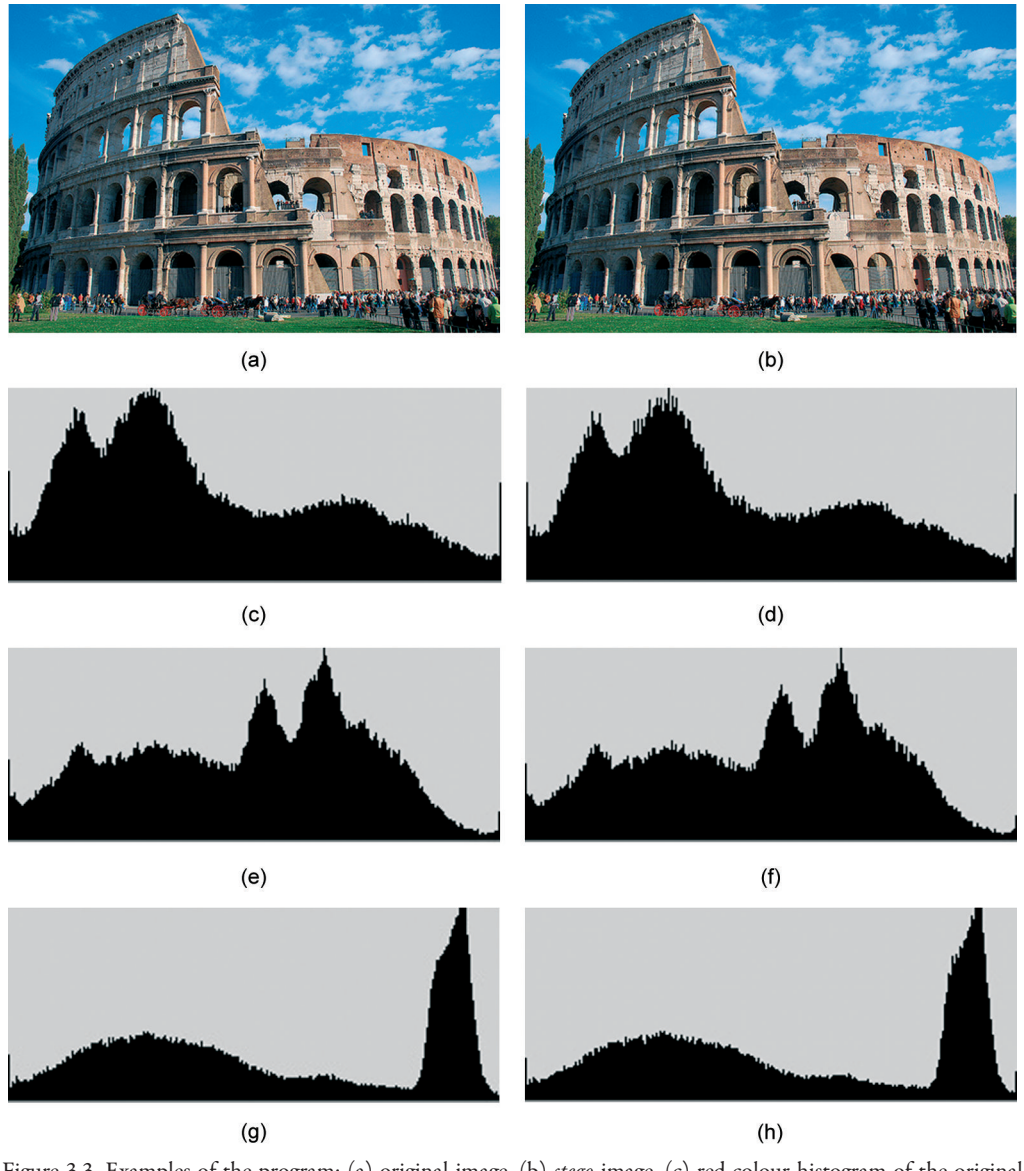


Figure 3.3 Examples of the program: (a) original image, (b) *stego* image, (c) red colour histogram of the original image, (d) red colour histogram of the *stego* image, (e) green colour histogram of the original image, (f) green colour histogram of the *stego* image,(g) blue colour histogram of the original image and (h) blue colour histogram of the *stego* image.

3.6.2 Data stash

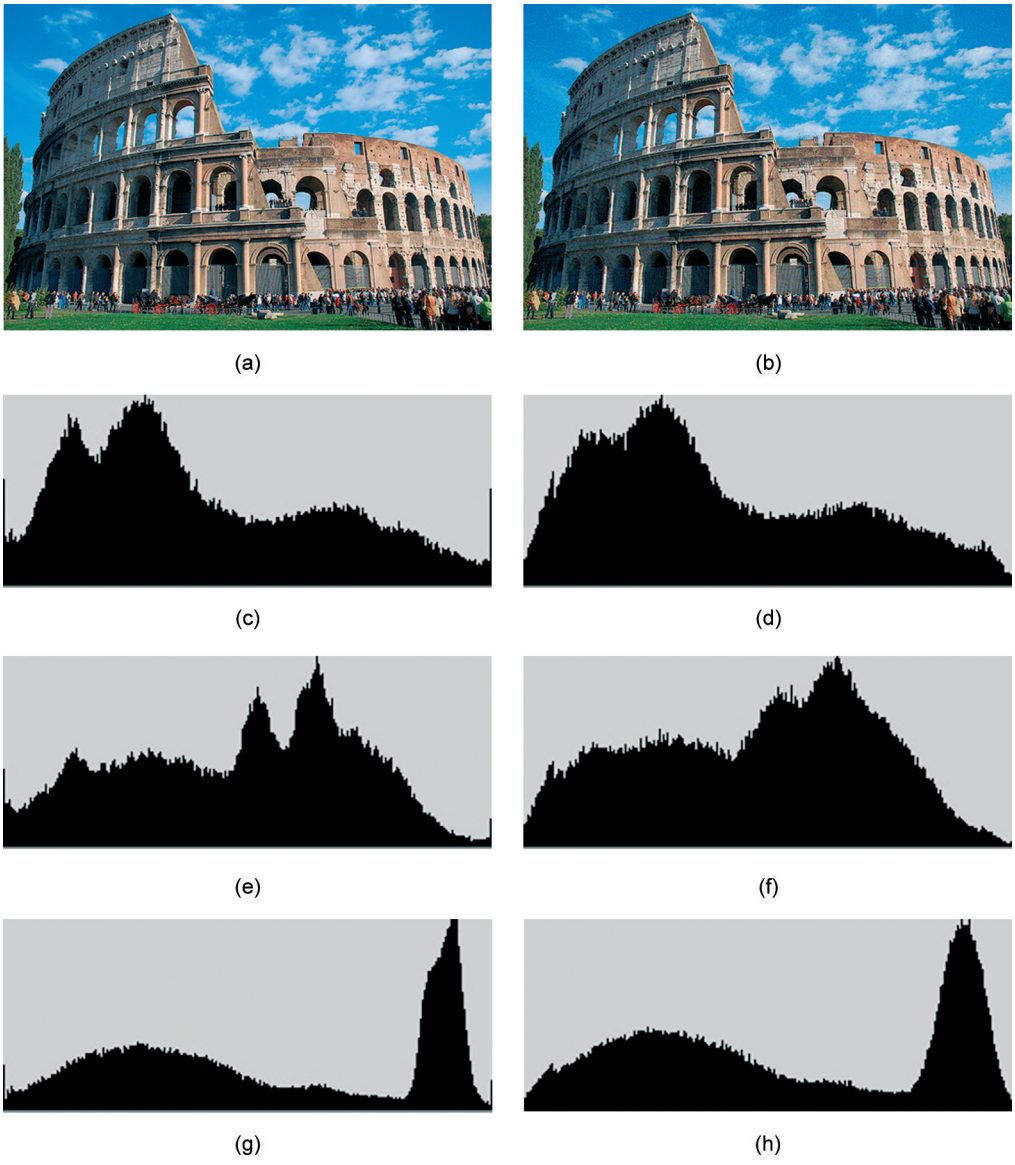


Figure 3.4 Examples of the program: (a) original image, (b) *stego* image, (c) red colour histogram of the original image, (d) red colour histogram of the *stego* image, (e) green colour histogram of the original image, (f) green colour histogram of the *stego* image, (g) blue colour histogram of the original image and (h) blue colour histogram of the *stego* image.

3.6.3 Hermeticstego

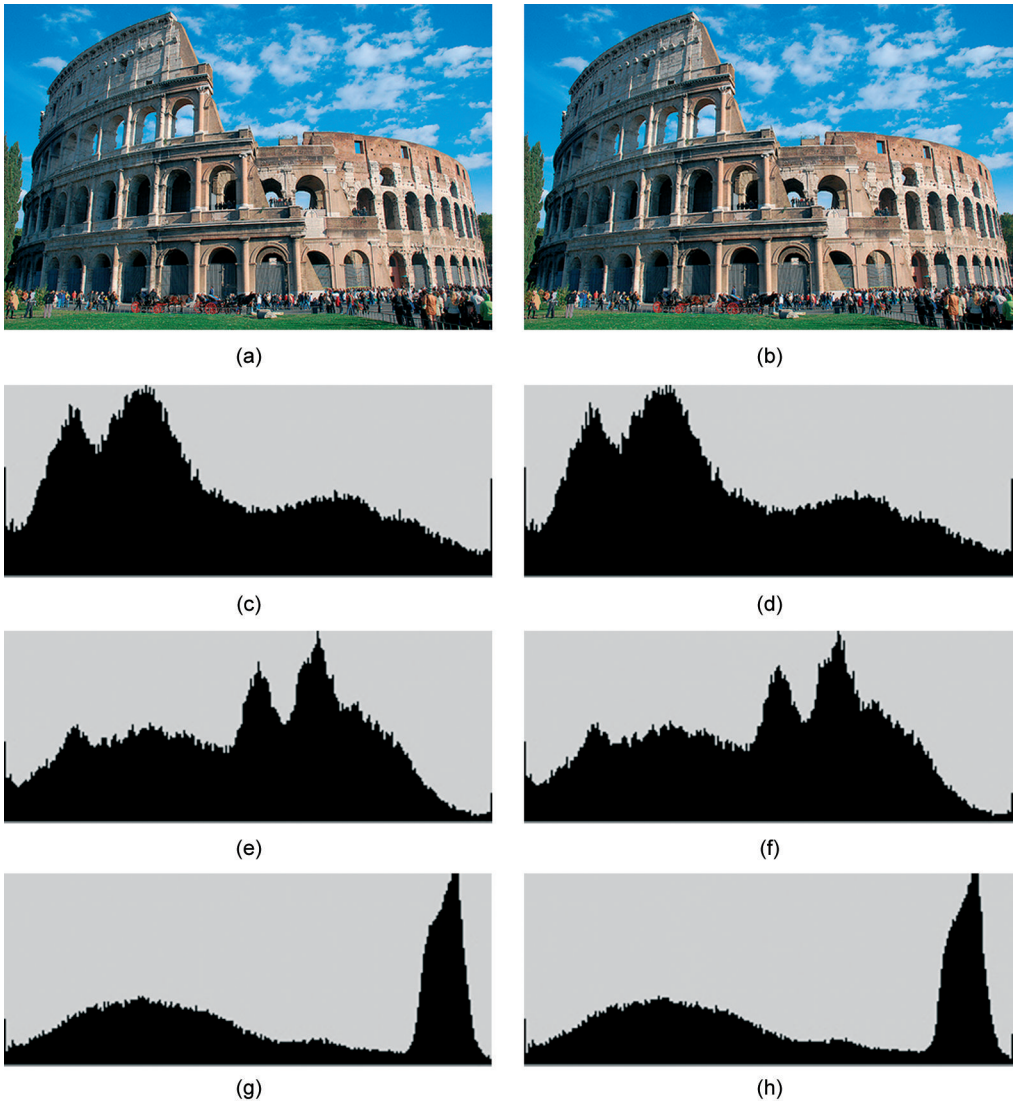


Figure 3.5 Examples of the program: (a) original image, (b) *stego* image, (c) red colour histogram of the original image, (d) red colour histogram of the *stego* image, (e) green colour histogram of the original image, (f) green colour histogram of the *stego* image, (g) blue colour histogram of the original image and (h) blue colour histogram of the *stego* image.

3.6.4 Hide in picture – Blowfish

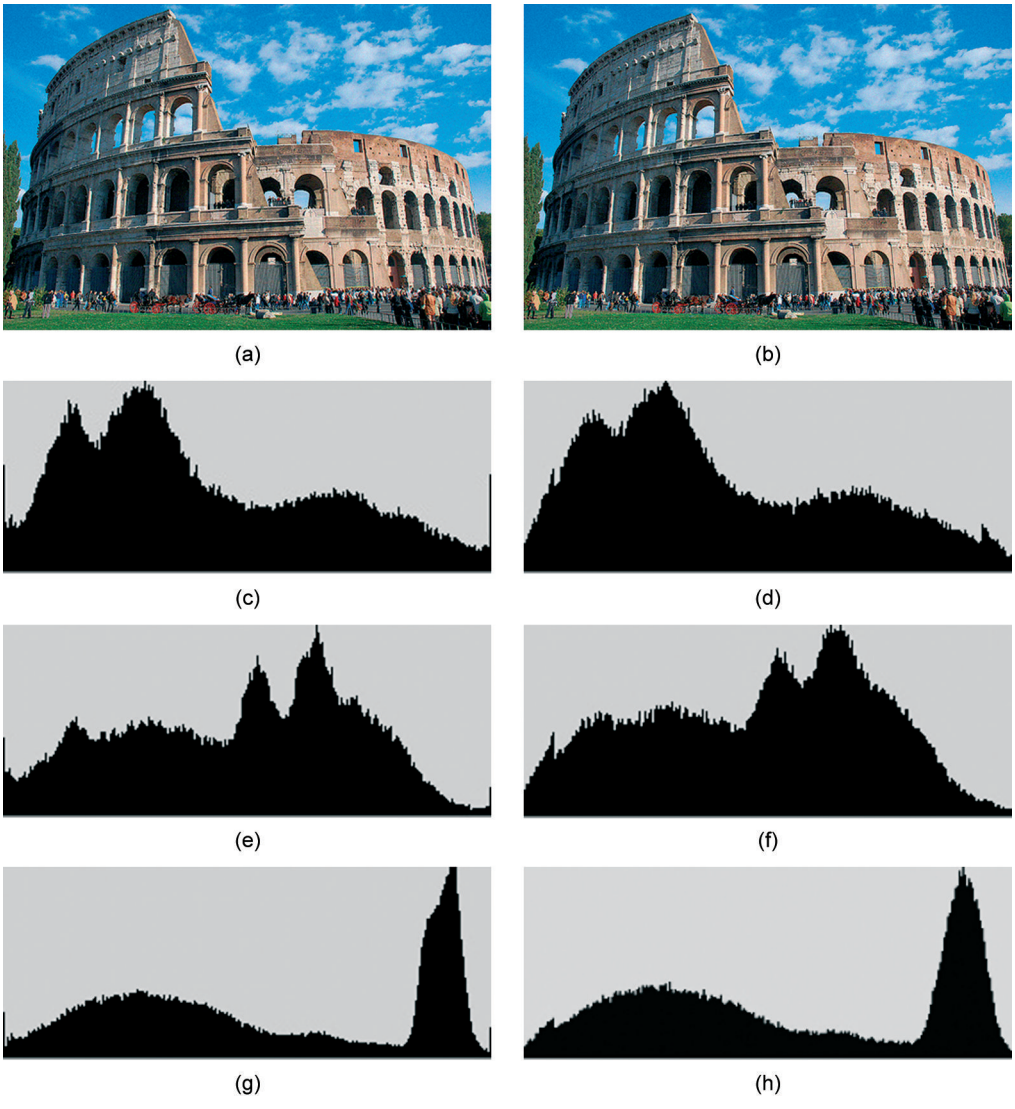


Figure 3.6 Examples of the program: (a) original image, (b) *stego* image, (c) red colour histogram of the original image, (d) red colour histogram of the *stego* image, (e) green colour histogram of the original image, (f) green colour histogram of the *stego* image, (g) blue colour histogram of the original image and (h) blue colour histogram of the *stego* image.

3.6.5 Hide in picture – Rijndael

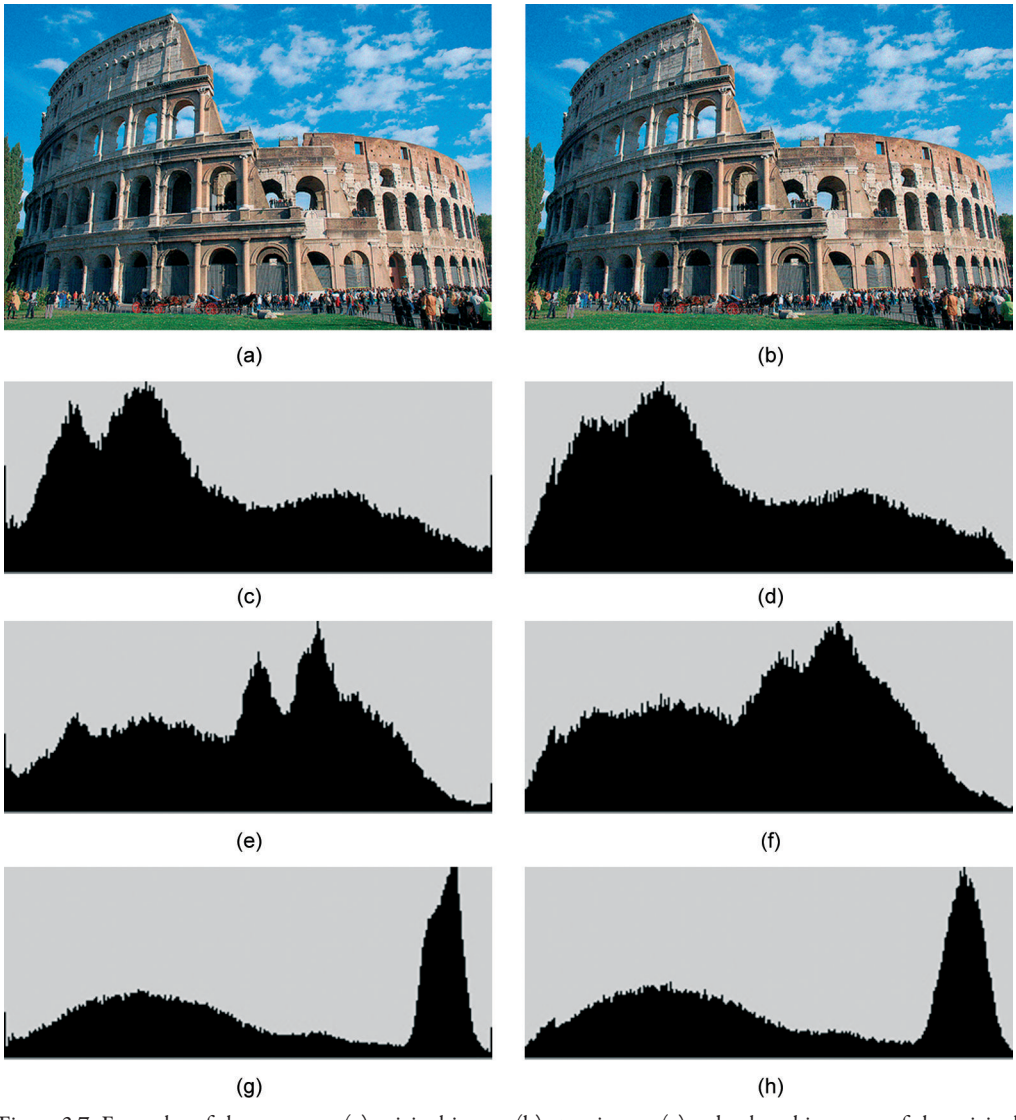


Figure 3.7 Examples of the program: (a) original image, (b) *stego* image, (c) red colour histogram of the original image, (d) red colour histogram of the *stego* image, (e) green colour histogram of the original image, (f) green colour histogram of the *stego* image, (g) blue colour histogram of the original image and (h) blue colour histogram of the *stego* image.

3.6.6 OpenPuff

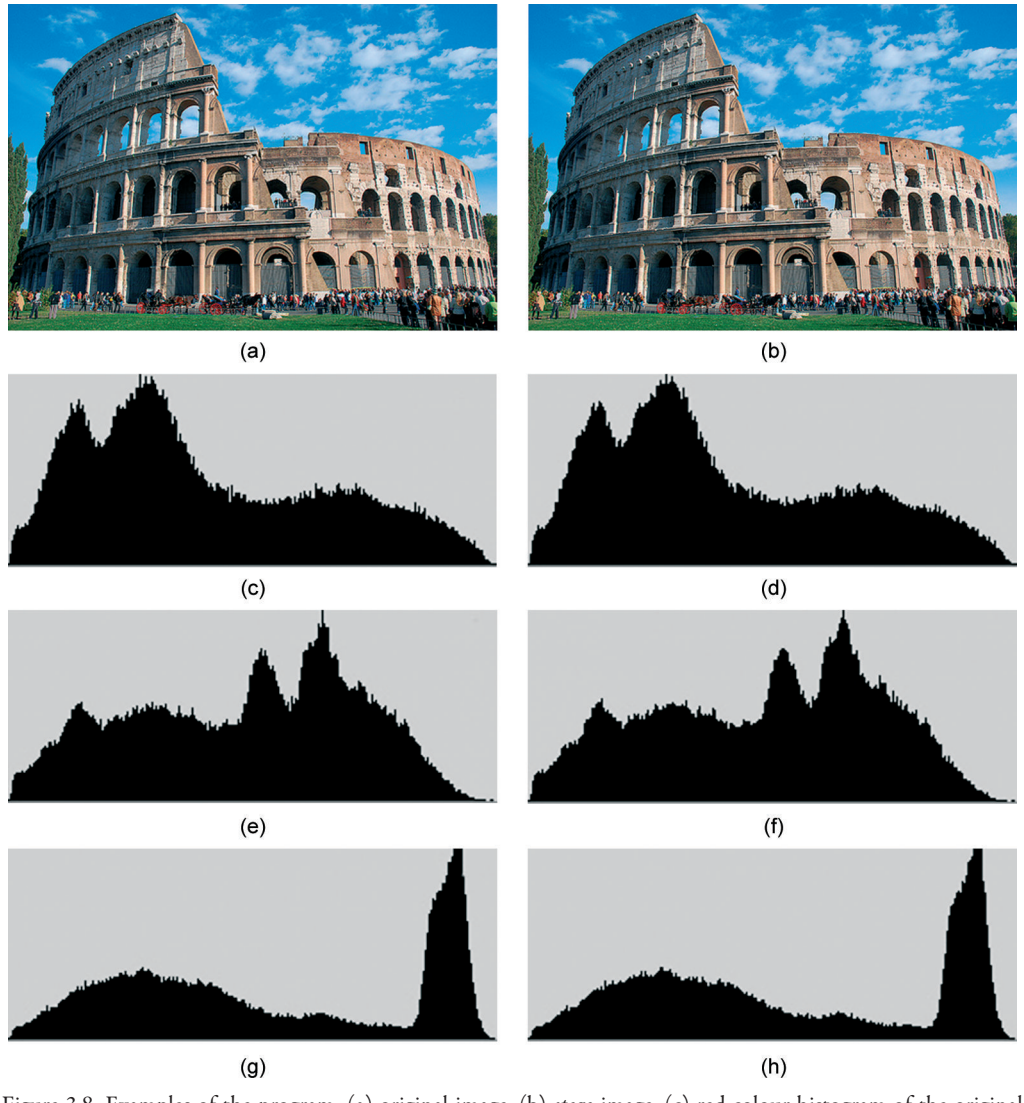


Figure 3.8 Examples of the program: (a) original image, (b) *stego* image, (c) red colour histogram of the original image, (d) red colour histogram of the *stego* image, (e) green colour histogram of the original image, (f) green colour histogram of the *stego* image, (g) blue colour histogram of the original image and (h) blue colour histogram of the *stego* image.

3.6.7 S tools – Data Encryption Standard (DES)

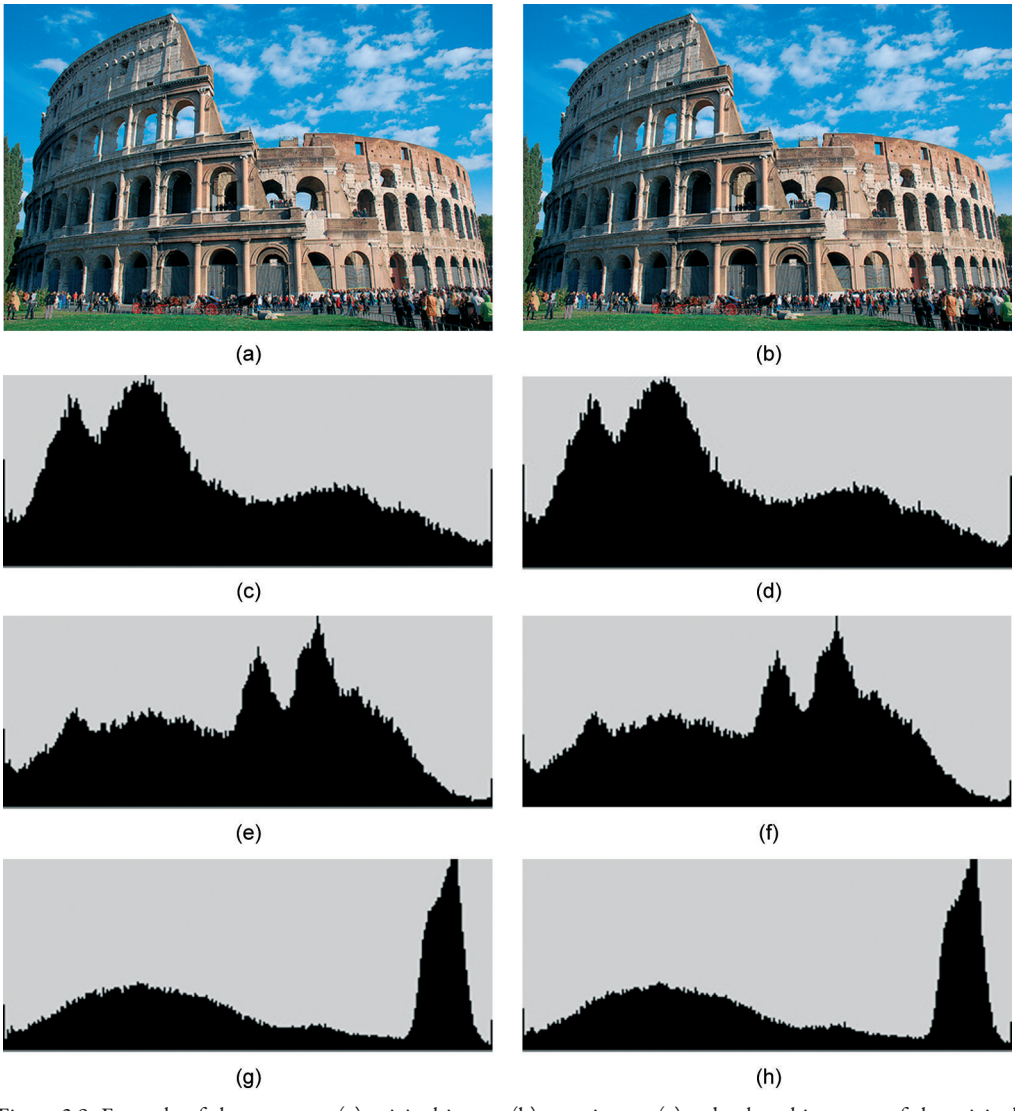


Figure 3.9 Example of the program: (a) original image, (b) *stego* image, (c) red colour histogram of the original image, (d) red colour histogram of the *stego* image, (e) green colour histogram of the original image, (f) green colour histogram of the *stego* image, (g) blue colour histogram of the original image, (h) blue colour histogram of the *stego* image.

3.6.8 S tools – International Data Encryption Algorithm (IDEA)

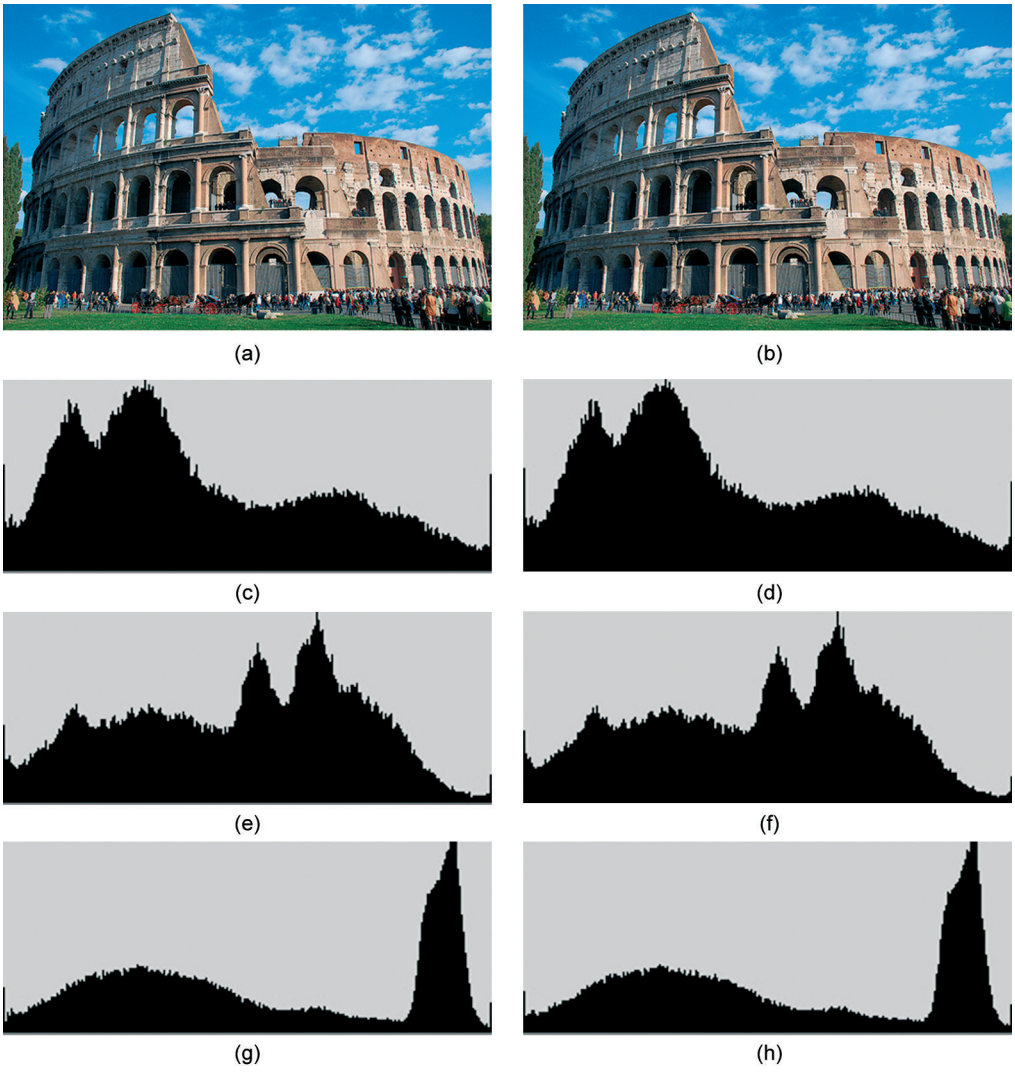


Figure 3.10 Examples of the program: (a) original image, (b) *stego* image, (c) red colour histogram of the original image, (d) red colour histogram of the *stego* image, (e) green colour histogram of the original image, (f) green colour histogram of the *stego* image, (g) blue colour histogram of the original image and (h) blue colour histogram of the *stego* image.

3.6.9 S tools – MDC

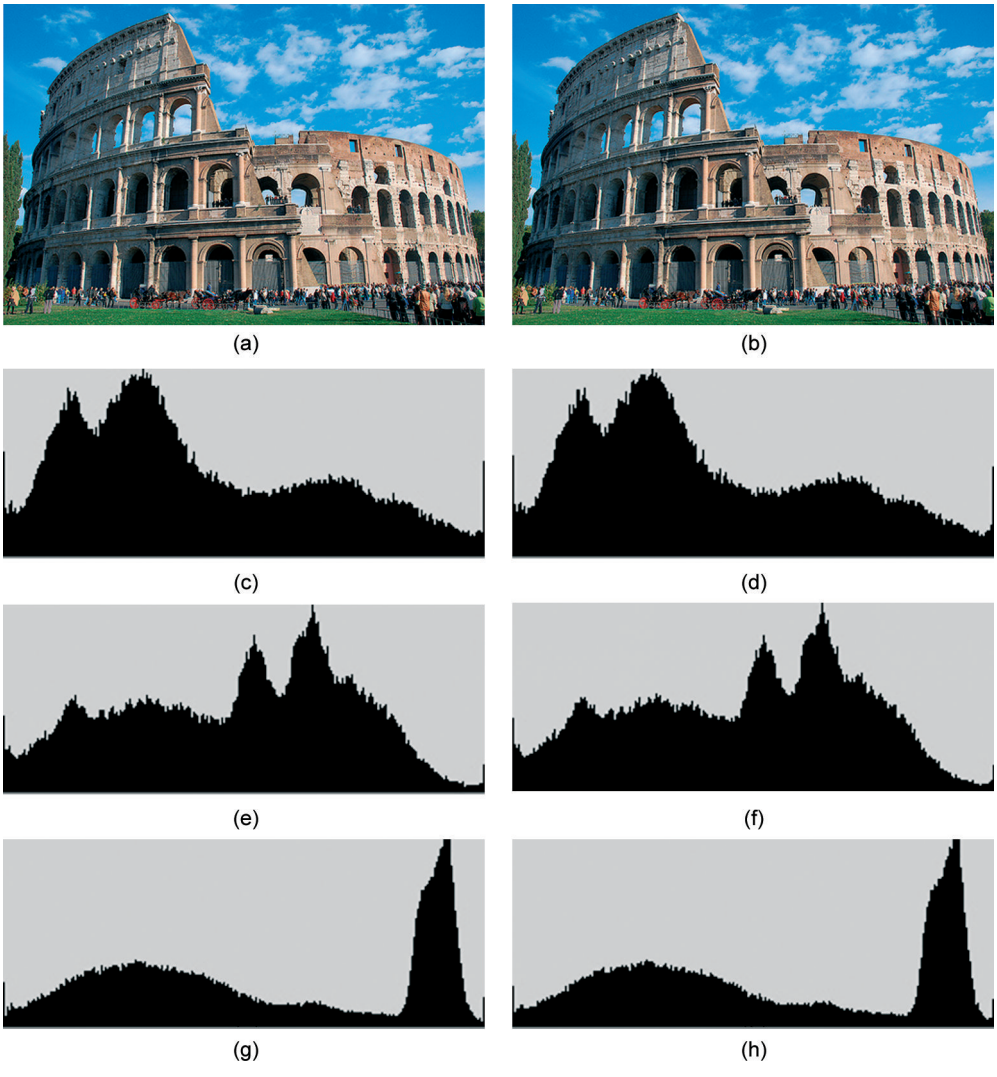


Figure 3.11 Examples of the program: (a) original image, (b) *stego* image, (c) red colour histogram of the original image, (d) red colour histogram of the *stego* image, (e) green colour histogram of the original image, (f) green colour histogram of the *stego* image, (g) blue colour histogram of the original image and (h) blue colour histogram of the *stego* image.

3.6.10 S tools – Triple DES

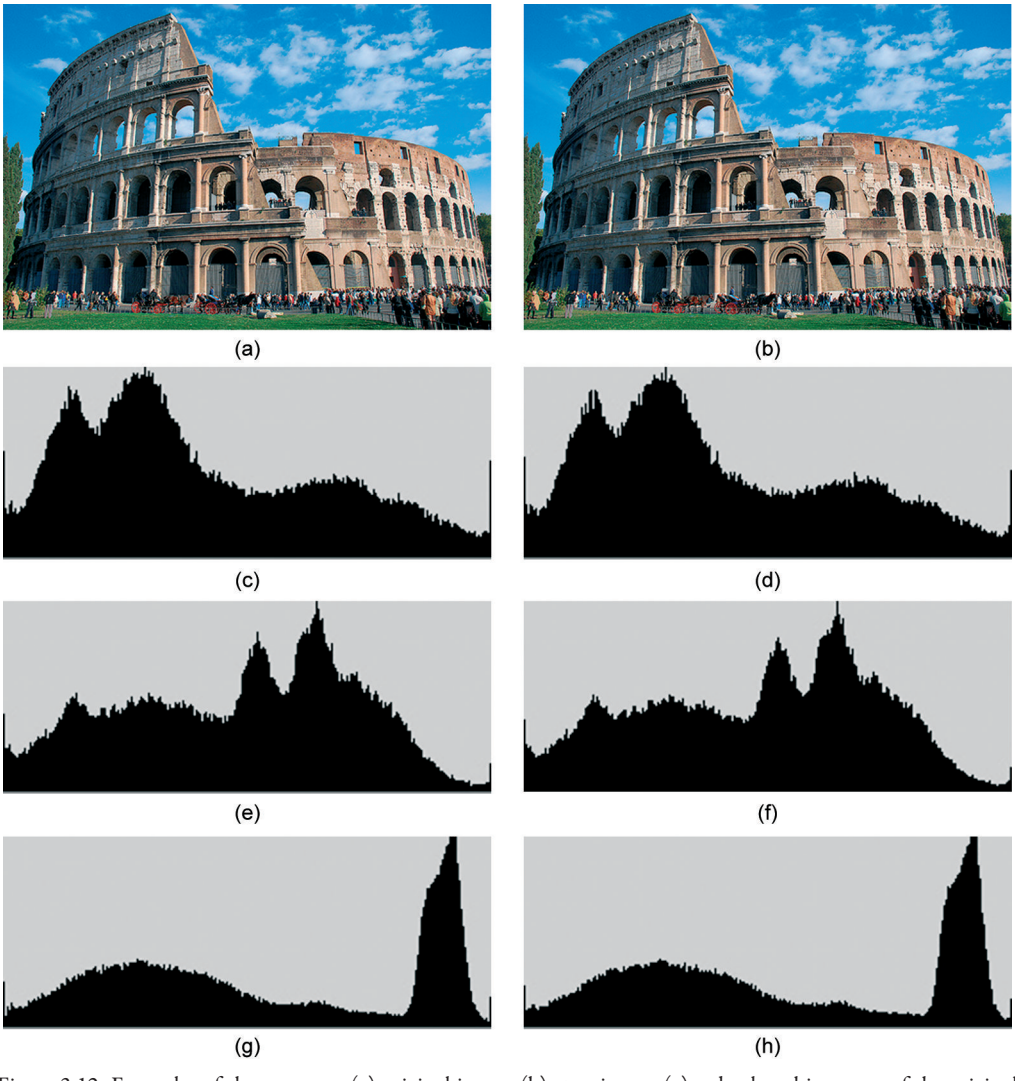


Figure 3.12 Examples of the program: (a) original image, (b) *stego* image, (c) red colour histogram of the original image, (d) red colour histogram of the *stego* image, (e) green colour histogram of the original image, (f) green colour histogram of the *stego* image, (g) blue colour histogram of the original image and (h) blue colour histogram of the *stego* image.

3.6.11 SilentEye

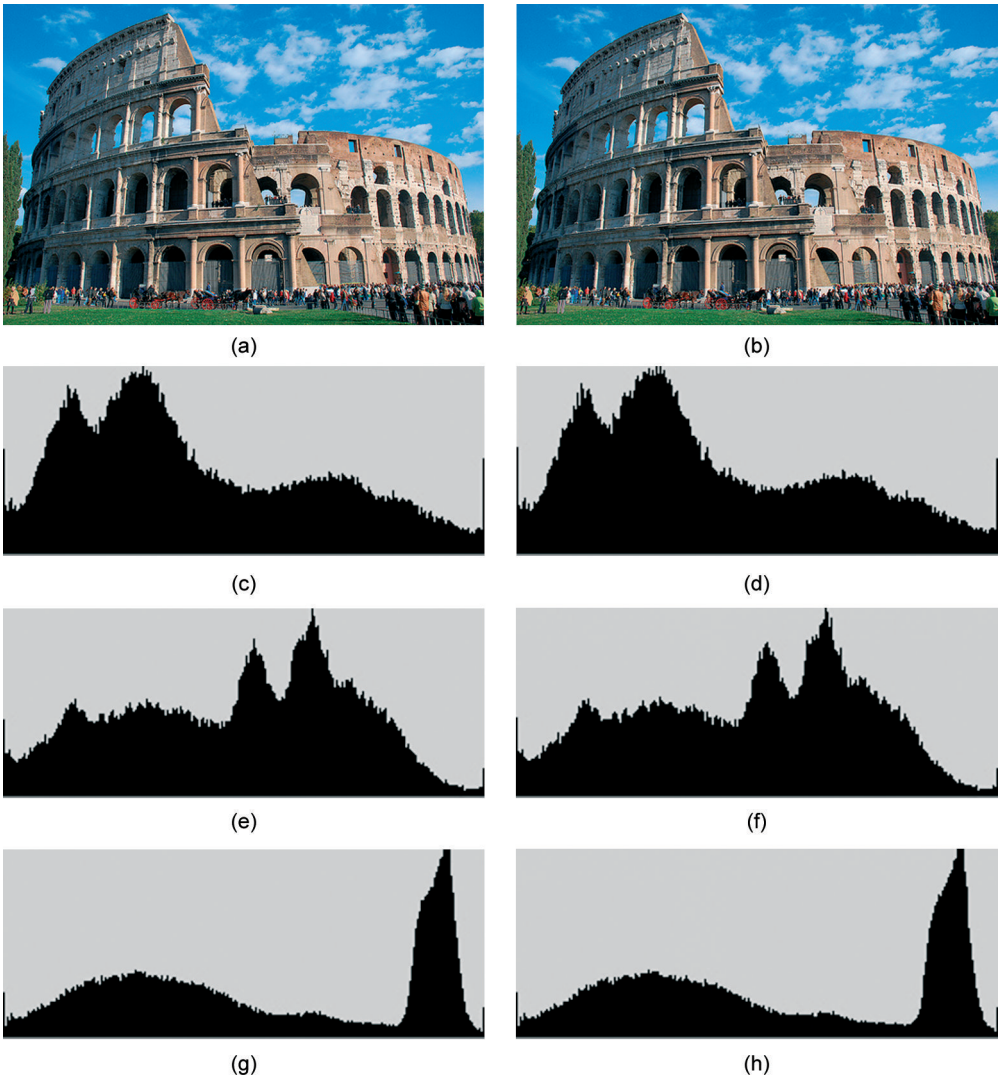


Figure 3.13 Examples of the program: (a) original image, (b) *stego* image, (c) red colour histogram of the original image, (d) red colour histogram of the *stego* image, (e) green colour histogram of the original image, (f) green colour histogram of the *stego* image, (g) blue colour histogram of the original image and (h) blue colour histogram of the *stego* image.

CHAPTER 4

DIGITAL WATERMARKING

4.1 Introduction

Digital watermarking and steganography are the technologies that are used to hide information in an imperceptible manner within appropriate covers. However, there are significant differences between the two technologies. In fact, steganography makes reference to a point-to-point communication between two subjects and the related methods are not, in general, particularly robust in modifying the *cover* data, changes that can be generated from transmission errors and storage, from format conversions, from compressions or from digital/analogue conversions. *Watermarking*, on the contrary, is characterised by a high resilience against any removal attempts. Its purpose is not so much to hide the data, the existence of which the attacker is virtually certain, but to resist removal of the latter. A typical application of *watermarking* is represented by the insertion of a copyright mark within a digital document. Other applications are represented by data monitoring and its tracking.

4.2 History and terminology

Watermarking was first used to mark paper around the year 1300. The oldest manuscript marked was dated 1292 and originates from Fabriano, in the region of Umbria in Italy, which is a city that has witnessed great development of the watermarking of paper. At the end of the 13th century, there were about 40 paper mills competing in the paper market together in Fabriano, producing paper characterised by different prices, quality and format. The paper produced by such mills was extremely crude and inappropriate for writing. For this reason, a subsequent intervention of smoothing by hand was required. Since competition for the production, distribution and sale of paper was very tight, there was a need to identify origin of the same in some way. For this reason, watermarking was conceived that quickly spread throughout Italy and the rest of Europe. Watermarking then became very useful in the printing of banknotes and this term was also used for digital documents from which the term digital watermarking is derived. Digital watermarking has seen a rapid increase in terms of research and development, starting from 1995, the year in which interest in this sector was relatively small. Terminology refers to visible watermark if the digital document contains photos, videos, etc. of figures or logos that are clearly visible and distinguishable and that, therefore, represent technologies that are very similar to the watermarking of paper.

It has already been stated that a characteristic that differentiates watermarking from steganography is represented by strength and even if the attacker knows of the existence of hidden information, it is very difficult to delete the same without knowledge of the watermarking algorithm and the corresponding key used. Usually, in order to achieve a high degree of resilience, the ability to hide a great quantity of information is relaxed. At the beginning of its development, the term embedded signature was used instead of the term of watermarking. Fragile watermarkings are watermarks that typically have a low resilience. They are used to detect changes to the watermarked data rather than hide non-erasable data.

4.3 Basic principles

All methods of watermarking are characterised by the same functional blocks, represented by a watermarking system and a watermark extraction system (Figure 4.1).

The watermark is supplied at the entrance to the system, the cover and the public or optional secret key. The watermark may be of any type such as a number, text or an image. The key may be used to reinforce security, avoiding removal of the watermark. In practice, all systems use at least one key or a combination of different keys. Watermarking systems are called secret or public if they use, respectively, a secret key or a public key. The main properties of watermarking systems are represented by imperceptibility, redundancy and the use of keys.

In terms of imperceptibility, the changes caused by the watermarking process should be less than a predetermined threshold: this means that a very specific policy must be defined and that the changes themselves can be quantified. With regard to redundancy, to ensure a certain level of resilience, the watermarking information is distributed within the document redundantly, allowing recovery of the watermark also using the parts of the document instead of the entire document.

Concerning the use of keys, the watermarking process uses one or more than one of them, similar to encryption, to avoid manipulation or deletion of the watermark. If the watermark can be read by an attacker, it can be easily removed as the same has gained knowledge not only of the watermarking process but also of the position of the watermark itself. The properties just described can be applied to all the watermark schemes and for all types of data such as images, video, audio, etc. The watermark recovery system is shown in Figure 4.2.

In a watermark recovery system, the watermarked document is provided in input, the secret key or public key and, depending on the method, also the original document and/or the original watermark. The W watermark is recovered in output and, in most cases, a parameter that measures the degree of confidence that indicates if a given watermark is present in the document to be checked

There are essentially three types of marking systems and their difference lies in the nature and the combination of inputs and outputs. They are private watermarking, semiprivate watermarking and public watermarking.

Private watermarking, also called non-blind watermarking, requires knowledge of the original document. There are two basic types: type I and type II. In type I, the system extracts the W watermark

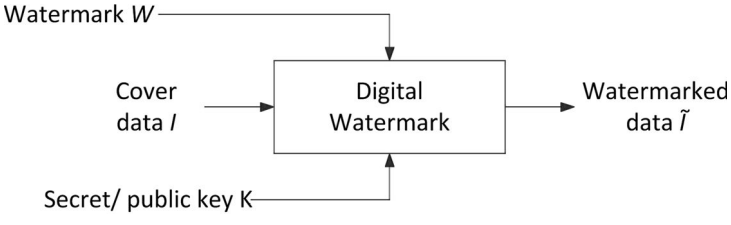


Figure 4.1 Basic diagram of a watermarking system.

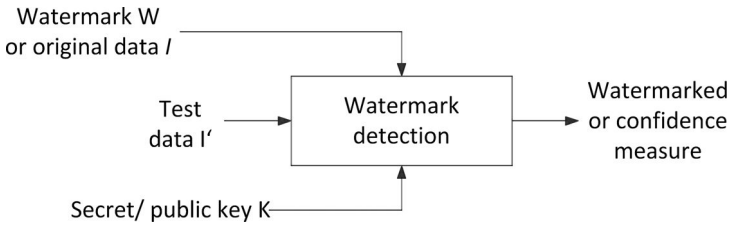


Figure 4.2 Schematic diagram of a watermark recovery system.

from document I, possibly distorted, and uses the original document to find the position of the watermark in I′. In type II, a copy of the watermark is instead requested to perform the extraction: it provides only the positive or negative output information on the presence of the watermark itself. This scheme is very resilient compared to other schemes as a small amount of information and access to restricted material is required.

Semiprivate watermarking, also called semi-blind watermarking, does not need the original document for detection and is able to ascertain if the watermark is present or not. Private and semiprivate watermarking is used to demonstrate the properties of a document in judicial phase, to check for the illegal copying of applications, such as the Digital Versatile Disk (DVD), and to identify original copies from counterfeits.

Public watermarking, also called blind-watermarking, does not require any knowledge of the original document I or knowledge of watermark W. In practice, it extracts a certain amount of bits that are the W watermark from the watermarked document.

The algorithm data input, depending on the applications, can be both uncompressed and compressed and the algorithm itself should be able to work, in the event, directly on compressed data to avoid the performance of lengthy operations of decompression and recompression that would be computationally demanding for the system that has to perform them.

4.4 Applications

As watermarking is used for different purposes, there is no universal technology valid for all applications but there are various technologies that can vary from time to time, also characterised by different strength. The main applications are copyright protection, data tracking, protection from unauthorised copies, image authentication, etc.

Copyright protection is one of the most important applications of *digital watermarking*. The aim is to hide information about the owner to prevent third parties from being able to claim ownership of it. For this reason, this class of applications must be characterised by a high strength. In addition, the system used for this type of applications must not be ambiguous and be able to resolve the possible effects of an addition of a further watermarking on an existing marking and for this reason the most stringent requirements are necessary, simple resilience being somewhat insufficient.

In terms of data tracking, watermarking is very important for controlling the distribution of individual documents and the production of illegal copies. This type of application is also called *fingerprinting* and consists of the insertion of a different watermark in each copy. In this type of application, the mark must be able to be extracted with relative ease. In addition, always in this type of application, a high level of resilience is required as opposed to the standard processing of images and given targeted attacks.

With regard to protection from unauthorised copies, it is a very complex operation to implement in open systems but relatively easy to produce in closed systems. In such systems, it is possible to use watermarking to indicate the status of copies, as, for example, in DVD systems where entering an

appropriate watermark enables only the one copy for an unlimited number of times, without however being able to make copies.

With image authentication, watermarking is used to detect changes to the same. This goal is achieved by using the so-called fragile marking, characterised by reduced strength only in relation to certain specific operations such as compression, remaining unaltered compared to other operations.

4.5 Algorithm requirements

Watermarking algorithms must be characterised by specific requirements depending on the specific application. In general, non-perceptibility of the watermark is required that is independent of the specific application. Other requirements are represented by the extraction or verification of a given watermark, imperceptibility, the strength, recovery of the watermark with or without original document, extraction of the watermark or verification of the presence of a given watermark, the security and use of keys and resolution of property rights.

With regard to imperceptibility, this is one of the most important requirements, together with resilience in *digital watermarking*. Any distortions introduced by a less than perfect watermarking system could also reduce or destroy the commercial value of a product. In this sense, it is very important to use watermarking systems that take into account the human audio-visual system in order to maximise the energy of the mark without introducing perceptible distortions both at an audible and visual level.

Resilience is one of the fundamental requirements together with imperceptibility. Usually, the higher the resilience, the greater is the perceptibility. For this reason, it is important to find the right compromise between these two parameters. Depending on the application, the resilience required influences the structure of the watermarking algorithm. If the distortions to which the watermark is exposed when it is subjected to changes whether intentional or unintentional are taken into consideration, two major groupings can be identified: destruction and synchronisation attacks. In destruction attacks, noise is added to the data while in synchronisation attacks, changes can be made to the spatial or temporal geometry of data. Depending on the application and requirements of the watermarking system, it is possible to draw up a general, but not exhaustive, list of possible attacks:

1. signal processing (increase of the contrast, colour correction, gamma factor correction);
2. additive and multiplicative noise (Gaussian, uniform, speckle, etc.);
3. linear filtering (low pass, high pass, bandpass);
4. non-linear filtering (median filtering, morphological filtering);
5. lossy compression (images: Joint Picture Expert Group (JPEG); video: H.261, H.263, H.264, Moving Picture Expert Group (MPEG)-2, MPEG-4; audio: MPEG-2 audio, MP3, MPEG-4 audio, G.723);
6. affine local and global transformations (translation, rotation, scaling, etc.);
7. data reduction (cutting and modification of histogram);
8. data composition (logo insertion, scene composition);
9. transcoding (from H. 263 to MPEG-2, from Graphics Interchange Format (GIF) to JPEG);
10. Digital/Analog (D/A) and Analog/Digital (A/D) conversion (prints scanning and analogue TV broadcast);
11. multiple watermarking;
12. collusion attacks;
13. statistical average;
14. mosaic attacks.

Some types of attack are described below. The basic principle of marking is that the system is sufficiently robust and such that a successful attack also destroys the commercial value of the watermarked document.

With regard to watermark recovery, with or without original document, it can be said with some certainty that watermarking methods that use original data in the process of recovery generally have a greater strength not only against distortions introduced by noise but also against distortions in the geometry of the data as they allow the revelation and reversal of geometrical distortions. In many applications, such as monitoring and tracking, access to the original data is not possible. In other applications, such as video watermarking, it can be difficult to make use of the original data given the high amount of data to be processed.

In terms of watermark extraction or verification of the presence of a given watermark, it has already been stated that there are two types of systems: those that introduce a specific piece of information and then ascertain its existence and those that enter arbitrary information within the document to be watermarked. These latter systems are mainly used for the tracking of images on the Internet. It should be emphasised that both schematics are interchangeable. A scheme that allows verification of a watermark may be considered as a 1 bit watermark recovery scheme and such a scheme can be extended to any number of bits by modulation of the hidden arbitrary information.

With regard to the security and use of keys, it should be remembered that in many applications, such as copyright protection, secrecy of the hidden information must be ensured. This requirement is normally called watermark security. If secrecy is required, a secret key must be used for the process of insertion and extraction of the watermark. Two levels of security can be identified. At the top-most level, an unauthorised user cannot read or decode the hidden watermark and cannot detect whether a given document contains it. At the lowest level, every user can identify whether the document is watermarked but the watermark cannot be read without possessing the reading key. These schemes can contain multiple watermarks with public and private keys, and can mix one or more public keys with a private key, and can hide public and private watermarks.

With regard to the resolution of property rights, it must be possible to determine the subject that initially watermarked the document in the case where there are multiple watermarks. This objective can be achieved by implementing properties such as non-invertibility or stamping.

4.6 Evaluation of systems

The evaluation of watermarking systems is a very important element, concerning both the strength and the level of imperceptibility achieved. It has already been stated that there is a compromise between the resilience of watermarking and relative perceptibility and in order to make valid comparisons, certain objective evaluation parameters must be introduced. The basic evaluation parameters are represented by the amount of information that can be inserted as a watermark, watermark strength inserted, size and nature of the data and confidentiality of the information.

With regard to the amount of information that can be inserted as a watermark, this is an important parameter as it influences the resilience of the watermark. The greater the amount of information to be inserted, the smaller, in general, is the resilience of the watermark.

In terms of the strength of the watermark entered, it has already been stated that there is a compromise between this and perceptibility. To increase the resilience of the watermark, a greater force of watermarking is required that inevitably increases the perceptibility of the watermark. In relation to the size and nature of the data, these are parameters that directly influence the resilience.

With regard to confidentiality of the information, although the level of such does not affect the perceptibility and resilience of the watermark, the same is very important with regard to security. The space of the keys must be large enough to make a brute-force attack practically impossible, that is an exhaustive

search of all the possible keys. These systems often resist sophisticated attacks but offer poor defence against trivial attacks as often creators of the same have not taken into account the basics of cryptography.

Taking into account the parameters outlined previously, valid comparisons can be performed, provided the same input parameters are used such as the amount of information that makes up the watermark.

Since perceptibility represents a parameter of fundamental importance, it is necessary to evaluate the same through subjective testing or through quality metrics. When applying subjective testing, a well-determined protocol must be followed that accurately describes procedures for testing and evaluation. It is usually divided into two phases: in the first phase, the set of distorted data is ordered from best to worst while in the second step the subject in question is asked to quote all the data, also describing the degree of perception of the alteration. Evaluation can be performed using the International Telecommunication Union – Radiocommunications Sector (ITU-R) Rec 500 recommendations that classify the data into level 5, imperceptible and excellent quality; level 4, noticeable but not annoying and good quality; level 3, slightly annoying and normal quality; level 2, annoying and low quality; level 1, very annoying and poor quality. Subjective tests are very practical for final evaluation of the quality but are not very useful for the purposes of research and development.

For this reason, quantitative metrics for evaluation of the distortion are introduced that allow a comparison between the different methods, as the results do not depend on subjective evaluations.

Given $p(x, y)$ the value of the pixel in the position (x, y) of the original image, $p^W(x, y)$ the value of the pixel in the position (x, y) of the watermarked image, X the number of pixels of the lines and Y the number of columns, a set of quantities can be defined. The first group of quantities belongs to what is called metric difference of distortions.

The first quantity is represented by the average absolute difference (AD), defined as:

$$\frac{1}{XY}\sum_{x,y}\left|p(x,\ y) - p^{w}\left(x,\ y\right)\right| \tag{4.1}$$

The second quantity is represented by the mean squared error (MSE), defined as:

$$\frac{1}{XY}\sum_{x,y}[p(x,\ y) - p^{w}(x,\ y)]^2 \tag{4.2}$$

The third quantity is represented by the standard L^P (L^P norm), defined as:

$$\left[\frac{1}{XY}\sum_{x,y}\left|p(x,\ y) - p^{w}\left(x,\ y\right)\right|^{p}\right]^{1/p} \tag{4.3}$$

The fourth quantity is represented by the Laplacian MSE (LMSE), defined as:

$$\sum_{x,y}[\nabla^2 p(x,\ y) - \nabla^2 p^{w}(x,\ y)]^{2/\Sigma_{x,y}}[\nabla^2 p(x,\ y)]^2 \tag{4.4}$$

The fifth quantity is represented by the signal-to-noise ratio (SNR), defined as:

$$\sum_{x,y}[p(x,\ y)]^{2/\Sigma_{x,y}}[p(x,\ y)] - p^{w}(x,\ y)]^2 \tag{4.5}$$

The sixth quantity is represented by the peak SNR (PSNR), defined as:

$$XY\ \max_{x,y}[p(x,\ y)]^{2/\Sigma_{x,y}}[p(x,\ y)] - p^{w}(x,\ y)]^2 \tag{4.6}$$

The first group of quantities belongs to what is called metric correlation of distortions.

The first quantity is represented by normalised cross-correlation (NC), defined as:

$$\sum_{x,y} p(x,\ y)p^{w}(x,\ y)/\sum_{x,y}[p(x,\ y)]^2 \tag{4.7}$$

The second quantity is represented by correlation quality (CQ), defined as:

$$\sum_{x,y} p(x, y)p^{w}(x, y)/\sum_{x,y} p(x, y) \tag{4.8}$$

A final quantity that belongs to a group apart is represented by histogram similarity, defined as:

$$\sum_{n=0}^{255} \left| f_I(n) - f_I w(n) \right| \tag{4.9}$$

where $f_I(n)$ represents the relative frequency of the n level in a 256 level image.

Having explained the objective evaluation parameters, it is now possible to illustrate the evaluation of resilience from a visual point of view. Table 4.1 provides a list of graphs useful in this sense, together with the fixed and variable parameters useful for performing objective comparisons.

For the purposes of assessments, the test should be performed several times using data and different keys that should be followed by averages. Resilience is usually measured through the *bit error rate,* defined as the ratio of the incorrect bits extracted and the total number of hidden bits or through the *detection error* defined as one minus the *bit error rate* raised to the power of the number of bits. The visual quality term relates to all those metrics suitable for evaluation of visual distortion due to the watermarking process.

The strength graph depending on the force of attack is one of the most important charts. It shows the bit-error or the *detection error* depending on the force of attack for a given visual quality. The strength graph depending on the visual quality shows the relationship between the *bit error* or the *detection error* and the visual quality for a determined attack. It may be useful to determine the minimal visual quality for a desired bit error rate under a determined attack.

The attack graph depending on the visual quality shows the ratio between the maximum allowable attack depending on the visual quality and a certain strength. This graph allows an immediate evaluation of the watermark attack permitted depending on the visual quality. It is also very useful in the comparison of different watermarking techniques.

Given an image, a watermark detector must substantially perform two tasks: to decide whether the image is watermarked and to extract the information contained within it. In the first task, the detector must decide between an alternative hypothesis, that is the watermark is present, and a null hypothesis, that is the watermark is not present. In the case of binary test, two types of error can occur: the alternative hypothesis is accepted when, in reality, the null hypothesis is true, and the null hypothesis is accepted when, in reality, the alternative hypothesis is true. The first type of error is referred to as type I error or even false positive, while the second type of error is referred to as type II error or even false negative. In this sense, the so-called receiver operating characteristic (ROC) graph is very useful. Usually, in test hypothesis, a statistical test is compared depending on a given threshold to decide on one or another hypothesis. The comparison of various watermarking schemes using the same threshold for all can generate considerable errors. The ROC graph prevents this because it compares the test using different thresholds and shows the ratio between the true positive fraction (TPF) on the y-axis

Table 4.1 Different types of charts, together with the fixed parameters and variables useful for objective comparisons.

Type of graph	Parameter			
	Visual quality	Resilience	Attack	Bit
Resilience according to the attack	Fixed	Variable	Variable	Fixed
Resilience according to the visual quality	Variable	Variable	Fixed	Fixed
Attack according to the visual quality	Variable	Fixed	Variable	Fixed
Receiver operating characteristic (ROC)	Fixed	Fixed	Fixed/variable	Fixed

and the false positive fraction (FPF) on the *x*-axis. The TPF is defined as:

$$\mathrm{TPF} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4.10}$$

where TP is the number of results of the true positive test and FN is the number of results of the false negative test. The FPF is defined as:

$$\mathrm{FPF} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \tag{4.11}$$

where FP is the number of results of the false positive test and TN is the number of results of the true negative test. In practice, the ROC graph shows the TPF-FPF pairs for variable thresholds with continuity. An optimal detector has a curve that starts from the bottom left-hand corner and proceeds towards the top left-hand corner and then towards the upper right-hand corner. The diagonal line from the bottom left-hand corner to the upper right-hand corner describes a detector that chooses one or the other hypothesis with the same probability. It can thus be concluded that a detector characterised by a high degree of accuracy is very close to the top left-hand corner and the greater its accuracy, the lower its distance from that angle. The integral is often used below this curve as a parameter of measurement of the success of the detector performance. To generate such graphs, the same number of marked and unmarked images must be used. If the overall performance of a watermarking method must be assessed, the test should include a high variety of attacks by continuously varying the parameters.

4.7 Watermark removal algorithms

There are many algorithms available for free that have been developed to continuously remove watermarks in an unauthorised manner. Depending on the type of document, there is dedicated software. Images, for the most part, introduce reduced distortion in the quality of the image in order to eliminate the watermark itself.

4.8 Future evolution and standardisation

Interest in electronic watermarking is very high, both from industry and the scientific world, demonstrated, with regard to the former, by the large number of companies in the industry that have been founded in recent years while in relation to the second, by the large number of scientific articles produced.

In addition to industrial and scientific activities, there are several international projects of the European Community in the field, such as TALISMAN, OCTALIS and OKAPI. The consortia of international standardisation are also interested in the techniques of digital watermarking, with particular reference to video and DVD standards, in order to avoid the generation and distribution of illegal copies. In spite of the high level of research in the field, electronic watermarking is not yet a mature technology and many issues in the field still remain open. In addition, this sector is characterised, in most cases, by a low level of theoretical analysis, since the algorithms generated, for the most part, are heuristics-based. A weak point is represented by the great difficulty encountered in comparing the various algorithms on an objective basis. In addition, it should always be remembered that there is a compromise between strength and amount of information that can be inserted into the document to watermark it. In many cases, algorithms are resistant to attacks by non-experts but succumb, inexorably, under the attacks of expert subjects.

With regard to judicial investigations, electronic watermarking is not always accepted as evidence demonstrating the intellectual property of a document. For all the reasons given above, even if various algorithms of electronic watermarking are currently used, it is not yet possible to say that this technology will be consolidated with success.

4.9 Watermarking technologies

Although electronic watermarking technologies involve most formats of electronic documents, most scientific articles of the sector are aimed at images. There are three fundamental aspects that must be taken into consideration: the ratio between the information contained in the watermark and the information contained in the host signal, the degradation of the image due to watermarking and resilience.

With regard to the ratio between the information contained in the watermark and the information contained in the host signal, this parameter depends on the nature of the message that must be hidden and the nature of the host signal. Regarding degradation of the image due to watermarking, this is a parameter of vital importance for evaluation of the success of algorithms. In terms of strength, algorithms must be able to withstand any type of attack and to cede only in the event of attacks that would result in the watermarked document no longer being usable.

Below are illustrated a series of technologies based on the following points:

1. Selection of pixels or blocks where watermarking information can be hidden.
2. Selection of the working space for the operation of concealment of the watermark such as the spatial domain or the transformed domain such as discrete cosine transform (DCT), Mellin–Fourier or Wavelet.
3. Formatting strategies of the watermarking signal to be hidden in the host signal.
4. Method of fusing the message in the document to be watermarked.
5. Optimisation of the watermark detector.

4.9.1 Selection of pixels or blocks

A direct application of Kerckhoffs' principle to electronic watermarking provides confirmation that the watermarking algorithm should be public while the watermark should not be accessible. This can be done through a careful choice of the point at which to insert the watermark. A pseudo-random number generator is used in many applications that are initialised via a secret key, and this generator provides the position of the locations where the watermark should be inserted. The secret key is known only to the owner of the document and to the counterpart.

An example of such a system is represented by the patchwork algorithm that not so much permits a watermark to be hidden but allows checking of whether a particular person knows the key. In this algorithm, a secret key is used to initialise a pseudo-random number generator that provides at its output the location of the document where the watermark is to be hidden. The basic version of this algorithm operates in the manner shown below. During insertion, the subject selects n pairs of pixels selected in a pseudo-random manner according to a secret key k. He/she changes the values of luminance (L_{1i} and L_{2i}) of n pairs of pixels using the following formula:

$$L_{1i}^{*} = L_{1i} + 1 \tag{4.12a}$$

$$L_{2i}^{*} = L_{2i} - 1 \tag{4.12b}$$

In practice, the user adds the value 1 to all the L_{1i} and subtracts 1 from all the L_{2i}. In the process of extraction, the n pairs of pixels used at the stage of coding are retrieved by using the secret key k and the

following sum S is calculated:

$$S = \sum_{i=1}^{n} L_{1i}^{*} - L_{2i}^{*} \tag{4.13}$$

If the document contains a watermark, this sum is $2n$ otherwise it should be approximately equal to zero. Extraction is based on the statistical assumption that:

$$E[S] = \sum_{i=1}^{n} E\left[L_{1i}^{*}\right] - E\left[L_{2i}^{*}\right] = 0] = 0 \tag{4.14}$$

if numerous pairs of pixels of an image are randomly chosen and it is assumed that they are independent and identically distributed. As a result, only the user who knows the position of the pixels that have changed can obtain a value $S \approx 2n$.

Secret key based watermarking algorithms are characterised by the disadvantage of not allowing the public recovery of the watermark. To avoid this limitation, public key algorithms have been proposed that use two keys: a private key and a public key. In this sense, the private key is used to watermark the document while the public key is used to verify the existence of the watermark. To do this, for example, spread spectrum technology can be used. In this sense, use is made of a sequence S both for expansion and compression. In any case, thanks to the strength of the coding, it is possible to reconstruct the original signal without knowledge of the entire expansion sequence.

4.9.2 Work selection space

Changing the working space, it is possible to hide the watermark, appropriately changing domain.

The simplest case is that of the discrete Fourier transform (DFT) that offers the possibility of checking the frequencies of the host signal. In this sense, it is very useful to select the more appropriate parts of the image in such a way as to obtain a compromise between strength and visibility.

Given a two-dimensional (2D) image $f(x, y)$, of size $N \times M$, the DFT is defined as:

$$F(k_x, k_y) = \sqrt{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} f(n, m) \exp\left(-i2\pi n k_x / N - i2\pi m k_y / M\right) \tag{4.15}$$

being $i = \sqrt{-1}$. The inverse DFT (IDFT) is defined as:

$$f(n, m) = \sqrt{NM} \sum_{k_x=0}^{N-1} \sum_{k_y=0}^{N-1} F(k_x, k_y) \exp\left(i2\pi n k_x / N + i2\pi m k_y / M\right) \tag{4.16}$$

The DFT is very useful for the purposes of watermarking because it makes it possible to perform phase modulation between the watermark and the document. It is also greatly applied in its derived forms represented by DCT and Mellin–Fourier transform.

With regard to discrete cosine transform, its use in JPEG and MPEG encoding has already been addressed. It is very useful as the watermarking performed in the DCT domain is, in general, very resilient with respect to JPEG or MPEG compression. Furthermore, it allows direct intervention in the compressed domain in order to reduce processing time. There are many methods for operating in this way. The first consists of the sum of the coefficients of the image with those of the watermark. A second method would be to impose the ratios between multiple coefficients of the DCTs depending on the value of the watermark bits. A third method is to vary quantisation of the coefficients depending on the watermark bits. However, there are many other methods that are not illustrated for reasons of space.

Most watermarking algorithms present problems in extraction of the watermark when an affine transformation is applied to the watermarked object. To avoid this, it is possible to use the Mellin–

Fourier transformation. It is based on the following properties of translation:

$$f(n+a, m+b) \leftrightarrow F(k_x, k_y) \exp(-iak_x - ibk_y) \tag{4.17}$$

It is possible to verify how the phase is exclusively altered by a translation. This implies that if the watermark is inserted in the amplitude of the transformed domain, the result is independent of any translations of the original image. To make the watermarking system insensitive to rotations and magnification, log-polar mapping (LPM) can be considered and defined as:

$$(x, y) \leftrightarrow \begin{cases} x = \exp\rho \cos\theta \\ y = \exp\rho \sin\theta \end{cases} \tag{4.18}$$

where ρ being real and θ variable between 0 and 2π. It is clear that the rotation of each element (x,y) in the Cartesian coordinate system results in a translation in the logarithmic coordinate system. In a similar manner, a magnification in the Cartesian coordinate system results in a translation in the polar coordinate system. Using a suitable modification of the coordinate system, both the rotation and magnification can be traced back to a translation. The invariance property of the translation can be used to reconstruct a space that is insensitive to every type of rotation and magnification that is performed on a watermarked image.

Another domain transformation technology is represented by wavelet transform. Wavelets are used as a base for representation of the image called JPEG 2000. Wavelets can be used profitably for the watermarking of images. Because of multi-resolution, they allow an excellent level of distribution of the message in the document in terms of resilience against visibility. The wavelet transform enables space-frequency multi-scale decomposition of the image. Figure 4.3 shows the schematic decomposition of an image with three scale factors while Figure 4.4 shows the decomposition of an image with two scale factors.

The lowest frequency band to the lowest scale factor is shown in the upper left-hand corner (LL_3). At the same level of resolution, the HL_3 block contains information on higher horizontal frequencies and lower vertical frequencies. In the same way, the LH_3 block contains information on lower horizontal frequencies and higher vertical frequencies. The same decomposition is repeated for the intermediate level and the level of higher resolution. A possible way to obtain these different levels of decomposition consists of putting two channel filter banks into cascade, as shown in Figure 4.5.

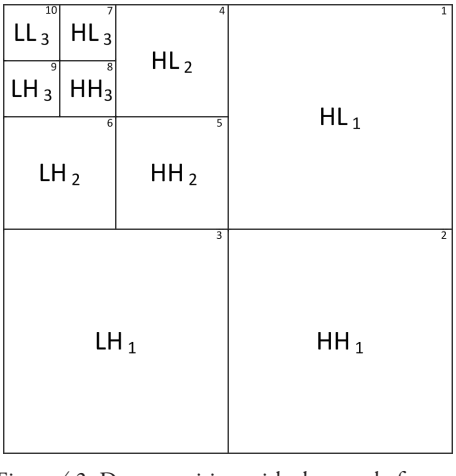


Figure 4.3 Decomposition with three scale factors.



Figure 4.4 Decomposition of an image with two scale factors.

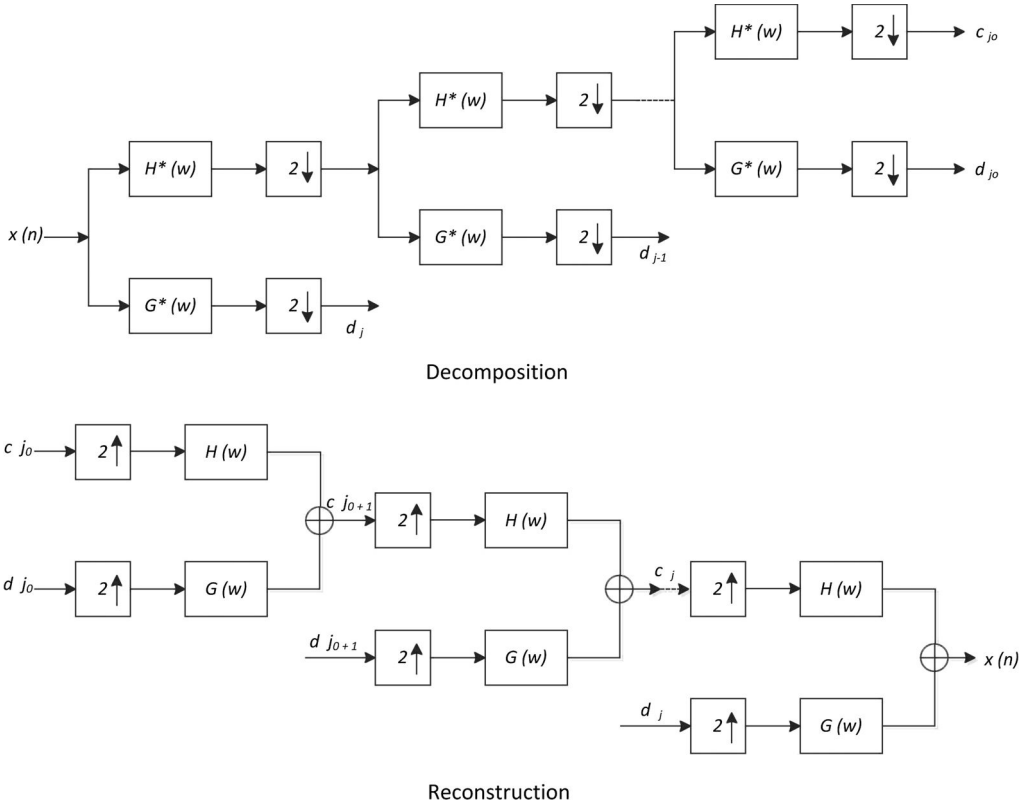


Figure 4.5 Diagram of wavelet transformation.

The two channel filter banks should be orthogonal and are defined by the following relationship:

$$H(\omega) = \sum_k h_k \exp(-jk\omega) \tag{4.19a}$$

$$G(\omega) = \sum_k g_k \exp(-jk\omega) \tag{4.19b}$$

Equation (4.19a) applies to the high-pass filter and (4.19b) applies to the low-pass filter.

The iterative process of the decomposition is given by:

$$c_{j-1,k} = \sum_n h_{n-2k} c_{j,n} \tag{4.20a}$$

$$d_{j-1,k} = \sum_n g_{n-2k} c_{j,n} \tag{4.20b}$$

and the iterative process of reconstruction is defined by:

$$c_{j,n} = \sum_k h_{n-2k} c_{j-1,k} + \sum_k g_{n-2k} d_{j-1,k} \tag{4.21}$$

There are many ways to use wavelet transform for digital watermarking and a few of the many are illustrated in the following. One method consists of the search for significant coefficients that do not change much as a result of repeated operations of image processing, and that, if changed excessively, result in loss of quality of the original image. Another method consists of the addition of wavelet coefficients of the watermark to those of the image at various levels of resolution. Before being added, watermark coefficients are modulated using a human visual model.

A further technique that operates in the transformed domain is represented by the division into perceptive bands taking advantage of the selective masking that, for example, our ear performs on adjacent frequencies. Due to these properties, it is possible to enter the watermark where there is a masking, decreasing perceptibility of the same. In the same way, the eye divides the stimuli into different components that depend on three basic parameters: its position in the visual field, spatial frequency (calculated as the amplitude of the Fourier transform) and orientation (calculated as phase of the Fourier transform). These components are transmitted from the eye to the cerebral cortex through different channels. The masking effect occurs when a component of a channel becomes invisible because of the high energy of the adjacent channel. In this sense, it is possible to split the image into different channels, for example by using Gabor filters, and calculate the energy of the various channels. Finally, a contrast function is calculated that depends on the frequency, the direction and position of the channels. This feature allows us to control the channels that are masked, it being possible to insert the watermark that is not perceived within the same watermark.

4.9.3 Formatting of the watermarking signal

In many cases, it is necessary to format the watermark bits before the insertion process. Some technologies allow direct insertion of the watermark bit strings while other technologies require a prior transformation before insertion.

One piece of technology that is widely used is spread spectrum, in its variations of direct sequence and frequency hopping, already illustrated in detail above. In this sense, the watermark signal is considered as a narrow-band signal with respect to the document band to be watermarked. Spread spectrum technology allows spreading of the bandwidth of the watermark signal to bring it into line with that of the document to be watermarked. High frequencies are also important for invisibility of the watermark but are deleterious with regard to strength. Low frequencies, on the contrary, are deleterious with regard to visibility but superior from a strength perspective. In this sense, spread

spectrum can be extremely useful as it allows the option of hiding the low-frequency signal of the watermark in any one of the desired frequencies. Spread spectrum technologies, moreover, allow protection of the watermark using a secret key to control a pseudo-random noise generator. From the receiving side, the original signal is reconstructed by demodulating the signal received via the pseudo-random signal. The original signal is reconstructed without error, even if some of the frequencies are disturbed during transmission due to the fact that the original information is distributed over various frequencies. To demodulate the original signal correctly, perfect synchronisation between the received signal and the pseudo-random signal is required. In particular, in the case of digital watermarking, it is possible to check synchronisation problems following attacks on visual images.

The advantages and disadvantages of having a low-frequency watermark have already been addressed. To create a low-frequency watermark, it is possible to operate in the manner shown below. Given an initial watermark, a reduced version of this is created on which a Fourier transform is performed. To this transform (2D) are added zeros up to the size of the original watermark. At this point, the inverse Fourier is carried out obtaining a filtered version, by means of a low-pass filter, of the original watermark, within which there are only low frequencies.

4.9.4 Fusion of the message in the document to be watermarked

All systems examined up to this point assume that the data of the watermark and those of the document to be watermarked are independently processed. Instead, it is also possible to combine watermark and document, allowing better management of the compromise between strength and visibility.

A technology that is widely used is that of phase modulation. When transforming an image through DFT, and the image is symmetrical, a real transform is obtained. This allows representation of the image via module and phase. Phase components have a greater psycho-visual impact on the image with respect to amplitude components. For this reason, if the watermark is inserted within phase components with high redundancy, a possible attack designed to eliminate the watermark would cause an unacceptable alteration of the watermarked image. In addition, from the theory of communication, it is well known that the modulation phase is more resilient with respect to disturbances.

Another technology that is widely used is represented by amplitude modulation even if it is less suitable due to its lower contribution of amplitude components of the Fourier transform for image quality. In any case, amplitude modulation can be performed in the spatial domain of the image or on a part of it. In this sense, it is possible to modulate one or several colour components using bits of the watermark and luminance.

Another technology that is widely used is based on DCT coefficients and is similar to that seen in Chapter 3, in which two medium frequency coefficients are selected and are left unchanged or are inverted, depending on whether a one or a zero is to be entered and if the first coefficient is greater than the second or vice versa. This technology allows each image block of 8×8 bits to represent a bit of the watermark. This algorithm may however produce artefacts that can be avoided by carefully selecting the blocks to be used, the number of DCT coefficients to be used and the way in which the coefficients are modified.

4.9.5 Optimisation of the watermark detector

Most of the operations are directed at the insertion of the watermark but there are also watermark extraction operations that will be explained later in the following paragraph. A very simple way to compensate for geometric attacks, such as rotation or magnification, is represented by the possibility of applying an inverse transformation of the attacks themselves with greater or lesser accuracy. There is greater precision with which it is possible to define the inverse transformation along with the greater

quality of the final result. The choice of this transformation is, for the most part, suggested by the knowledge of the original document.

To take into consideration all the possible actions of reorientation and resizing of the watermarked image, two technologies can be used. The first is to enter the watermark in a space invariant to scaling and rotation, as shown above, while the second, which will be illustrated below, consists of a post-estimation of the geometrical transformation of the image being attacked and application of the inverse transform before extraction of the watermark. In practice, the first approach is of the preventive type while the second approach is a curative. The main advantage represented by the second method compared to the first is that it does not need to decrease the capacity of the watermark by narrowing the choice of the locations where the watermark itself can be inserted in an invariant space that could, among other aspects, be very small. This advantage must be balanced by the difficulty of performing an efficient estimation of the transformation as well as prompted by this type of curative approach.

A typical operation of this type is represented by the phase correlation for reorientation and resizing. It requires a fragment of the original image or original watermark as a reference. Subsequently, representation of a three-dimensional (3D) grid of the phase correlation maximum is performed. The axes of the graph correspond to the horizontal scale factor, to the vertical scale factor and to the rotation angle of the reference plane of correlation of the watermarked and possibly manipulated image. This 3D grid allows determination of the maximum phase correlation. The coordinates of this maximum provide the value of the horizontal scale factor, the vertical scale factor and the angle of rotation.

4.9.6 Watermarking of video images

The constant search for compromise between resilience and imperceptibility for the images is inevitably changed when passing to video. In the latter case, the ratio between the information contained in a given watermark and in the host signal becomes less critical. Furthermore, due to the addition of the temporal dimension, the visual distortions due to the watermark become difficult to manage. Further variation is represented by an increase in the number of possible attacks, such as variation in the number of frames per second or frame rate. There is, moreover, the issue of operating in real time and thus of reducing the computational load.

In most cases, that already seen for images is still valid, given the strong similarities. For example, the method of exchange of the DCT coefficients can safely be used for streams of JPEG images or for *i*-frames of MPEG streams. In any case, particular attention should be paid to the fact that the change to frames in order to enter the watermark can damage adjacent P and B type frames due to the form of motion compensation forming part of the MPEG video encoders/decoders.

In this sense, rather than entering the watermark within frames, it is also possible to insert it in the quantisation of the motion vector by different technologies.

4.10 Strength requirements

The strength of the watermark against removal is achieved by means of a compromise between image quality and the calculation time, prompting the attacker to do irreparable damage to the watermarked document or to spend an unreasonable amount of time in eliminating the watermark. A possible definition of resilience is provided by the proposal of the International Federation for the Phonographic Industry (IFPI) that states:

1. the mechanism of watermarking should not affect the quality of the recorded sound;
2. watermarking information should still be recoverable after the application of a wide range of operations of filtering and processing, including A/D and D/A conversions, compression, additive

or multiplicative noise, the addition of a second signal, distortion of the frequency response up to 15 dB, group delay distortions, etc.
3. there should be no other way to delete or alter the information entered without causing a degradation of the quality of the sound such as to render it unusable;
4. given a SNR of 20 dB or more, the hidden data channel should have a passband of 20 bps after the correction of errors, regardless of the level of the signal and type.

Requirements for the watermarking of images, videos and, in general, multimedia objects are very similar.

The objective of a potential attacker is to eliminate or to degrade the validity of the watermark and to control the watermarked content. In a manner similar to complex systems, the strength of the whole system depends on the strength of its weakest component. For this reason, the owner of the content must be sure that each component of the watermarking system is characterised by the same level of security and strength.

There are currently different classes of attacks on the schematics of watermarking and every attack focuses on a different component of the watermarking process. There are therefore attacks aimed at eliminating or reducing the presence of the watermark and attacks intended to reduce its effectiveness without, however, deleting it. For this reason, strength is a necessary condition but not the only requirement.

It is possible to divide the scenario of attacks into four areas:

1. resilience;
2. presentation;
3. interpretation;
4. legal aspects.

The attacks on strength imply a decrease in the signal and are the most obvious attacks out of the whole group. They start from simple operations, such as compression, to attack the watermark through specific programs.

Presentation attacks are designed to bring about what is defined as failure of the watermark detector in such a manner that the document becomes non-watermarked in checking of a possible watermark detector. They are not therefore aimed at the brutal elimination of the watermark.

Interpretation attacks are designed to bring about what is called counterfeiting of the watermark, generating a situation in which the original watermark can no longer be determined and thus losing its meaning.

Legal attacks indeed exploit legal issues and, in most cases, are beyond the capabilities of the sciences and engineering of the industry.

4.10.1 Signal decrease

Removal of the watermark through simple degradation of content is a possible type of attack. Watermarking system designers are well aware of this aspect when designing their systems that are intended to withstand typical operations such as compression, cutting, shading, printing and subsequent scanning.

One possible attack is represented by the addition of appropriate noise generated by subsequent watermarking. This operation generates a slight degradation of the watermarked document, if the algorithm used is of quality. In this sense, a public watermark is easier to overwrite with respect to a private watermark as the information of the latter can be hidden in one of the many possible locations. This allows the existence of the two watermarks, each with its own secret key, which can be used in two different locations without mutual degradation. The public watermark, on the contrary, must be inserted within a very specific location that is also known to the watermark detector and two

watermarks of this type overlap in the same location, interfering with each other. There are a number of programs that do not allow the inclusion of a second watermark if the presence of a first watermark is detected. This can be obviated by appropriately damaging the first watermark.

Another possible attack is that can be conducted by JPEG compression, as shown previously. In this case, the high-frequency components are eliminated, allowing the compression operation. Since the watermark is mostly inserted within those frequencies, to reduce its visibility, we have a situation whereby elimination of the same causes elimination of the watermark.

Another possible attack of this type is that conducted by using the medium operation. In fact, when there is large number of images, a potential attacker can perform a medium on them to produce an image containing the watermark. This method can be applied in video applications in which there is the same W watermark that is added to a series of images $\{I_i\}_{i=1,n}$ forming part of a video sequence. If f is the extraction function of the watermark characteristics, the sum of n watermarked images produces the following result $nW + \sum_{i=1}^{n} f(I_i)$ in which the expected value, for high values of n, is in fact nW. This allows generation of an approximate estimation of the watermark in order to delete it. To avoid this, it is possible to insert several watermarks and make them dependent on the image.

If an attacker knows the details of the watermarking algorithm, he/she can devise a targeted attack in order to delete the type of watermark produced. For this reason, it is very important that watermarking algorithms, similar to cryptography algorithms, are made public in such a manner that all possible attacks can be conducted on them in order to verify the level of resilience and reliability.

4.10.2 Malfunction of the watermarking detector

It has already been stated above that it is not strictly necessary to remove the watermark to make it unusable: in some cases, it can simply be manipulated in order to make it undetectable by the relative detector. These attacks are called presentation attacks and are mainly represented by geometric distortion and mosaic attacks.

Most watermarking systems are designed to withstand basic manipulation that can be implemented using standard programs available to the general public. They are often not able to withstand combinations of the same attacks and, in particular, random geometrical distortions. There are, in this sense, test programs of watermarking systems that generate such deformations. If A, B, C and D are the vertices of a figure, a generic point P of this figure can be expressed as:

$$p = \alpha[\beta A + (1 - \beta)D] + (1 - \alpha)[\beta B + (1 - \beta)C] \tag{4.22}$$

where $0 \leq \alpha, \beta \leq 1$ being the coordinates of P relative to the vertices.

This situation is represented schematically in Figure 4.6.

If a distortion is applied by moving the vertices of a small random quantity in both directions, the new coordinates of P are given by the formula (4.22), taking α and β constants, and using the new

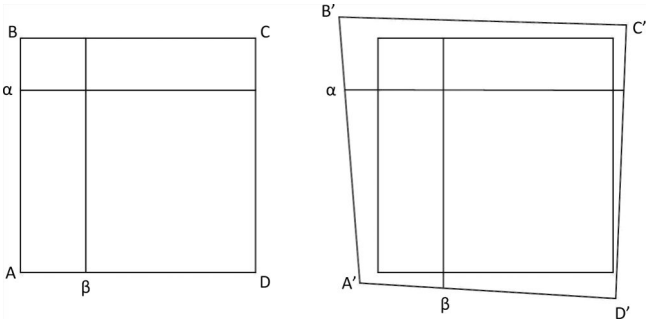


Figure 4.6 Example of bilinear random distortion.

vertices. This transformation does not remove the watermark but prevents certain systems from being able to detect it or to retrieve it.

There are other technologies to distort the images in a manner that is not perceptible in order to render the watermark undetectable. In addition to the bilinear distortion just illustrated, it is possible to apply a slight deviation to each pixel that is greater than in the centre of the figure and almost nothing at the edges. This can be achieved, for example, by means of a sinusoidal function. If X and Y represent the horizontal and vertical dimensions, respectively, and x and y represent the coordinates of the generic pixel, the changed coordinates x' and y' are:

$$x' = x + c\,sen\left(\frac{\pi y}{Y}\right) \tag{4.23a}$$

$$y' = y + c\,sen\left(\frac{\pi x}{X}\right) \tag{4.23b}$$

where c being a constant. A high frequency of the type $f = c\,\text{sen}\,(\omega_x x)\text{sen}(\omega_y y)[1 + n(x, y)]$ is, subsequently, added ω_x and ω_y being two appropriate spatial pulsations and n a random number, in such a manner that $x'' = x' + f, y'' = y' + f$.

If a medium-type JPEG compression is added to this distortion, a situation is reached whereby the watermark is not deleted but the detector is prevented from revealing it, desynchronising it appropriately. The problem is not therefore adding the watermark but recognising it at a later stage.

These reduced geometrical distortions can also be applied to video as long as the same geometrical parameters for all the frames are used, failure to do so possibly resulting in watching a video that is variable in size. Other options include increasing the strength by trying to anticipate all possible attacks or using multiple watermarking technologies.

Another limit that exists is that of the *bit rate* of the watermark to achieve a compromise with resilience of the same. It is evident that the greater the image, the easier it is to hide a small amount of information within the same. The opposite is also true, namely that an image may be excessively small to contain the watermark: this event represents the basis of a mosaic attack.

A false alarm occurs when a watermark detector reveals a watermark that in actual fact does not contain a watermark. If the image contains another watermark, this is referred to as watermarking collision. There is generally a collision if both marker subjects use the same scheme of watermarking. Both false alarms and collisions of watermarking can cause problems when an issue of ownership must be resolved. Since revelation of a watermark implies possession of the image itself, a possible false alarm can produce significant problems. A very important point is to be able to detect the presence of a watermark on the inside of an image, where the mark is made by a key K. Each individual system used commercially should be able to solve this problem.

4.10.3 Watermark counterfeiting

We have already seen that attacks can be conducted at each phase of the watermarking process. A type of attack that can be conducted is the so-called protocol attack. It has already been illustrated that most watermarks are able to resist the insertion of a second watermark since the location where the watermark is hidden is secret. If there is enough space within the document, the potential attacker could try to attack all the possible locations in order to eliminate the watermark, producing an inevitable degradation of the document itself. It might be objected that a potential attacker would simply add the second watermark but this objection loses its validity because the rightful owner can always prove to be in possession of the original non-watermarked document.

From a general point of view, the owner of a document adds the watermark w to the document, producing a document watermarked $I^* = I + w$. This document is distributed to all users and, if a I' suspect copy is found in circulation, the subtraction operation $I' - I^*$ need simply be performed to

obtain the watermark w if the two documents are identical and should provide a result close to w if the second document is derived from the first, remaining very similar and the watermarking is scheme is resilient. In this sense, if x represents the calculated datum, a correlation function $c(w, x)$ is used, that is appropriate to determine the degree of similarity between the original watermark and the result of the difference operation.

A possible attack by the operation of subtraction of a second watermark rather than by the addition of the same will now be illustrated. In this case, the owner of the document adds the watermark w, for example, to an image I, obtaining $I^* = I + w$. The potential attacker, rather than adding a second watermark w*, subtracts it, obtaining $I' = I^* - w^* = I^* + w - w^*$. At this point, the attacker ascertains that image I, rather than I*, represents the original. If the owner of the document verifies the presence of his/her watermark in image I' watermarked by the attacker, the following is obtained: $I' - I = w - w^*$ and the correlation function c is $c(w - w^*,w) = 1$. Since two watermarks can coexist within the same image, the subtraction operation performed by the attacker does not affect the presence of watermark w. The attacker, in this case, has not removed the watermark w but can demonstrate that $I - I' = w^* - w$ and that the correlation function c is $c(w^* - w,w^*) = 1$. In practice, the watermark of the attacker is present in the original image even if the owner has kept in it an appropriate manner. This type of attack works by subtracting the watermark rather than by adding it and is based on the reversibility of the watermarking system. To avoid this, it is possible to resort to a one-way method of insertion in a manner that makes it impossible for the attacker to perform the subtraction operation.

In most cases, a potential attacker has access to the watermark detector. This can be represented by a simple piece of software or an electronic device placed within a DVD player. Even if the attacker does not know the details of the watermarking system, he/she can always try to make small changes to the image until the detector fails to recognise the watermark. To perform this type of attack, the attacker starts building an image that appears to be very near the decision threshold of the detector. If we change the image by a small degree, the detector can vary its response from the presence of the watermark to the absence of the watermark, with probability close to 50%. It should be emphasised that the constructed image does not necessarily resemble the original image. This can be achieved by repeatedly defocusing the image until the detector no longer reveals the watermark or gradually replaces the pixels with grey.

The second step analyses the sensitivity of the detector to the changes of each pixel. The luminance of a given pixel is increased or decreased until the detector changes its response. This is done for each pixel. Performing this analysis, the attacker is able to find a combination of pixels and changes such that distortions of the image are reduced to a minimum while the effects on the detector are maximum, causing it to change the outgoing response.

A possible countermeasure consists of making the detection process random or making the process of decoding burdensome from a computational point of view. In the latter case, in order to make this approach really effective, it is necessary to resort to the assistance of sabotage-proof hardware.

There is another attack that can be conducted if the attacker has the algorithms of watermarking and revelation. In this case, the attacker marks the image already watermarked one or several times using the same method as the rightful owner. The attacker uses his/her own watermark as probabilistic indicator of the strength of the watermark itself. Then, he/she attempts the attack referred to above until all the new watermarks are removed. If we consider that the original watermark is weakened when the image is altered in a random manner, it is obvious that it is weakened also when inserting the new watermarks. For this reason, it can be assumed that the force of the new watermarks provides an upper limit of the strength of the original watermark and, thus, once all the new watermarks are removed, it is highly likely that the original watermark is also removed.

4.10.4 Watermark detection

The attack technologies seen so far are general and consider the watermarking algorithm unknown. In some cases, however, a number of details on watermarking and revelation are known and the watermarking process is relatively simple. In these cases, the hidden information can be detected and removed afterwards.

An attack that can be conducted with relative simplicity is *echo hiding* in which the echo is revealed and the same is removed by suitably reversing the convolution formula. In this situation, the problem is represented by the revelation of the echo without knowledge of the original file or the characteristics of the echo. To detect the echo, cepstrum analysis is used that analyses the signal $y(t)$ given by the sum of the signal $x(t)$ and by one of its delayed replicas $x(t - \Delta t)$ that is: $y(t) = x(t) + \delta\, x(t - \Delta t)$. If φ_{xx} indicates the power spectrum of x, then we have $\varphi_{yy}(f) = \varphi_{xx}(f)(1 + 2\delta\cos(2\pi f\Delta t) + \delta^2)$ whose logarithm is given by $\log(\varphi_{yy}(f)) \approx \log(\varphi_{xx}(f)) + 2\delta\cos(2\pi f\Delta t)$. The value is obtained according to frequency f: if we are calculating its power spectrum, a so-called quefrency Δt is obtained that represents the frequency of $\cos(2\pi f\Delta t)$. If the autocovariance of the function obtained is subsequently calculated, the peak that appears at quefrenza Δt is increased, isolating it appropriately.

Experiments on random signals and on musical signals have demonstrated that this method provides accurate estimates of the delay when an artificial echo is added to the signal. Echoes with delays ranging between 0.5 and 3 ms are considered in detection function. Under 0.5 ms, the function begins to operate incorrectly while above 3 ms the echo becomes too audible.

In some cases, the image that must be watermarked has features that can provide information of interest for possible attacks. A typical case is represented by an image characterised by a small number of colours, providing a histogram in which clear peaks appear. Twin peak attacks arise from this to retrieve simple spread spectrum, watermarks. Such an attack is not shown in the following for reasons of space.

4.10.5 System architectures

The attacks seen so far are based on signal processing while many attacks are based on other weak points such as human factors, user interfaces and weaknesses in implementation. With regard to human factors, the typical user possesses a limited knowledge of the mechanisms of watermarking and does not intend to spend a great deal of time in order to become familiar with the use of a program. Watermarking should be a black box within which the document to be watermarked is inserted and the watermarked image can be obtained in output, since further knowledge is not required. Also, usually, designers and artists do not like degrading their works by watermarking or exposing the same, from the security and copyright point of view.

With regard to user interface, within the context of the human factors referred to above, it becomes an essential component for the security of the system. It should prevent the user accidentally deleting the watermark from the document of interest. One possible improvement that could be implemented is represented by an indication of the strength of the watermark to show the user how the watermark reacts in real time to changes to the image. Ideally, watermarking should represent the last operation to be performed on the document. In this sense, the software itself should delay the watermarking operation until the user decides to save the document. In a similar manner, when the user loads the document, the software should extract the watermark, giving the same user the option of working on the non-watermarked document. In this sense, watermarking should be very transparent to the user.

With regard to weaknesses of implementation, it must be remembered that the majority of attacks that are carried out in respect of cryptographic systems exploit vulnerabilities of the system rather than their own technologies of cryptoanalysis and the same applies for watermarking systems.

In many cases, watermarking software provides an identifier and a password. This information is securely included within the document. A first method of attacking such a system is to use some form of debugging technology that allows entry into the software and disabling of the password control mechanism. A second method is, given a certain identifier, to try all the possible combinations of the password, provided the latter is not too long.

4.11 Digital fingerprint

Fingerprints are the specific characteristics of an object that allow it to be distinguished from similar objects. They have various applications including copyright protection. Fingerprints are generally not aimed at the protection of counterfeiting but at the tracking of documents should a user make unauthorised copies and distribute them. For example, in the case of encrypted satellite transmission, the end user could be equipped with a secret key, which is typically contained in the smart card that is inserted in the receiver, and the broadcasting station could insert bits in each traffic packet to detect unauthorised use of the emitted signal.

Fingerprints can also be used for high-speed searches giving each volume an appropriate code composed of a small number of characters. There are however more complex technologies that employ a hashing operation to search for objects within an archive. Fingerprint therefore refers to a process of assigning an identity to an object among many similar objects by means of which it is possible to trace the object during its existence, its use or its movements. Fingerprints have been used since time immemorial for various applications such as human fingerprinting, firearms, serial numbers, explosives, maps and mobile phones.

With regard to the fingerprint, it is well known that every individual is characterised by a pattern different from that of other individuals. The fingerprints of criminals or prisoners are collected for the purposes of investigation. As they represent a very practical means for identification, some countries use them in identification cards and in passports for citizens. They are also used in access control systems. In a similar manner, other human characteristics such as the iris or the voice are also used for identification.

With regard to firearms, each of them has its own identifying code that depends on the manufacturer and type of weapon.

Serial numbers are used by manufacturers and are unique to each product in order to identify it during its normal life cycle.

Explosives are encoded by fine particles that can be found after the explosion. From subsequent examination of the particles, it is possible to trace back to the producer, to the type of explosive and the date of processing.

Maps are designed with certain slight and voluntary variations, with respect to reality, to identify the copies.

In the case of mobile phones, each of them is equipped with a unique code International Mobile Equipment Identity (IMEI) that is transmitted to the telephone company every time the phone asks for access to the network. In the case of loss or theft, the owner of the phone can communicate the code to the telephone company to block access to the network. Moreover, the same code is communicated, in general, to all the telephone companies to immediately identify the fraudulent user of the lost or stolen phone via identification of the new subscriber identity module (SIM) card that has replaced that of the legitimate owner.

Certain fingerprints affixed to digital data represent a simple and inexpensive means for the protection of copyright and their growth is constant in the world of computers and communications.

Definitions:

1. mark: a part of a subject characterised by a series of possible states;
2. fingerprint: a set of marks;

3. distributor: an authorised provider of a fingerprint to users;
4. authorised user: a person that has access to an object equipped with a fingerprint;
5. attacker: a person who manages to gain unauthorised access to objects with fingerprint.

The purpose of the distributor is to identify the user compromised by an attacker and the purpose of attacking is to avoid identification by the distributor.

This fingerprint only provides a means for detection and not for prevention, in the hope that the possibility of detection of illegal uses acts as a deterrent to prevent any fraudulent actions by potential attackers.

The fingerprint aimed at tracking copies and their reduction also includes tolerance in collusion that means that if an attacker has access to a certain number of copies, he/she should not be able to find, generate or delete fingerprints through comparison of the copies. In particular, fingerprints must have a common intersection. Another important feature is the tolerance of the quality of the object that means that the fingerprint should not diminish the usefulness or quality of the object in question to a decisive extent. A further quality is represented by the tolerance to manipulation of the object that means that if an attacker effectively sabotages an object, the fingerprint must still remain intelligible unless an amount of noise has been added that makes the object itself unusable. In particular, the fingerprint should be able to tolerate data compression with loss.

Fingerprints can be classified according to:

1. the objects on which to insert the fingerprint;
2. sensitivity to detection;
3. the methods of insertion of the fingerprint;
4. the print generated.

With regard to the objects on which to insert the fingerprint, two methods of classification can be distinguished: fingerprint and physical print. Reference is made to fingerprint when the format itself is digital and a computer can be used to carry out the above operation. Physical print refers to when use is made of specific physical characteristics of the object. A specific case is represented by human fingerprints, the human iris, the human voice and the particles included in explosives.

With regard to sensitivity to detection, the following perfect classification can be performed: perfect fingerprint, statistical fingerprint and threshold fingerprint. If any alteration of the objects that makes the digital fingerprint unusable also makes the object unusable, this is referred to as perfect fingerprint. With regard to statistical fingerprint, given a certain number of objects not used correctly, the fingerprint generator can acquire a certain degree of confidence concerning identification of the attacker. Threshold fingerprint is a synthesis of the two types just illustrated. In this sense, it allows us to identify an illegal copy only when a predetermined threshold is exceeded, allowing us to make copies of the object as long as the threshold itself is not exceeded: when the number of copies exceeds the threshold, the attacker is identified. The methods of insertion of the fingerprint can be classified into recognition, cancellation, addition and modification.

In relation to generated fingerprint, two types can be distinguished: discrete and continuous fingerprint. If the fingerprint generated is characterised by a discontinuous set of values, this is referred to as discrete fingerprint. If the fingerprint generated is characterised by a continuous set of values, there is referred to as continuous fingerprint. Most fingerprints belong to the latter category.

CHAPTER 5

SECURITY IN WIRED NETWORKS

5.1 Introduction

Every day there are reports of attacks on networks and computers connected to them, managing to breach even sites with a high degree of security. The concepts discussed in this chapter are also valid for wireless networks, which will be explained later.

Attacks occur more frequently than is thought because most of them are not made public in order not to lose credibility in the eyes of users. Often the attacks do not originate from the outside but from within organisation.

To understand how to best protect our network, it is necessary to try and think like the aggressor, as the majority of attacks are not random and the attacker that tries to breach a system usually has a very clear final objective.

Terms that are often used to identify attackers are aggressor, hacker and cracker. There are well-defined differences between these terms. An attacker is a subject that tries to steal or damage what another subject possesses. He/she may be a technical expert and his/her behaviour is very similar to that of a spy or a thief. A hacker is a subject that is very familiar with computers and networks. He/she tries to understand the functionality of the systems from all points of view. The hacker can have either a positive or a negative connotation. A cracker is a hacker who tries to penetrate systems and networks illegally to perform unlawful operations.

To characterise the different behaviours of hackers, three terms have been coined: white hat, grey hat and black hat. A white hat hacker is a subject that manages to find a security vulnerability of a program and reports it publicly. A black hat hacker is a subject that manages to find a security vulnerability of a program and uses it for illicit purposes. A grey hat hacker is a subject classifiable as a white hat hacker at day, carrying out lawful work, and a black hat hacker at night, performing unlawful work.

It has already been stated that the attacks may have origin both within and outside the organisation. Attacks from within the organisation represent about 70% of the total, motivated by staff with bad intentions or former staff, for reasons of revenge. In these attacks, the additional use of firewalls (which will be discussed in detail in the following) does not offer any protection as the attackers are very familiar with the network from the inside. Most organisations employ a huge amount of resources to defend themselves against attacks from the outside not considering that the majority of attacks happen from within. In many cases, even the location of network administration is located in a

place accessible to all and with weak or even absent passwords. In this case, any subject with bad intentions and a minimum of technical knowledge can easily access the same and attack the system or create a flaw within it in order to be able to attack, at a later date, from the outside. A constantly growing threat is represented by theft and data compromise rather than by destruction of the same. In this situation, reference is made to industrial espionage.

Attacks from the outside may still originate from disloyal internal staff as well as from a vast number of other subjects, all, however, interested in the theft or destruction of data, or the deactivation of network resources in order to acquire a direct or competitive benefit. One reason why attacks are performed is to acquire a certain reputation in a short space of time.

However, there are other reasons for attacking a system including the so-called email bounce that consists of attacking the email system of another subject in order to use it as a type of email spam repeater (unsolicited advertising). In this case, hackers are called spammers and their aim is to reach with their unsolicited advertising the greatest number of users in order to arouse some interest in the products being advertised. This results in a violation of the privacy of others and an overloading of the email system that may no longer function correctly for normal users.

5.2 Introduction to security policies and risk analysis

It is very important, for the purposes of a correct security policy, to carry out an analysis of risks through which the assets to be protected and the potential threats against them are identified. In this sense, there is an exploration of the assets requiring protection, the dangers from which the assets are to be protected, the identity of persons that could attack the network, the motives that lead these individuals to attack the network, the probability of breach of these assets, the costs of withstanding the breaching of an asset, the cost of recovery from an attack and the policies for the protection of assets and the related costs and the existence of international references to determine the level of security of a network.

With regard to the goods to be protected, these are physical resources, intellectual resources, time resources and resources of perception.

Physical resources are all those assets that take a physical form. They can be workstations, servers, terminals, network and peripheral devices. In practice, any computing resource characterised by physical form is a physical resource. These must be identified at the risk analysis stage, the objective of which is to develop a security plan characterised by the best cost/benefit ratio. During this analysis, the more obvious risk areas and their solutions should not be underestimated.

Intellectual resources are for the most part in electronic format and are much more difficult to identify with respect to physical resources. An intellectual resource is information of any kind that plays an important role in the normal functioning of an organisation. It can be software, technical information, financial information, data storage areas, etc. It is very important to identify with precision all the intellectual resources available to an organisation in order to achieve correct risk analysis. The same email messages exchanged between the employees of an organisation represent an intellectual resource.

Temporal resources represent an important resource that an organisation has. When the cost of lost time must be assessed, we must also consider all its possible consequences, assessing all the possible impacts due to the loss of a resource as it is rarely the case that the loss of a single resource alone has an impact on a single individual of the organisation.

Resources of perception are those resources that, if attacked, cause a high perception of damage. For example, consider the site of a bank that is attacked and the possible effects on the confidence of their savers. In this sense, the protection of the image becomes more important than the protection of the resources themselves.

Network attacks can originate from any one subject that has access to the network. If the network is connected to the Internet, it is easy to imagine that there is a very large and diverse number of potential attackers depending on the size of the organisation under attack and the type of access allowed to the network. In carrying out risk analysis, it is necessary to consider all the possible attackers that could be internal systems, access to off-site offices, access by Wide Area Network (WAN) link to a business partner, access via the Internet, access through a group of modems, etc. In this sense, the subject attacking the network is secondary to the means available to ensure access to the network resources.

Potential threats can be employees, temporary staff, consultants, competitors, individuals whose objectives are different from those of the organisation to which they belong, subjects who want to take revenge on the organisation or on one of its employees and subjects who want to gain fame due to the organisation itself. Depending on the type of organisation, the number and type of potential threats can easily increase.

It is also very important to define and quantify the probability of attack on our network to defend it suitably and to justify the costs necessary to undertake appropriate security measures.

To quantify the cost of an attack, it is necessary to take into account all of the resources that could be attacked and the relative cost: the cost of the attack is provided by the cost of all the resources attacked. In some cases, quantification of the cost becomes difficult: think of the case in which, following an attack, all the details of a new product are stolen and these details are used to develop a better product. In this case, the losses become incalculable also as a result of indirect costs such as loss of the confidence of customers, investors and staff. It is very important, therefore, to quantify not only the direct costs but also the indirect ones. In addition to downtime costs, the costs of recovery due to a fault or an attack must also be taken into consideration. In this sense, it is necessary to quantify all the levels of loss of the financial impact. Take for example a server that contains business information. Here, the momentary instability due to disconnection of all connected users, the cost due to the unavailability of resources, the cost of restoration of critical files that have been deleted or damaged, the cost of restoring any hardware components damaged and the cost of restoring information must be taken into consideration. If we take into account the cost of the various levels of damage, together with an estimate of the frequency of the attack, this provides an idea of the cost of recovery that must support an organisation as a result of an attack, allowing a correct assessment to be made of the benefit/cost ratio of the security measures to be undertaken.

A general rule should be that the cost of all the security measures taken to protect a given asset should be lower than the cost needed to restore this asset following a disaster. In this sense, it is very important to quantify both the potential threats and the related costs of recovery, remembering that, even if security measures are regarded as essential in a modern network, it is, however, always necessary to justify their costs.

It should also be remembered that greater protection of the network causes lower ease of access and use of the resources of the same network, and this might cause an increase in the working times of users that employ it, with a consequent increase in the time cost.

Once all the risks have been quantified, the relative frequencies of occurrence and the costs of recovery, proper risk analysis has been carried out that allows a security directive to be drawn up representing a discussion tool to identify and clarify the security aims and objectives for the entire organisation. A quality security policy involves every employee concerning their responsibilities regarding the security of the entire network.

Security criteria are usually motivated by specific issues. In order for a security policy to be acceptable, it must be consistent with other corporate policies, accepted by the network administration service, which can be applied using the equipment and the existing procedures, consistent with the law.

With regard to consistency, this ensures that users consider the same rules as reasonable and rational. A security directive should reflect both acceptable business practices and the perspective of the organisation on matters of security. If the organisation pays no attention to the physical security or the use of corporate assets, the same is unlikely to impose strict criteria of use of the network.

With regard to acceptance, in order for a guideline to be applicable, it must be accepted within the organisation itself at all levels of management and must be applied equally to all users of the network.

With regard to applicability, this is a fundamental precondition for the functionality of a security directive, creating policies to always be observed and not left to the will of the individual employee. Non-compliance with a criterion of use of the network can generate a domino effect, leading to most of the staff failing to comply with all the security criteria of use of the network. In this sense, compliance for 100% of the time need not be verified but it would be wise to resort to a monitoring system if its application were to become a problem. In many cases, it is not sufficient to control all the aspects of a certain directive but it is very important that the same is appropriately disseminated in its entirely in order to be fully applicable. It is also very important to include appropriate indications of private property by resorting to logon script and messages on the terminals.

With regard to conformity with the law, it is very important to ensure that the security criteria provided comply with the law.

A good security policy should:

1. be easily accessible by all employees;
2. be clear in the definition of objectives;
3. define with precision every point contained in the policy;
4. clarify the position of the organisation on every issue;
5. accurately describe the criterion in every point;
6. clarify the circumstances of applicability;
7. clarify the roles and responsibilities of all employees;
8. explain the consequences in the event of non-compliance with the policy;
9. provide a contact to obtain more information on the subject;
10. define the level of security for each user;
11. clarify the organisation's position on any issues not specified in detail.

With regard to accessibility, it is very important to disclose the security policy carefully within the organisation in order to achieve the best results. In this sense, use can also be made of the already mentioned logon script and messages on the terminals. If the organisation has an internal set of regulations for employees, it is very important, if possible, to include the security directive within this document. To this end, a possible internal web site may be used to disseminate the same directive.

Additionally, a statement of intent that explains the importance of the security of the organisation can be very effective, albeit sometimes seeming banal, demonstrating that the criteria are not unsubstantiated but the result of detailed research and analysis. In fact, it is demonstrated that employees more easily accept regulations if the same are able to understand the benefits to be derived from them.

With regard to the specification of any relevance, it is very important to be clear and precise on every point.

It is also very important to justify every security policy provided, in order to demonstrate the importance for all users of the network, possibly providing more information on the anticipated impact, ensuring that there is no uncertainty felt by those expected to comply with it.

It is also very important that all members of the organisation are responsible for the security of assets, as the same is an issue that affects everyone. Furthermore, the manager for the implementation of security policies and the type of authority that is recognised by the organisation must be identified with precision.

In the event of non-compliance with the directives of security by an employee, the organisation must react immediately and implement all the corrective policies that must be very clear and well defined.

Since it is always difficult to define with clarity all the potential aspects of a policy, it is very important to locate and indicate a manager that is able to provide further information. It is advisable to

refer to that manager as a corporate department rather than by name, because people can change role but the corporate department remains same.

Confidentiality is also very important, which plays a key role for every type of organisation. In this sense, the organisation must clearly communicate ownership of the stored information. Failure to do so may result in employees considering such information as theirs.

With regard to security standards, restrictive criteria can be provided, that is what is not expressly permitted is prohibited or open criteria, that is what is not expressly forbidden is permitted. In this way, a reference point is provided if there were any issues that are not specifically described in the security criteria. It is very important to include statements that indicate the organisation's stance on matters not expressly covered in the security directive. Usually, a certain position on security is taken to begin with, which is generally rigid, gradually adding other criteria depending on future developments.

5.3 Firewall

A firewall is a system or group of systems that operates an access control on network traffic, unlike a router that only directs network traffic. Once the levels of connectivity to be provided have been defined, the firewall ensures that no access beyond that already defined is permitted. The firewall ensures that the policy of access control is respected by all users of the network.

Firewalls, in a manner similar to other network devices, have the task of controlling network traffic. However, unlike other network devices, they must check the traffic by taking into account that not all of the data packets that they display may be what they seem.

They operate by assuming that the host may try to deceive it by capturing the information fraudulently. The firewall may not use the rules of communication as a support but, on the contrary, it must assume that the rules are not being followed. This eventuality has an enormous influence on the design of firewalls that must take into account any possibility.

Usually firewalls are used to control access between an internal network and an external network, typically the Internet. However, there are a number of other situations in which firewalls are used. They can, for example, be used to control access through calls via modem, or for external connections to remote business partners or even to control the internal traffic to a network, and can be divided into various internal controlled areas.

Before choosing a firewall model, it is very important to define the rules that control the flow of data traffic in input and output in order to define a correct access control policy. In practice, this is to define who can access what for all users or groups of users. A correct access control policy simply defines the direction of the flows of data input and output from various points of the network. It also specifies the type of permissible traffic, assuming that all other addresses are blocked. There are several parameters that must be considered when defining an access control policy. The most important parameters are as follows.

1. Direction that is a description of the flow of traffic allowed and based on direction, such as, for example, traffic from the Internet to the internal network in input and traffic from the internal network to the Internet in output.
2. Service that represents the type of application server being accessed, such as, for example, access to the web (http), File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP).
3. Specific host that allows a greater level of definition with respect to the single direction or service. In this sense, it is possible to define what can be performed, in terms of traffic flow, by every single host.
4. Individual users that allows us to specify, through appropriate authentication, the type of traffic that each authenticated and authorised user can perform on the network.

5. Timetable that allows us to define what type of traffic can be conducted on the network depending on times or time bands that are well defined.
6. Public or private that allows us to select the mode of transmission of traffic depending on the type of network. In this sense, it might be advisable to increase the level of security of transmissions, using cryptography, if these occur on a public network, leaving transmission in plaintext if the same occur on a private network.
7. Quality of service that allows us to restrict access using as a parameter the amount of bandwidth available.
8. Role that allows network administrators to group individuals with similar needs, thereby simplifying the complexity of the access control and the work of management and control.

In its simplest form, a firewall consists of a bastion host to which all users refer as if to an Internet server. It can be any existing computer. The bastion host must be particularly prepared, by means of suitable software and hardware, to repel attacks on the same internal network. It is placed at the first line of defence. It represents a fundamental point for all communication between the network and the outside environment, typically the Internet, and vice versa. In practice, no computer inside the network can access outside without having to pass through the bastion host and no computer from the outside can access within the network without having to pass through the bastion host. In this way, concentrating every access and output into a single machine allows full control of the inputs and outputs, appropriately programming a single machine for this type of traffic (Figure 5.1).

In addition to traditional software and hardware firewalls, it is very important to use a security device called screening router that is capable of considerably increasing the level of security. It is able to filter the data streams based on specific criteria. The screening router can be a PC or a workstation or may be a remotely programmable device on the network. At the time of purchase, this device comes equipped with all the necessary software. Once the security policy has been defined, the device is programmed via a set of rules that, at the end of programming, are stored in an appropriate file. These rules tell the screening router how to behave with each incoming and outgoing data flow according to the origin and a number of other parameters. The screening router is installed between the local network and the network from which attacks or threats to the security may originate. In order to ensure maximum security, screening routers must be the sole point of transit between the two networks (Figure 5.2).

The fact that attacks and intrusions do not come only from outside but also from the inside is often overlooked. In this sense, the screening router, a computer, can be used to check the internal traffic between the various departments of an organisation, operating according to the preset rules of security. If using an in individual computer as firewall, a network card that is different for each subnet being controlled must be provided (Figure 5.3).

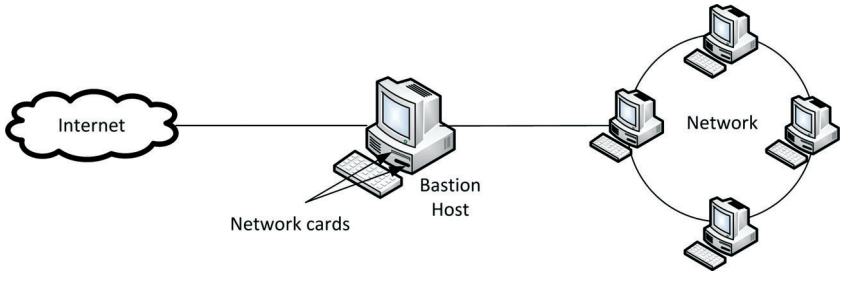


Figure 5.1 Use of a normal network PC as bastion host.

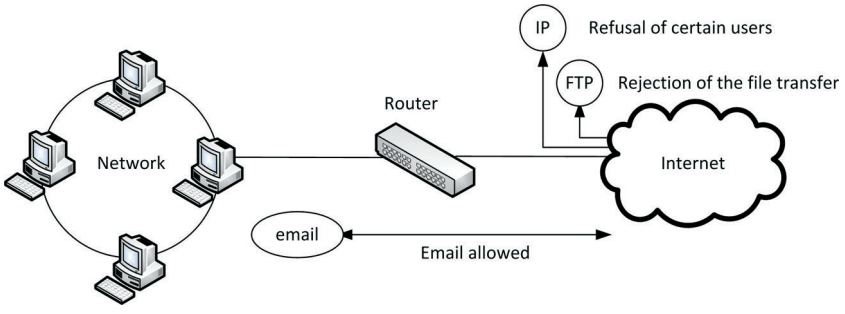


Figure 5.2 Screening router.

5.3.1 Design of a firewall

The use of a firewall came about from the need for the connection of an internal network of an organisation to an outside network, typically the Internet. To do this, it is necessary to understand, at the outset, the type and number of communications that must occur from the inside towards the outside and vice versa. The most important parameters for the design of a firewall, which must be taken into account, are:

1. need of users to take files from a remote server;
2. need of external users to send files to servers on the network;
3. need to prevent any access to well-determined users;
4. need to include web pages accessible from the Internet;
5. need for any telnet support;
6. need for the use of human resources for the management of firewall security;
7. need to develop an emergency plan in case of intrusion on the local network.

We have seen in Chapter 1 that Internet programs communicate using the appropriate ports, most of which are fixed according to the type of application. For example, FTP uses the ports 20 and 21; HyperText Transport Protocol (HTTP) uses port 80; SMTP uses port 25; Telnet uses port 23; Whois

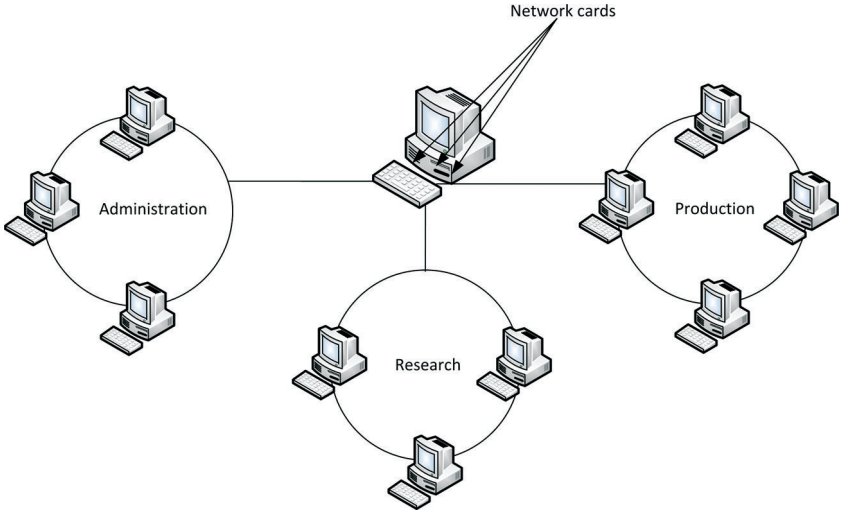


Figure 5.3 Firewall represented by a computer equipped with a network card for each of the subnets to be controlled.

uses port 63 and so on. The firewall, depending on the type of activity required, can partially or totally inhibit the use of certain ports, blocking the activity of relevant applications that use them.

5.3.2 Limits of firewalls

Even if an organisation is not equipped with access to the Internet, it can always be subject to the loss of confidential information. This may occur, for example, via normal paper correspondence, which can be stolen, acquiring valuable information.

Organisations often invest in very powerful and expensive firewalls, forgetting the protection of normal routes of exchange of information similar to the exchange of documents between authorised users. In this sense, the firewall can do nothing to prevent this because users can exchange any type of information on the network, including confidential information, because the same are in possession of the necessary permissions.

It is therefore evident that firewalls have limitations with respect to the degree of security that they can guarantee. They are not, for example, able to guarantee data integrity, protection against disasters, authentication of the data sources and data confidentiality.

With regard to data integrity, even if the most recent firewalls are equipped with software to detect the transit of viruses in incoming packets, this possibility is not a strength of the same firewalls. In fact, for large volumes of traffic, if the firewall were to perform this work indiscriminately, this would considerably slow its operating speed. In this sense, for the control of data integrity, it is necessary to establish a reduced number of access points (AP) in which to position machines dedicated to the control of incoming viruses that, after analysing the incoming packets, let them pass if free of viruses.

With regard to protection against disasters, no firewall is able to protect data from fire, earthquakes, floods and other disasters. It must be remembered that the firewall controls access to network resources but does not guarantee any physical security against intrusion or physical destruction.

With regard to the authentication of data sources, the firewall is not capable of performing this function, since it cannot prevent one of the major weaknesses of the Transfer Control Protocol/Internet Protocol (TCP/IP) protocol, not being able to ensure that the identity of a person is actually that which has been declared.

With regard to data confidentiality, the firewall can do nothing to protect the same on the internal network. Many last generation firewalls use appropriate tools for the encoding of data output, but this requires the use of a similar firewall on the reception side, and this is not always possible.

5.3.3 Risk regions

In order to prevent any external entities from entering a local network, the internal network must be completely isolated from the external network, directed towards the Internet. As networks are generally connected via cables, the cables of the local network must not be in contact with the cables of the external network. It is however always possible to ensure access to the Internet by providing two series of cables, a series for the internal network and a series for the external network. These cables will be connected, on each computer that requires access to the external network, via distinct network cards: each card connects the computer to a different port, isolating the two ports and consequently the two networks. Access to one or another network may be performed by switching software operated directly by users. This situation is represented schematically in Figure 5.4.

The use of a firewall requires the search for a compromise between security and functionality. It must allow authorised users to be able to communicate on the internal network and the external network without excessive constraints but must be able to define the access to a restricted area of users that are not recognised. Usually, these areas are called risk regions. A risk region consists of information

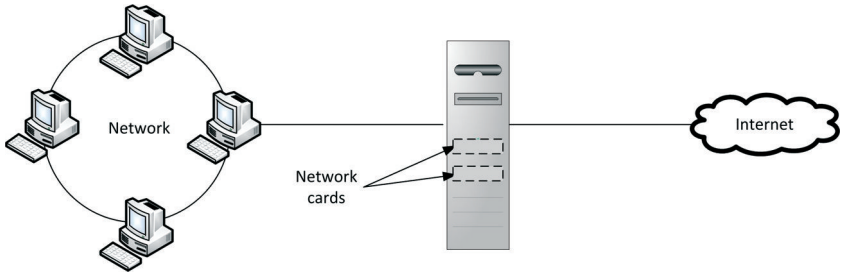


Figure 5.4 Use of two different network adapters for access to the internal network and the external network.

and network systems that a hacker can manipulate during an attack. If an internal network is connected to an external network without resorting to any security system, the entire internal network becomes a risk region.

Each host on an unsecured network becomes a vulnerable subject and all the data contained on it may be potentially manipulated and/or destroyed. If we are using a firewall configured in a suitable manner, the risk region is limited to the firewall itself or to a restricted number of hosts around it. This situation is represented schematically in Figure 5.5.

If a hacker manages to penetrate within a firewall, then the risk region will also extend to the protected internal network. In this case, the hacker can use the firewall as an instrument of attack towards the network itself, possessing from the same all the confidential information relating to the network. However, since the attack is carried out through the firewall, the same attack becomes more difficult as the hackers must move within a controlled environment and are subjected to a higher probability of detection and blocking. On the contrary, the absence of a firewall allows a hacker to move easily within a network without any control or restriction. Moreover, even if the attacker manages to open a passage through a firewall, all his/her activities are monitored by the firewall, allowing the network administrator to detect abnormal activity and to readily close the passage.

A well-designed and quality firewall allows the system administrator to have valuable information about passing traffic, the type of traffic and the number of packets that have passed or that have tried to pass, a parameter that allows quantification of the number of attack attempts made by hackers.

5.3.4 Introduction to firewalls

The simplest form of defence that can be performed on a network is the installation of a screening router that conducts packet filtering at the data link and network level allowing the network traffic to be controlled without requiring the modification of any application or host (Figure 5.6).

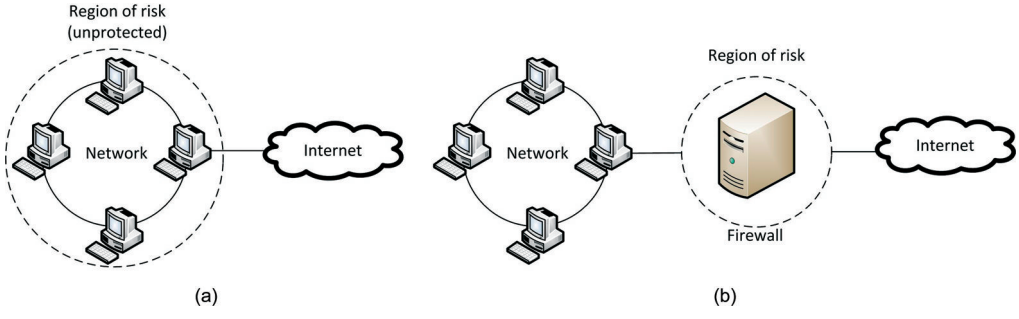


Figure 5.5 The regions of risk (a) on an unsecured network and (b) on a network protected by firewall.

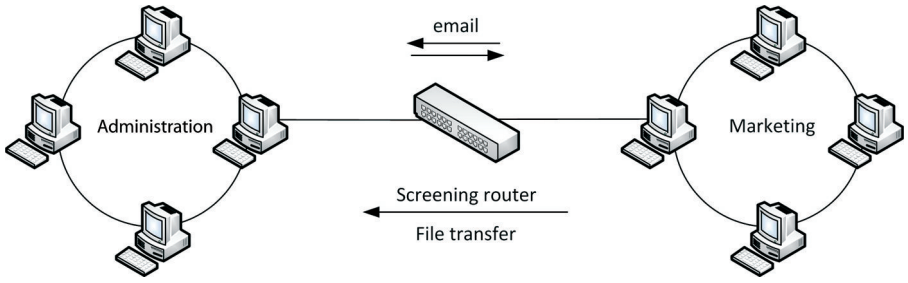


Figure 5.6 Example of installation of a screening router.

Screening routers are very comfortable and cheap but they are not a very effective solution since they are valid for limited network use, only being able to operate at the transportation network levels of the International Organization for Standardization/Open System Interconnection (ISO/OSI) model (Figure 5.7).

To be valid, a firewall should operate at all levels of the ISO/OSI model and for this reason multiple devices are often used, each operating at different levels.

The limitation of screening routers is the fact that they filter packets on the basis of inadequate or in any case limited data. In fact, as they can only operate at data link and network levels, they can access limited information such as IP addresses, port numbers, TCP flags and a few other forms of data available at those levels. Since they do not have contextual information available, they are rarely capable of filtering protocols such as User Datagram Protocol (UDP). Screening routers can prevent many attacks but could not warn the administrator that the same attacks had occurred. For this reason, it is very important to use multiple devices in order to check all levels of the ISO/OSI model.

Firewalls, being equipped with filters that operate at the higher levels of the ISO/OSI model, can operate as they can rely on all the information that reaches the application level. At the same time, firewalls can also operate at network and transport levels, controlling the IP and TCP headers of packets that are received or sent. For this reason, firewall may decide, on the basis of filtering rules that can be defined as desired by the network administrator, whether to route the packets or not.

5.3.5 Types of firewalls

Firewalls can be divided into three major groups:

1. network-level firewalls;
2. application-level firewalls;
3. circuit-level firewalls.

Each of these firewalls uses a different approach to fulfil its mission and will be illustrated in detail in the following sections.

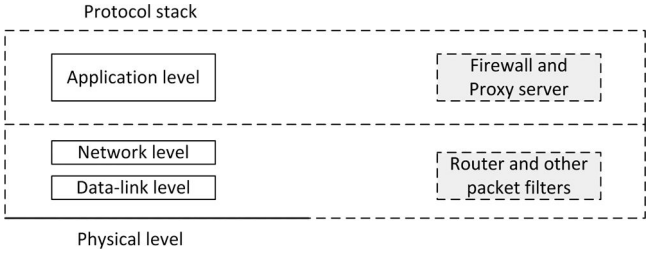


Figure 5.7 Use of various devices at various levels of the ISO/OSI model.

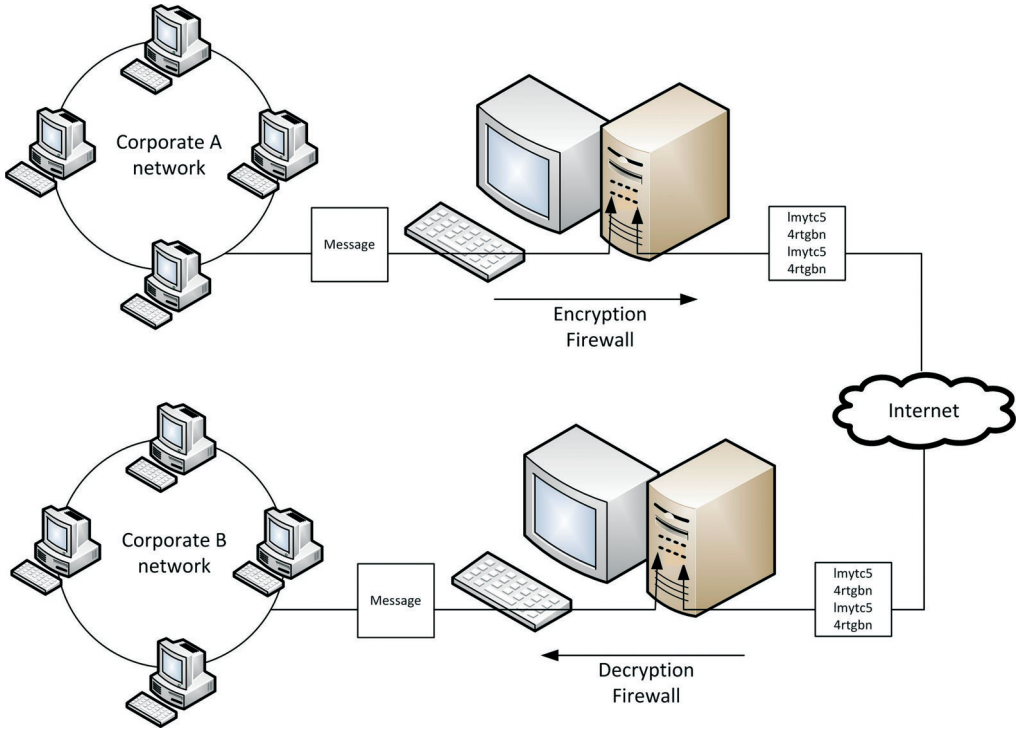


Figure 5.8 Example of firewalls equipped with cryptographic functions.

Most firewalls also support one or several levels of encryption, protecting the data output before sending it to the external network and then onto the Internet. On the receiving side, a similar firewall decodes the data and places it on the internal network in plaintext. Using cryptography, it is possible to connect securely and through the Internet various locations of the same organisation, including those situated a great distance from each other (Figure 5.8).

5.3.5.1 Network-level firewall

Network-level firewall is generally a screening router that operates on a computer. It examines the address of each packet to decide if the same packet should be transferred to the internal network or blocked. Because IP packets contain information about the sender and the recipient and a series of other information, such information is used by the firewall to manage the level of access of these packets.

In general, the screening router is programmed to block all packets that contain a specific address such as the sender or recipient. In this sense, a blacklist of all the addresses that should be blocked is drawn up and this list is provided to the same screening router. To do this, we will be helped by appropriate software to prepare the lists by inserting block addresses of entire sites or entire networks and not of individual computers.

Packets reaching the screening router can contain any kind of information and, depending on how the device itself has been programmed, different behaviour will be demonstrated depending on the type of communication activity that is required.

It is usually possible to program a screening router that takes into account, packet by packet, the following information before making any decision:

1. Source address of the data.

2. Destination address of the data.
3. Data protocol (TCP, UDP or Internet Control Message Protocol (ICMP)).
4. Port of application of origin and destination of the service requested.
5. Initial packet of a connection request.

If a network-level firewall is correctly installed and configured, it will be very quick in operation and virtually transparent to users except in the case where an unauthorised activity is being run and in this case, the users themselves are blocked. It is clear that the users on the blacklist will not be able to perform any type of activity on the network.

5.3.5.2 Application-level firewall

The application-level firewall is generally a host computer on which a software named proxy server operates and for this reason, this type of firewall is abbreviated to proxy server. A proxy server communicates with other external servers on the network on behalf of internal users of the network, by controlling, in fact, the traffic between the two networks. In many cases, a proxy server can manage communications of one or more users with one or more existing services outside the network, which will communicate directly with the proxy server: it is the task of the proxy server to sort correctly communications with internal users. On a network, a proxy server can provide access to an application to be kept secure without requiring transmission of the same in plaintext.

If we are using an application-level firewall, the internal network is not connected to the Internet and the traffic that flows on a network does not flow onto the other network because the two networks are physically disconnected. The proxy server transfers single isolated copies of packets from one network to the other. Application-layer firewalls are able to anonymise the origin of the internal connection, preventing any external hackers from gathering information about users on the internal network.

Proxy servers are able to recognise network protocols and it is therefore possible to configure them in such a way as to check the services that are allowed on the network, with the difference being, with respect to the router, the ability to use a proxy server for each of the services to be offered.

When using an application-level proxy server, the users of the same network must use programs that are able to support these proxy servers. In most cases, network protocols (TCP/IP, HTTP, FTP, etc.) are able to support them. This functionality is also implemented in web browsers and users can manage it with relative ease.

Application-layer firewalls represent a very convenient means of selecting the type and volume of traffic that accesses certain sites. Because of the ability to completely isolate the internal network from the external network, they ensure a high level of security. As they must analyse all the packets in transit and decide on what to do, they tend to reduce the performance of the network itself: this means that an application-level firewall is slower than a network-level firewall and in this sense, to bridge this gap, it is necessary to run the proxy software on a high-performance computer.

5.3.5.3 Circuit-level firewalls

Circuit-level firewalls are very similar to application-level firewalls because they, too, are proxy servers with the difference being that the former do not require the use of client-dedicated applications, unlike the latter that require a dedicated proxy software for each service to be delivered (http, FTP, etc.).

In this case, in practice, the client and server communicate through the firewall at circuit level without affecting, in the communication, the firewall itself. Circuit-level firewalls protect the beginning of the transaction without interfering with the transactions already active. They are extremely useful because they ensure their service to a wide range of protocols. In contrast to the application-level firewall that requires a proxy server for each application, they are able, with only one

firewall, to provide a range of services without the dedicated software, integrating automatically what is already in place. In addition, they are easier to maintain, needing to intervene on only the one device.

5.3.5.4 Additional functionality of proxies

The proxy servers, unlike other systems, do not route any traffic and if properly configured, they deactivate all routing capabilities. Proxy servers replace and represent each system on each side of the firewall as the systems at the two ends never actually exchange direct information. The proxy is an integral part of the conversation to be certain that everything is proceeding securely.

As a proxy server must understand the application protocol used, it may also implement a specific security to the protocol.

It has already been stated that proxy servers are specific to applications and, in order to support a new protocol via proxy, the latter must be specially developed for that protocol. When selecting a proxy firewall, we must ensure that it supports all applications to be used.

There are many advantages arising from the use of a client proxy software. The first advantage is the ease of configuration because, as the client is designed to forward all requests of non-local data to the proxy, the only configuration information that we need is a valid IP address and a subnet mask, and the router and domain name servers (DNS) parameters may be neglected because such information must only be configured on the proxy.

Client proxies can also ensure a transparent authentication to validate connection attempts in output based on logon name and password. User authentication of outbound sessions is used to increase management and recording. If authentication is not being used, a firewall should refer to the source IP address to identify the subject that has accessed resources on the Internet. This can be a problem as a potential attacker, to change its identity, need only change its IP address. This event can be a serious problem in a Dynamic Host Configuration Protocol (DHCP) or bootp environment if we want to check all users.

Unfortunately, there are a number of disadvantages arising from the use of proxy servers. The first disadvantage is installation. In fact, if there are a large number of machines that need to use the proxy server, additional software must be installed on each of them, with considerable expenditure of time and economic resources, not taking into account the possible incompatibility of software, which may always be an issue. In addition, there may be further incompatibility resulting from different operating systems being used by different machines on the network. Furthermore, client software can also be a problem for users of portable computers that perhaps work on the local network at day and via an Internet Service Provider (ISP) in the evening. In this case, users must activate the proxy server during the day when operating on the local network and deactivate it in the evening when they operate via the ISP and this can become a source of possible business disruption.

In addition, a client proxy can pose a real problem if there are different network segments. In fact, the client proxy tries to forward all the non-local traffic to the proxy server and this is not an optimal solution if working with a network that is very wide and divided into subnets, because at every change in network, the file present on the proxy server must be updated and adjustments must be made to the computers that operate on the network portion affected by the change.

Not all proxies need special client software since some of them can act as transparent proxies, which means that all internal hosts are configured as if the proxy were a normal router directed towards the Internet. If a proxy firewall is considered to be the best choice for the needs of security, it is very important to ensure that the use in transparent or non-transparent mode has been defined. Since in most cases, these modes are not clearly explained, it is important to ensure that the proxy supports SOCKetS (SOCKS): in this case, it is not a transparent proxy.

Proxies are able to analyse the content of data packets and to take the necessary decisions. This is an extremely powerful feature that allows the network administrator to control with increased capillarity the network traffic based on the type of data that can be transmitted over the network. In most cases,

when it comes to content filtering, the first consideration is Java and ActiveX, which have already been addressed in Chapter 1, and which are further discussed below. Java applets and ActiveX controls can be taken from a remote server and executed on any compatible web browser. There can be very diverse functionality of these programs and there are few limits to the type of programs that can be created. However, there are some contraindications because, although they were languages that were developed with particular attention to security, bugs have been discovered and identified in them. To avoid security problems, many proxy firewalls are capable of filtering partially or totally the Java or ActiveX programming code, allowing users to access remote sites without the risk of performing dangerous applications for security.

5.3.6 Firewall architectures

It has already been said that when we intend to install a firewall, decisions must be taken about the traffic that the firewall must allow to pass and the traffic that must be block. It has already been seen that to do this, a simple screening router or proxy server software can be used that runs on a host. If we have more sophisticated needs, it is possible to provide for the use of both devices, using firewalls on both router and proxy server.

Essentially, there are three main architectures:

1. dual-homed host firewall;
2. screened host firewall;
3. screened subnet firewall.

Dual-homed host firewall is a very simple yet secure configuration. In practice, it uses a host that divides the lines between internal network and external network, using two distinct network cards. When this configuration is used, it is vital to disable the routing capabilities of the host in such a way that the computer cannot connect via the software, at the same time, to both the networks, breaching the security of the internal network. Its main disadvantage is the possibility, by any expert user, of enabling the internal routing of the host, thus negating the use of the firewall and the relative security. The structure of this firewall is shown in Figure 5.9.

The dual-homed host firewall operates by executing a series of application-level proxies or a series of circuit-level proxies. Since the host is equipped with two network cards, it can see both networks, by controlling the traffic between the same through a proxy software.

Since the routing inside the host must be totally disabled, all traffic must pass through a certain point within the same at the application level called checkpoints for due controls.

Particular attention should be paid to the disabling of internal routing, failing which results in the risk of nullification of the use of the firewall and its security. UNIX networks are particularly exposed from this point of view as certain UNIX variants are preset to activate routing functions and in this

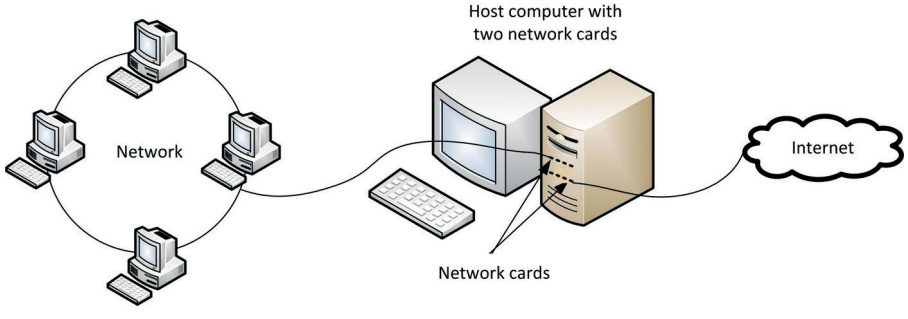


Figure 5.9 Configuration of a dual-homed host firewall.

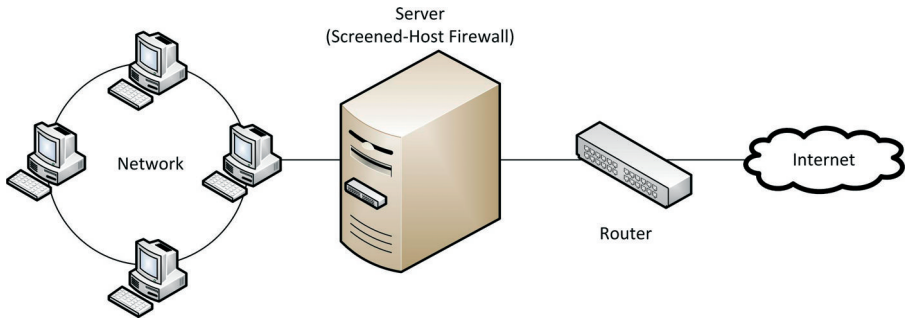


Figure 5.10 Configuration of a screened host firewall.

sense, it is necessary to intervene in a preventive capacity on the operating system to disable all the routing features of the same.

Screened host firewalls are considered safer than dual-homed host firewalls. In practice, they are composed of a screening router in direct contact with the external network and a host computer that serves as a further control and bridge between the screening router and the internal network (Figure 5.10).

In this way, we have a very simple, efficient and easy way to maintain configuration. The screening router must be configured to only see network host firewalls and users of the internal network must pass through this firewall host to access the external network. In this way, internal users have easy access to the external network, as they are enabled, while external users must pass the double check.

With regard to screened subnet firewalls, this represents an evolution of the previous configuration, further isolating the internal network from the external network. In this case, a further screening router is added within the network in order to monitor the internal traffic. This configuration represents an exceptional system of defence against attacks because the firewall isolates the host on a separate network, limits the result of a possible attack only to the host computer and reduces the possibility of attacks on the internal network. The internal router also reduces the chance of unauthorised access to the host computer.

5.3.7 Further types of firewalls

In addition to the subdivisions addressed above, firewalls are further divided into:

1. static filtering packet;
2. dynamic filtering packet;
3. filtering stateful.

5.3.7.1 Static filtering packet

The static filtering packet monitors traffic by using the information contained in the headers of data packets. When packets are received, the data attributes that are contained in the headers are compared with the criteria of access control contained in access control lists (ACLs). Depending on the comparison mode with ACL, the traffic is forwarded or blocked. A packet filter usually uses the following information to perform correct traffic control:

1. IP address or destination subnet.
2. IP address or origin subnet.
3. Service port destination.

4. Service port source.
5. Flag (TCP only).

Flag fields play a very important role in supporting the traffic of a static packet filter since a firewall is rarely programmed to block all the traffic that originates from a certain port or directed to a given host. The flags most used are:

1. ACK (Acknowledgement) that indicates that these data represent a response to a request for data and that there is information in the Acknowledgement Number.
2. FIN (Final) that indicates that the sending system wants to end the current session. Usually, each subject in a communication session transmits a FIN before terminating the connection.
3. PSH (Push) that prevents the sender placing the data in the queue before transmission. On the side of the receiver, Push instructs the remote system not to place the data in the queue and to pass on such information directly at the higher levels of the protocol.
4. RST (Reset) that restores the status of a current communication session. It is used when a transmission failure is not recoverable. This condition is typically caused by a host that does not respond.
5. SYN (Synchronise) that is used during initialisation of a communication session and must not be set in any other part of the communication process.
6. URG (Urgent) that is used to indicate that the transmitting of high-priority information characterised must be sent and that there is some useful information in the Urgent field. The receiver that receives this set flag processes the information before any pending data are queued.

TCP traffic is quite difficult to control and UDP traffic even more so as the latter provides less information about the status of our connection with respect to TCP and does not use the flag to indicate the status of a session. This means that there is no way to determine if a packet is a request for data or a response to a previous request. The only information that can be used to check the traffic is the number of the source port and by the number of the destination port. It should be emphasised, however, that such situations may not be useful in many cases because there are services that always use the same number of source and destination port. For example, when two DNS exchange information, they use 53 as source port and as destination port. Unlike many services, they do not use a response port greater than 1,023. This means that a static packet filter does not have any means for limiting one-way DNS traffic because if the incoming traffic is blocked on port 53, both the input and output data would be blocked. For this reason, the only way to check the UDP traffic by means of a static packet filter is to block the port.

The ICMP is used to provide a basic support for the IP protocol. It is not used to transmit user data but has management capabilities to ensure that everything will run smoothly. The ping command, for example, uses ICMP to verify that there is a connection between two hosts. ICMP does not use service ports and there is a type field that identifies the type of ICMP packet and a code field that provides more detailed information on the current session. It must be remembered that UDP does not use any flag field and for this reason is not able to indicate to the sender that the requested service is not available. To avoid this problem, ICMP is used to inform the sender.

Table 5.1 represents the various values of the type field.

Table 5.2 shows the valid codes that are used when the ICMP type is Destination Unreachable (type = 3).

Table 5.3 shows the valid codes that are used when the ICMP type is Redirect (type = 5).

If we are using firewall filters for the values of the type and code fields, it is possible to implement more capillary control with respect to the simple verification of source and destination IP addresses. Unfortunately, not all packet filters are able to implement the filtering operation using all type and code fields. Many, for example, filter type = 3, that is Destination Unreachable, without taking into

Table 5.1 Various values of the ICMP type field.

Type value	Name	Description
0	Echo Reply	Responds to an echo request
3	Destination Unreachable	Indicates that the host, the subnet or the destination service is not reachable
4	Source Quench	Indicates that the recipient or a routing device positioned along the routing is experiencing difficulties with input data flow control. Hosts that receive a Source Quench must lower their transmission rate to avoid the receiving system beginning to reject the data due to an overload at the entrance point.
5	Redirect	This informs a local host that there is another router device or gateway that is able to forward more appropriately data transmitted by the host. Redirect is sent by the local routers.
8	Echo	Requests that the system of interest provides an echo response. Echo is used to verify the existence of a connection between two hosts and to measure the response time.
9	Router Advertisement	Helps routers to identify themselves in a subnet. It is not a routing protocol, as no routing information is sent but is simply used to allow hosts of a subnet to be aware of the IP addresses of local routers
10	Router Selection	Allows a host to execute the request for router advertisement without having to wait for the next periodical update. This command is also called “router request”.
11	Time Exceeded	Informs the sender that the Time to live (TTL) value within the header of the packet has expired and that the same has not been able to reach the requested host.
12	Parameter Problem	This is a response that is sent to a transmitter when a problem occurs that is none of the other types of ICMP.
13	Timestamp	Used to measure the speed of a connection rather than that of the recipient. It is similar to an Echo request but in this case a faster response is supplied.
14	Timestamp Reply	This is a response to a Timestamp request.
15	Information Request	This command has been replaced by the use of bootp and DHCP It was used by configuring systems to discover our IP address.
16	Information Reply	This is a response to a request for information.
17	Address Mask Reply	Allows a host to make a request to the local subnet on which the most suitable subnet mask to be used is located. If no response is received, the host must have a subnet mask suitable for its class of addresses.
18	Address Mask Reply	This is a response to an address mask request
30	Traceroute	This is an effective way to trace a route from one IP host to another as an alternative to the previous Traceroute command. This option can only be used when all the intermediate routers have been programmed to recognise this ICMP command. Implementation takes place through the setting of a parameter using the ping command.

Table 5.2 Values of the code ICMP type 3 field.

Code	Name	Description
0	Net Unreachable	The destination network cannot be reached due to a routing error or an insufficient TTL value.
1	Host Unreachable	The destination host cannot be reached due to a routing error or an insufficient TTL value.
2	Protocol Unreachable	The destination host contacted does not offer the requested service. This code is usually sent from a host while all others are sent from routers along the path.
4	Fragmentation Needed and Don't Fragment Was Set	The data that we are trying to deliver need to traverse a network that uses smaller packets but where the "don't fragment" bit is set.
5	Source Route Failed	The transmitted packet specifies the routing that must be followed for the destination host but such information is incorrect.

account the code value and this can cause serious problems for communication, an example of which is shown in the following.

Suppose we have a network structure such as the one shown in Figure 5.11, where there is a local network with Token Ring topology and a remote network with Ethernet topology and where it is intended that the remote network can access the local Web server to receive update data. Suppose, further, that the router blocks ICMP input messages with destination unreachable. This block was designed to prevent denial of service (DoS) attacks (which will be described later), preventing anyone attacking from the outside from sending false host unreachable messages (type $= 3$, code $= 1$). Because the router is characterised by a limited packet filtering capacity, the ICMP traffic type $= 3$ must be blocked.

This block can generate problems because when the hosts on the remote network attempt to access the local network, these may not be able to display HTML pages. The problems that happen are indicated in the following:

1. The host browser on the Ethernet seems to be able to transform the name of the destination host into an IP address.
2. The browser seems to be able to connect with the destination server.
3. If the router provides a login session, the traffic does not seem to pass between the two connected systems.
4. The log on the local Web server indicates that the remote host is connected to the Web server and that several files have been transmitted.

Having said that, there are errors of transmission and the same has not been particularly successful. This is due to the fact that, by blocking all type 3 traffic, Fragmentation Needed (type $= 3$, code $= 4$) error messages are also blocked that prevent the router from adapting the mean transfer unit (MTU) of the traffic that is delivered. The MTU determines the maximum payload size that can be delivered by a data packet. In Ethernet, the MTU is 1.5 kB, whereas in Token Ring the MTU can reach 16 kB, and

Table 5.3 Values of the code ICMP type 5 field.

Code	Name	Description
0	Redirect Datagram for the Network or Subnet	This indicates that another router of the local subnet has a better routing to reach the destination subnet.
1	Redirect Datagram for the Host	This indicates that another router of the local subnet has a better routing to reach the destination host.

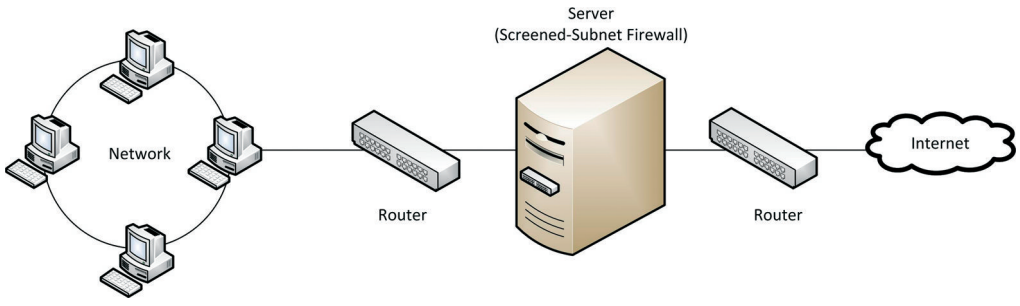


Figure 5.11 Configuration of a screened subnet firewall.

when a router receives packets that are too large for the destination network, it sends a request to the sender (ICMP type = 3, code = 4) to divide the data into smaller blocks. If the router tries to break up such data, certain problems may be encountered because its buffer could fill up. For this reason, it is much simpler to send smaller packets from the remote system.

1. The sequences of transmission and receipt are discussed below:
2. An Ethernet remote host sends an HTML data request.
3. The request is delivered to the destination Web server
4. The two systems perform a TCP three-packet handshake using 64-byte packets.
5. After the handshake, the Web server responds to the data request using an MTU of 16 kB.
6. The response reaches the router of the remote Ethernet network.
7. The router sends back a fragmentation request (ICMP type = 3, Code = 4) to the Web server requesting use of 1.5 kB MTU.
8. The remote router request reaches the Token Ring router.
9. The router checks its own ACL and sees that it must abandon all of the destination unreachable (ICMP type 3) packets and performs this operation.
10. The remote router request is not able to reach the local server and the remote host will never receive the requested pages.

From this, it can be seen that when using static packet filtering, we must ensure that we are aware, in detail, of the ramifications of the traffic that is being checked.

From what we have seen so far, it can be inferred that static packet filters are not particularly flexible devices and are not able to guarantee a high level of security against advanced attacks. They use a small amount of information to manage and control traffic. There are many routers on the market which are capable of performing static packet filtering.

5.3.7.2 The packet filtering dynamic

Regarding dynamic packet filtering, it goes beyond static filtering packet because it manages a connection table to check on the status of a communication connection session that is not based exclusively on flag settings. This characteristic allows a greater ability to monitor traffic.

If we used a static filtering packet, an attacker could send a data packet with a content designed to crash the receiving system. The attacker could perform targeted operations against packets to make them appear as a response to a request for information by an internal system. The filter, by analysing the packet, would check that the ACK part is set and would be misled because it would consider it a response to a request for data, allowing it to pass inwardly.

A dynamic packet filter would not be fooled so easily because, upon receiving the packet, it would refer to an internal connection table (state table) from which it could easily check that the response

packet received does not correspond to any request from the internal host within the network and the packet would be deleted.

Below, we illustrate the operation of a dynamic packet filter to show its better security features. For example, consider the configuration shown in Figure 5.13 where there is a static filter (Figure 5.12(a)) and a dynamic filter (Figure 5.12(b)).

The ACL of both firewalls should observe the following rules:

1. to allow the protected host to establish any service session with the remote server;
2. to allow the passage of any session already established;
3. to ignore all the rest of the traffic.

The first rule allows the protected host to establish connections to the remote server. This means that the only time in which a packet characterised by a set SYN can pass is when the source address comes from the protected host and the destination is the remote server. When this happens, any service on the remote server can be accessed (Figure 5.13).

The second rule is a general pass permit as if the traffic seems to belong to an already established session, the same is allowed to pass.

The third rule states that anything not falling under one of the two preceding rules must be deleted.

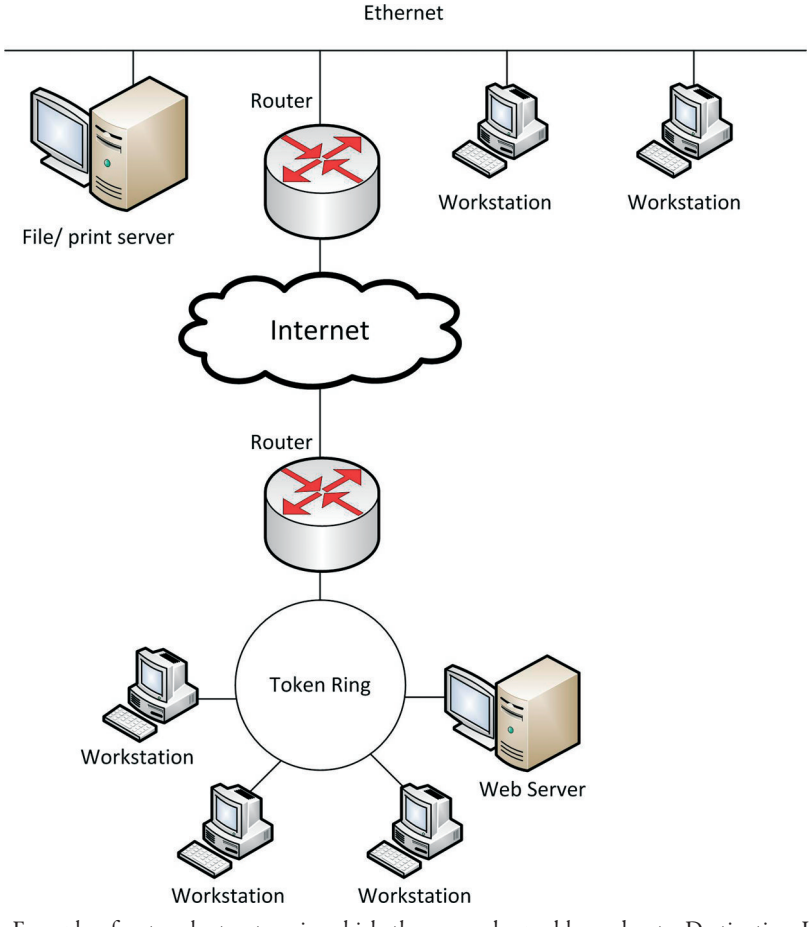


Figure 5.12 Example of network structure in which there may be problems due to Destination Unreachable messages block.

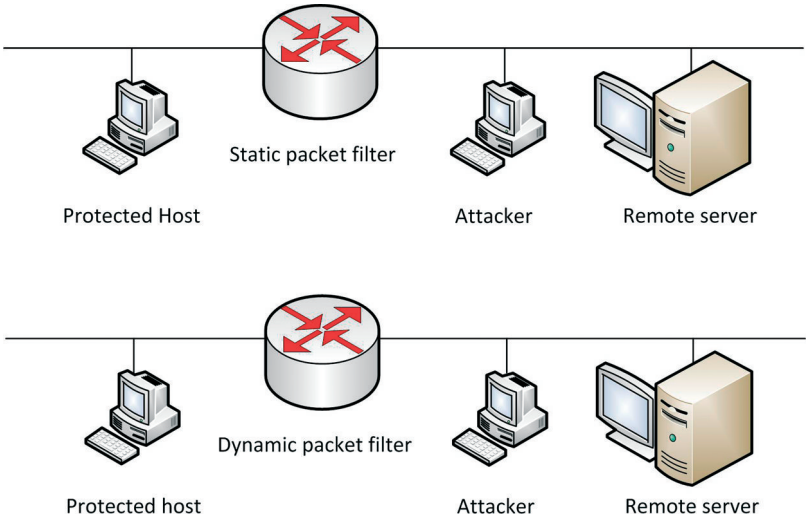


Figure 5.13 Example of static packet filter (top) and dynamic packet filter (below).

Both firewalls use the same ACL. The difference between the two cases is the amount of information available to control the traffic.

The following shows their behaviour in the case of traffic.

In setting a connection, the two systems behave as shown in Figure 5.14.

In this case, because the traffic meets the criteria for communication with the remote server, in both cases the firewalls allow it to pass.

After the handshake, the protected host sends a data request through a packet with ACK set and possibly also with PSH set. When the remote server receives a request, it will reply with ACK set and possibly also with PSH set. When the transfer is complete, the session closes and each system transmits a packet with FIN set. In this case, there are no problems with regard to the second rule of firewalls which allows the passage of any session already established. However, each of the firewalls performs this rule in a different manner because the static filter exclusively analyses the flag field to check if the SYN is the only bit set: since this is not true, this filter assumes that these data are part of a session already established and allows its passage.

The dynamic filter performs the same control, having created a heading in the state table when the connection was established for the first time. All the times that the remote server seeks to respond to the protected host, the table is consulted to ensure that the protected host has actually made a data request; the information on the source port matches the data request; the information at the destination port matches that of the data request. The dynamic filter can also check that the sequence and acceptance numbers match. When FIN packets are sent by each system, the entry in the state table is removed. In addition, if no response has been received for a configurable period of time between 1 min and 1 h, the firewall assumes that the remote server is no longer responding and deletes the entry from the table, helping to keep the state table updated. This is schematically represented in Figure 5.15.

Suppose, now, that an aggressor is observing that data flow and wants to attack the protected host. The first attempt he/she makes is a scan of the port of the protected system to check if there are any listening services. This type of operation is blocked by both firewalls as shown in Figure 5.16 because the packets of the initial scan are characterised by SYN set and by all other deactivated bits.

The attacker can make a second attempt trying to perform a FIN scan sending packets with ACK and FIN set to 1. In this case, the firewalls behave differently and the static filter allows this traffic to pass as it only checks that SYN is set to 1. The dynamic filter, however, acknowledges that SYN is not set and compares this traffic with the state table finding that the protected host has not set a

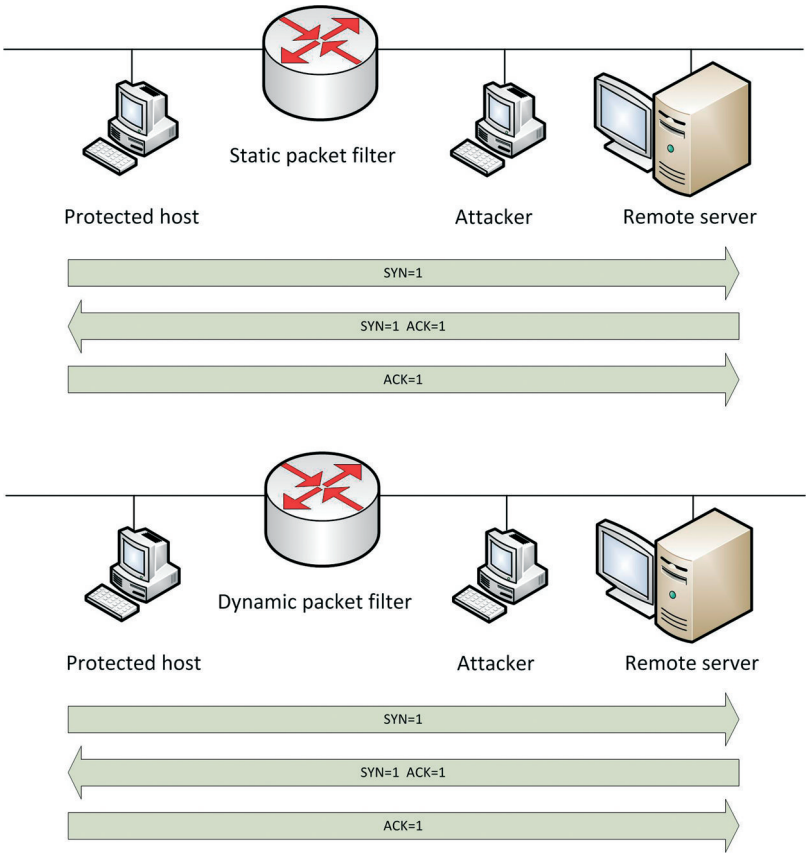


Figure 5.14 Establishment of a connection by a protected host.

communication session with the attacker and that there is no reason why the attacker is attempting to terminate a session if the protected host has not created it first. This situation is shown in Figure 5.17.

At this point, the attacker would try to pass the firewall trying to take the place of the remote server, but in order for that to happen, the following conditions should be in place:

1. The attacker should hide or take the IP address of the remote server.
2. If that address is taken, the attacker should act in such a way that the remote server is not able to respond alone to requests.
3. If the address has been masked, the attacker needs a technique to retrieve the responses from the cable.
4. The attacker needs to know the service ports of origin and destination used in such a way that its traffic is compatible with that contained in the state table.
5. Depending on the implementation, the numbers of acceptance and sequence should match.
6. The attacker should manipulate the communication session in a sufficiently rapid manner to avoid timeout both in the firewall and in the protected host.

It is clear that such an attack, albeit feasible, is very difficult due to the number of constraints.

It has already been seen that the dynamic filter has some issues with UDP traffic because the same does not contain information concerning the status of the connection. The dynamic filter, on the

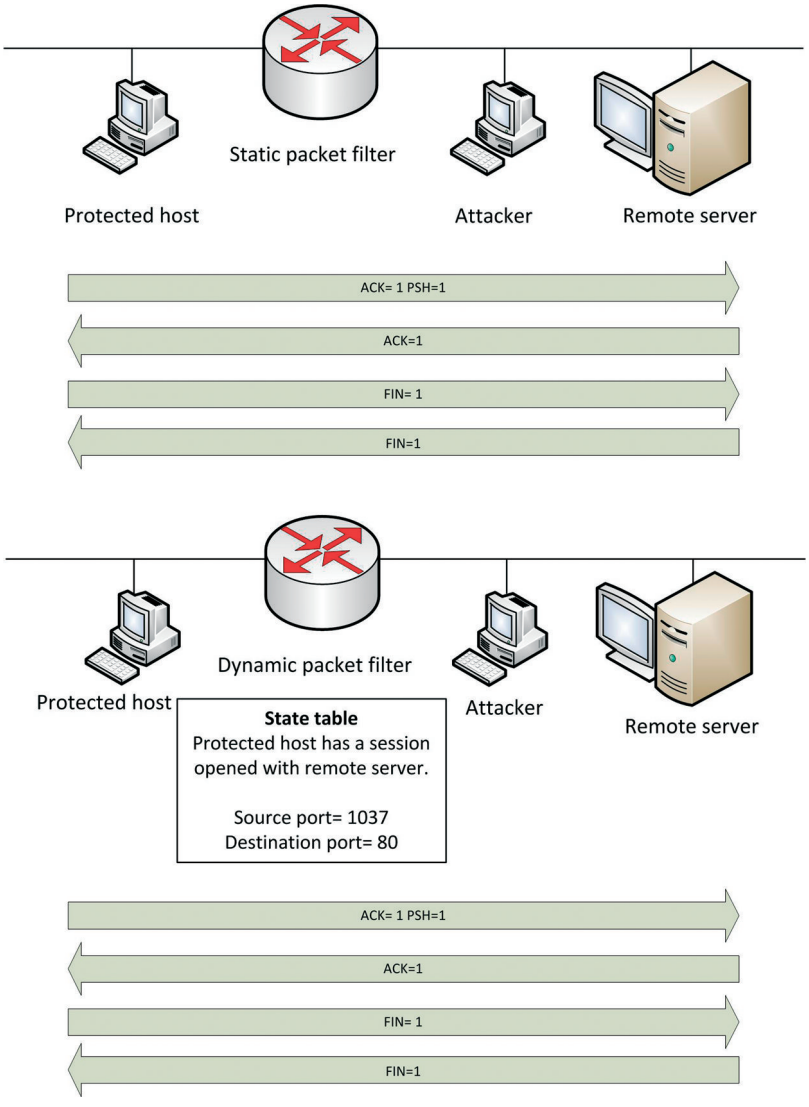


Figure 5.15 Session established between two hosts.

contrary, recalling the status information independently from the traffic, is able to operate with great efficacy against UDP traffic. For this reason, the use of dynamic filtering instead of the static filtering in the presence of UDP traffic is strongly recommended.

Implementation of the dynamic filtering is specific to transport. This means that it must be implemented specifically for each transport protocol such as TCP, UDP and ICMP. When selection of the dynamic packet filter is being made, it is very important to ensure that the firewall is able to check the status for all forms of transport that will be used.

From what we have seen so far, it can be inferred that dynamic filters are intelligent devices that make their decisions on the control of the traffic using the attributes of the packet and state tables. State tables allow the firewall to remember any prior communication packet exchanges and manage the traffic on the basis of this additional information. Their limit lies in their inability to make filtering

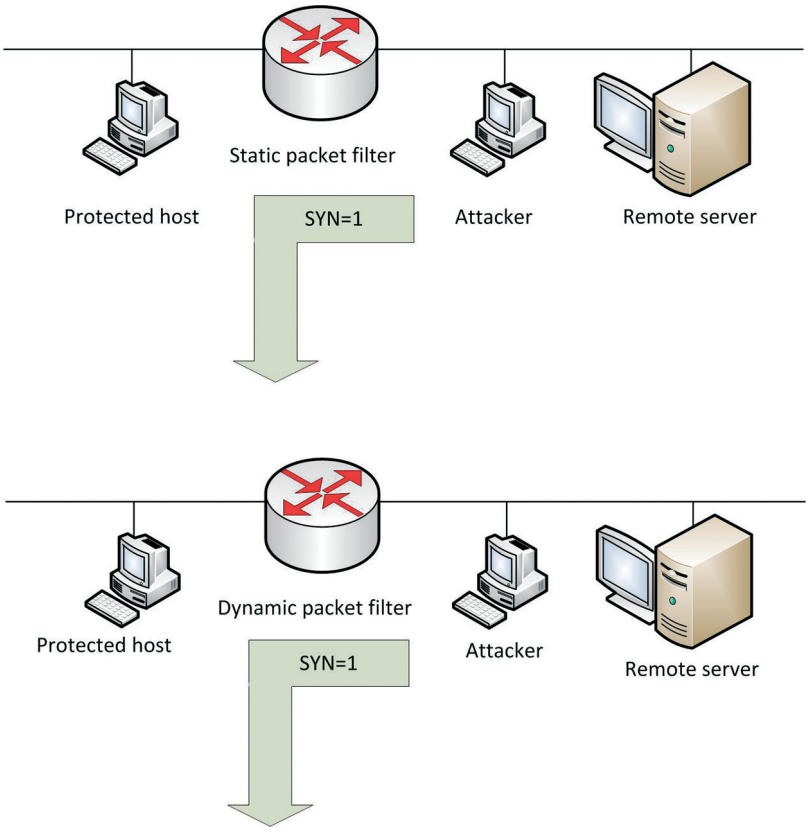


Figure 5.16 Block of a port scan by both firewalls.

decisions based exclusively on the payload of packets and, for this operating mode, proxy servers that have already been discussed must be used.

From what we have seen so far on dynamic filtering and proxy servers, it can be deduced that each has advantages and disadvantages. Dynamic filtering is generally easy to use with respect to proxy and adapts more easily to most organisational requirements but it is not sufficiently adapted to the verification of traffic as instead a proxy server is. Both are able to block traffic that is considered dangerous but can behave differently in the presence of suspect traffic. For example, consider the case where both a dynamic filter and a proxy receive a data packet with a high-priority flag set for a certain application and none of them has been programmed to manage this type of data. The dynamic filter would usually, but not always, allow this traffic to pass, while the proxy would block it. In addition, the proxy is also able to check the contents of the packets, providing greater security than the dynamic filter.

Therefore, proxies tend to be more secure but are not always adaptable to the specific needs of an organisation. The safest solution is, of course, to isolate the internal network with respect to the outside, but this, unfortunately, is almost never possible given the current needs of the majority of operating organisations.

Currently, there are products that incorporate both dynamic filtering technology and proxy technology, combining the advantages of both technologies in order to achieve the highest possible level of security.

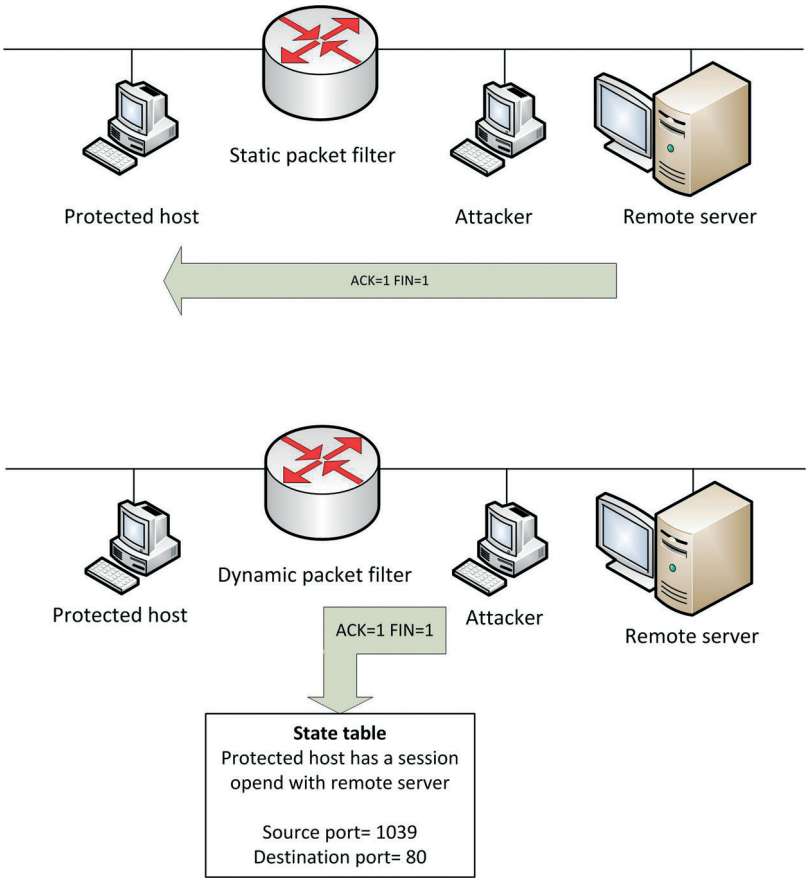


Figure 5.17 Effects of a FIN scan.

5.3.7.3 Stateful filtering

Stateful filtering is able to increase the efficiency of dynamic filtering. It was implemented for the first time under the name Stateful Multilevel Inspection, and the stateful rules are protocol specifications that control not only the context of a session but also its state. This allows filtering rules to differentiate between the different protocols without a connection such as UDP, Network File System (NFS) and RPC, which, precisely because of their nature, cannot be managed using a static filter and are not uniquely identified by a dynamic filter.

The major advantage of stateful filtering with respect to dynamic filtering is its ability to manage both the application status and connection status. The application status allows a user who has been previously authenticated to create new connections without requiring new permissions while the connection status controls the authorisation for the duration of a single session.

5.3.8 Firewall selection

The choice of a firewall platform specification is not an easy task because it is characterised by merits and defects that must be taken into consideration for individual purposes.

A distinction can be made between server-based firewalls and device-based firewalls.

A server-based firewall is an application that runs on an operating system.

A device-based firewall, or integrated solution, represents an application that is executed by the hardware and software of a particular manufacturer. By virtue of their integrated and dedicated nature, these devices are generally faster, more robust and considered more secure than server-based firewalls. The advantage of server-based firewalls is that they are able to provide additional options of configuration and support and are usually less expensive than device-based firewalls, also called integrated solutions.

It has already been stated that server-based firewalls represent applications that run on an operating system. The main firewalls of this type operate on the following operating systems: Macintosh, Unix, Linux, Microsoft Windows.

5.3.8.1 Macintosh-based firewalls

It is a common belief that the enforcement of a firewall on Mac ensures greater security because most hackers are not familiar with this technology. In addition, although weaknesses in applications running in Mac have been highlighted, many weak points in the operating system have not. The Mac is very easy to configure: being usable via a simple graphical interface and being provided with a reduced number of network services, its complexity is reduced to a minimum. A firewall based on this operating system is characterised by many of the benefits of performance, configuration and support tools.

However, there are certain contraindications due to the presence of vulnerabilities to the operating system, also known to hackers. In addition, since a Macintosh server is equipped with a limited number of configuration and application choices, administrators do not have many tools available to customise their own choices.

Since the majority of firewalls for Macintosh were designed to be a personal firewall and not to protect an entire network, this factor significantly limits the flexibility of this technology.

The issue of performance has also been present for some time because if the performance of the hardware increases and does not match by the software for which a Macintosh-based server is loaded and operates like a firewall, its performance may degrade significantly.

In addition, the operating system (OS X), because of its derivation from UNIX, has security hazards caused by daemon (services) that are installed by default. These aspects are discussed below.

5.3.8.2 UNIX-based firewalls

The UNIX operating system is characterised by a long existence, compared to other operating systems, including Microsoft Windows NT and derivatives (Windows 2000, etc.). Since UNIX is a quite dated operating system, the first firewalls were designed using the latter. This means that all the problems of this operating system are well known and firewalls based on it are very stable and functional. There are many versions of UNIX on the market, but nevertheless, it is considered an open system because many details relating to its facilities and its basic services are known. Most of the vulnerabilities do not refer to the core of the operating system but the services and applications that are performed by it. UNIX, compared to other operating systems, is characterised by its superior performance. This characteristic, together with its many applications and supported hardware configurations, makes it preferable due to the demanding applications that process a large amount of data. From an operational point of view, it is recommended that all the applications and components not needed for the proper functioning of the firewall are disabled and this operation is relatively easy to put into practice in the UNIX environment.

There are many advantages of UNIX. It has already been stated that it is very flexible and configurable, and is the largest operating system available on the market. There are many resources dedicated to its understanding and to the resolution of security issues that may occur. It can be run on different hardware platforms and multiprocessor systems, being able to support the huge amount of data necessary to manage the firewall that controls a large network. It does not require, for the most part, rebooting of the system as a result of changes in configuration, unlike systems based on Microsoft

Windows. Existing products on the market are directed more to UNIX compared to other operating systems, and this is a secure choice for most organisations that choose to manage their own firewalls.

However, there are also disadvantages of UNIX. Usually, problems occur when system administrators configure firewalls with typical installations and do not disable the programs and services (daemons) that may be vulnerable and are active due to the default settings. Furthermore, most daemons are configured to be run in the security context of the principal administrator, providing the potential attacker with full access to the system by exploiting this vulnerability. Deactivation of the daemon is an operation which is easy to implement: the administrator needs only eliminate or rename the scripts that trigger the daemon upon start-up of the computer or that should comment the line of the configuration file. UNIX is considered a system that is very difficult to understand and administer. Since there are several weak points known to all, the administrator must spend a lot of time on the security of the system to prevent a potential aggressor from exploiting these weaknesses, penetrating the system itself and its secure network.

5.3.8.3 Linux-based firewalls

Linux shares many of the advantages and disadvantages of UNIX and represents a significant challenge in the battle for the affirmation of operating systems. It represents a highly configurable platform, stable and well known, which has many security products. Its major advantage is its very open nature, an aspect welcomed by experts in the field of security, since the availability of source code to all allows it to be analysed in detail in search of vulnerabilities and security flaws. In addition, its open nature has allowed the spontaneous creation of numerous communities that provide all the necessary support around security issues.

The disadvantages of Linux are difficulties in learning about it, and by the many known vulnerabilities.

5.3.8.4 Firewalls based on Microsoft Windows

The great advantage of Microsoft Windows is its familiarity. There are several versions suitable for the management of firewalls (Windows NT, Windows 2000, etc.).

In the following, only Windows NT and Windows 2000 are discussed.

Windows NT is an extension of Windows Desktop and, as such, its environment is very familiar to most users who do not want to be familiar with a totally new environment for the mere configuration of a firewall, and, moreover, not forcing organisations to recruit dedicated staff. Systems based on NT are less expensive than those based on UNIX, and additionally the hardware itself, which is required to run this operating system, is less expensive. Its familiarity may also possibly increase the level of security as users will have less difficulty in maximising the system settings from the viewpoint of security. NT is also very useful because many organisations use it for other services and this allows greater standardisation of the platforms in use.

The biggest disadvantage is the fact that Microsoft is generally a very slow organisation and reluctant to admit any problems in its products and to correct them. Many flaws have been discovered but for the most part concern services that are not pre-installed and are not included in any firewall system. In addition, due to its proprietary nature, not much is known about the internal functions of the services that are not, however, configurable with the same immediacy as with UNIX. This situation can generate problems for security personnel seeking a more secure platform to install a firewall. Another disadvantage is the need to reboot NT servers after a configuration change or once updates have been made.

Windows 2000 shares many features with Windows NT including weaknesses. Its main advantage, like NT, is its familiarity that the average user has with these products. Windows 2000 is characterised

by a number of specific advantages over NT including the need to reboot servers after configuration changes have been made and the greater reliability of the server.

Regarding disadvantages, there are several vulnerabilities that have already been discovered, and which have been solved with patches such as that of the Telnet service that allows a hacker complete control of a Telnet administrative session, leading to exposure of the entire server.

5.3.8.5 Device-based firewalls

Device-based firewalls, also called integrated solutions, are based on dedicated hardware and software and owners and are composed of a special box containing all the electronics, complete with connectors and power supply. There are various devices on the market. They represent popular solutions and are very useful for small organisations that do not have sufficient resources to recruit dedicated staff for network security management and consequent firewalls.

The main advantage of integrated solutions is that they require less time for configuration as most of them are already preconfigured and ready to use: simply connect our internal network to a connector and the external network to the other connector and the firewall will start operating immediately and properly. Usually, any configurations are very simple to implement and are performed by means of a simple device connection Web page.

Since these firewalls use dedicated and programmable hardware, they are able to operate at very high speeds compared to server- and software-based firewalls. Dedicated design allows the reduction of costs of firewalls because there is no need to buy an operating system, or software and the relevant licences. These solutions in which everything is controlled, designed and supported by the manufacturers guarantee a high increase in security and minimise the use of dedicated personnel, reminding that simplicity is always a vital prerequisite for correct functionality of a security system.

The disadvantages are the limited flexibility of the devices themselves, almost everything being already provided during the manufacturing stage. This makes even the simple expansion of memory difficult if this has not been provided at the design and realisation stage. In addition, these devices are binding on the user to a single manufacturer and do not allow a certain modularity in assembly of the final product such as putting together the best components on the market.

5.3.9 Further firewall considerations

Regardless of the types of firewalls, there are functional characteristics common to all that are now examined.

The first group concerns the functionality of firewalls which can be summarised as translation of addresses, logging and analysis of the firewall and virtual private network (VPN).

The second group relates to management that can be summarised as detection and reaction to intrusions, integration and development, authentication and access control, and third-party tools.

With regard to the translation of addresses, this is considered a basic function of firewalls without which the firewall cannot be given much credit. Every time an IP address is converted from one value into another, reference is made to the translation of addresses. This functionality is implemented in most of the firewalls and is used when we wish to avoid communicating the real IP address of internal systems to remote systems. This feature has already been discussed in Chapter 1 and in this section will be further discussed from the perspective of security aspects. The situation is shown in Figure 5.18.

If an internal host wishes to access an external website, it sends a request, delivering the information to the gateway of reference, represented by the firewall. As the subnet on which the host is located uses private addressing, problems may arise.

Private addressing is the use of IP interval subnets that are used by any organisation to address its own internal host: this means that these intervals may not be used to access the Internet. Internal addresses may be used without any conflict, but this implies that any request sent to a remote system is

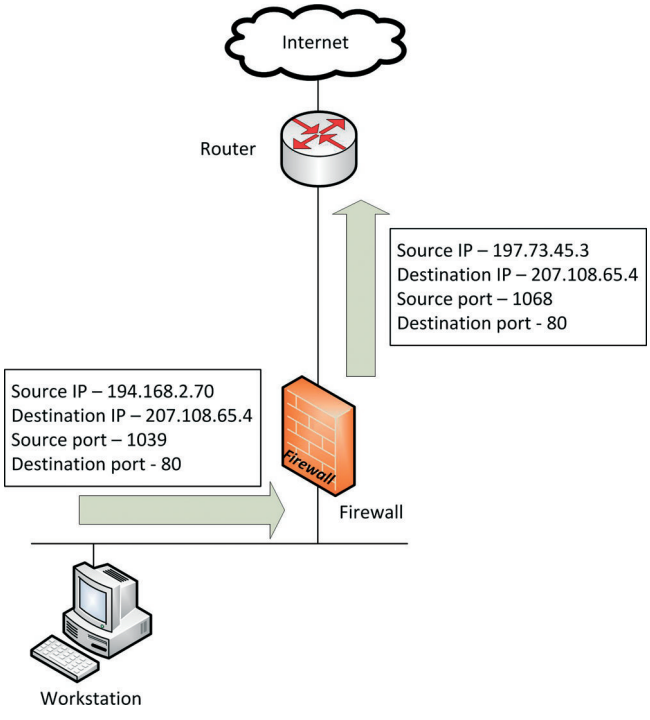


Figure 5.18 Translation of addresses.

not aware of the routing to be used to respond. These addresses are included in the following ranges: 10.0.0.0 to 10.255.255.255, 172.16.0.0 to 172.32.255.255 and 192.168.0.0 to 192.168.255.255. If, on the one hand, the internal host is able to reach the remote server, the remote server is not able to respond. To allow normal communication, use must be made of the translation of addresses that consists of mapping the IP address of the internal host onto another valid email address.

Below is shown how a firewall can distinguish between the responses directed to the host and the traffic destined for other hosts or the firewall itself and how the firewall tells the difference between different sessions. The source port represents a value that is dynamically assigned by the transmission system: this means that any value above 1,023 is considered acceptable and there are not, usually, problems with the firewall that changes this value for the necessary processing. In the same way in which the source port number can be used by systems to distinguish between different communication sessions, the firewall can use this source port number to check the responses to send to every internal system. The firewall changes the information in the IP header when it sends the packet to its final destination. Immediately after, it must also change the IP header in order to forward the data to the internal system. In the response packet, the destination IP address and the service port must be changed. This is because the remote server responds to the IP address and source port indicated by the firewall and, in this sense, the firewall itself must replace these values with those used by the host before sending the information.

There are three main methods of translation: Hiding Network Address Translation (hiding NAT), Static Network Address Translation (static NAT) and Port Address Translation (PAT).

With regard to Hiding NAT, all internal hosts are hidden behind a single IP address. This address may be the IP of the firewall itself or another valid email address. It is possible to use many hiding addresses for improved support, even if hiding NAT can support, in theory, thousands of sessions at the same time. The main disadvantage of hiding NAT is the impossibility of creating input sessions in that, as all systems are hidden behind a single address, the firewall has no method for defining the

internal system for which the session request from the outside is intended. Since there is no mapping to any internal host, all the requests for input sessions are blocked. This limitation may, however, be considered a feature that allows us to increase the security of the internal network. If the policy requires that internal users will not be able to run their own servers (Web, FTP, etc.) on the internal hosts, the use of hiding NAT is a method to ensure that it is not possible to directly access these services from the external network.

With regard to static NAT, this operates in a manner similar to hiding NAT with the difference that a single private IP address is mapped to each public address that is used. It is a useful feature when we are in the presence of an internal system that uses private IP addresses and we want to make the system accessible from the outside. Since each legal IP address is associated with a private internal IP address, the firewall has no difficulty in determining where to forward the traffic. Static NAT is very useful for services that have problems with hiding NAT such as communications between DNS servers, which require that the source port and destination are both conveyed to port 53. If we use hiding NAT, the firewall should change the port of origin into a random port number, interrupting the communication session. If, on the contrary, we use static NAT, the port number does not necessarily have to be amended and communication sessions can continue normally. The majority of NAT devices allow the simultaneous use of static NAT and hiding NAT.

PAT is used by the vast majority of proxy firewall devices. When it is used, all outbound traffic is sent to the external IP address used by the firewall, in a manner similar to hiding NAT, with the only difference being that the external address of the firewall must be used that cannot be set to another valid value. The way in which incoming traffic is handled varies from one device to another. In some cases, the ports are mapped to specific systems. In cases where there is a large environment, this can represent a limitation that, to be avoided, requires the use of proxy servers able to analyse the content of the data to support multiple internal services. If there are internal services that perform the same service, it is very important to be certain that the firewall is able to distinguish the same. In cases where this is not possible, the server must be placed outside the firewall.

The primary function of a firewall is, of course, the ability to monitor all traffic that passes through it. However, a secondary function of no lesser importance is the ability to document and analyse all traffic that passes through it. In this sense, logging (registration) is very important because it can leave a trace of the identity of the subjects that pass through the firewall or that are attempting to do so. Analysis represents a very important factor that should always be taken into consideration since it allows the recognition of unauthorised access attempts that can anticipate a future attack. The main features that a good log should have are:

1. presentation of all the entries by means of a plaintext and easy to read format;
2. viewing of all of the entries in a single log in such a way as to accurately identify traffic patterns;
3. clear indication of blocked traffic and what is allowed to pass;
4. possibility of manipulation of the log in order to highlight particular types of traffic;
5. impossibility of overwriting or failure to record due to limitation of the internal memory;
6. possibility of viewing the logs from remote locations;
7. ability to export the log in easy interpretation format such as ASCII files. This feature allows the data to be further processed using programs of reports, spreadsheets, databases, etc.

All the above features are important to ensure a certain level of network security. If the log is analysed periodically, it is possible, in most cases, to locate an attack before it occurs.

VPNs are a characteristic distinguished by a high-end-type firewall. VPNs allow authenticated and encrypted access to an internal network via the external network, typically the Internet. This means that, instead of using an expensive dedicated point-to-point network, users can use the more convenient and less expensive Internet to contact internal hosts of their organisation. However, it is necessary to determine which options of configuration, control and management must provide the

firewall of interest for the VPN in question. In most cases, the best solution is a dedicated VPN that integrates with the firewall.

Very important is the ability of a firewall to advise the network administrator when an attack is occurring, in order to correctly select the same firewall. In the case of high-profile attacks of the type DoS, which is further discussed in the following, it is very important to use firewalls that are able to report this type of attack instantaneously, to allow the rapid restoration of sites attacked.

Firewall systems of the future will almost certainly have to cope with heavy and widespread attacks and be able to automatically reset the network. These systems should generate automated and standardised reports to provide a sufficient level of information in order to allow defence actions.

Firewalls tend to integrate increasingly with other network systems and services, greatly simplifying the complexity of management and operating costs as the firewalls themselves need not duplicate existing network infrastructures. The integrations include directory and authentication services that allow us to eliminate unnecessary information from the user account and create diagrams of customisable authentication. Two reference diagrams are Lightweight Directory Access Protocol (LDAP) and Remote Authentication Dial In User Service (RADIUS).

LDAP creates a tunnel between two directory services or between a directory service and a client. From the viewpoint of firewalls, this means that instead of creating double user and group accounts, the system can use accounts and properties already present in a directory service from other sources to allow access. This factor generates a series of advantages, reducing the management overhead of creating and managing double user and group accounts, while reducing the complexity that is always a vulnerability with regard to the security aspects.

RADIUS is a platform of extensible and independent authentication. It allows the presence of custom authentication schemes, through, for example, biometric devices or smart cards, and download from the firewall of the actual authentication work. In this sense, it makes the authentication process easier and stronger, providing an infrastructure targeted exclusively at authentication.

Most modern networks are composed of various devices produced by different manufacturers and can become extremely complex to administer. However, there are solutions that offer an excellent management capacity in the following areas: applications, availability, networks, performance, services, systems, memory and data. To select a proper firewall, we must ensure that the same is able to operate with different management tools.

5.3.10 Location of firewalls

Once the firewall has been correctly selected, following the description above, we need to know where it should be placed correctly within a network. The most obvious choice is to place it at the input/output of the internal network but this, in some cases, can lead to problems of communication for the reasons mentioned above. The best location, according to common opinion, is the one shown in Figure 5.19.

As can be seen in Figure 5.19, all the internal systems are protected by the firewall from attacks from the Internet and remote sites connected with the organisation through the WAN link are also protected. The systems accessible from the Internet, such as the Web server and the relay of emails, are suitably isolated within their own subnet connected directly with the Internet via the router. This subnet is called demilitarized zone (DMZ); in fact, even if the same is safe from attack, it can never be so with absolute certainty because, in such systems, incoming connections are permitted. The use of a DMZ ensures further protection against attacks: since some input services are open, attackers may be able to obtain high-level-type access to such systems, reducing the likelihood of internal systems being attacked as they are isolated from the rest of the network.

It is also possible to add additional network cards to the firewall to check all types of remote access. To further improve security, we need to implement the functionality of static filtering of router

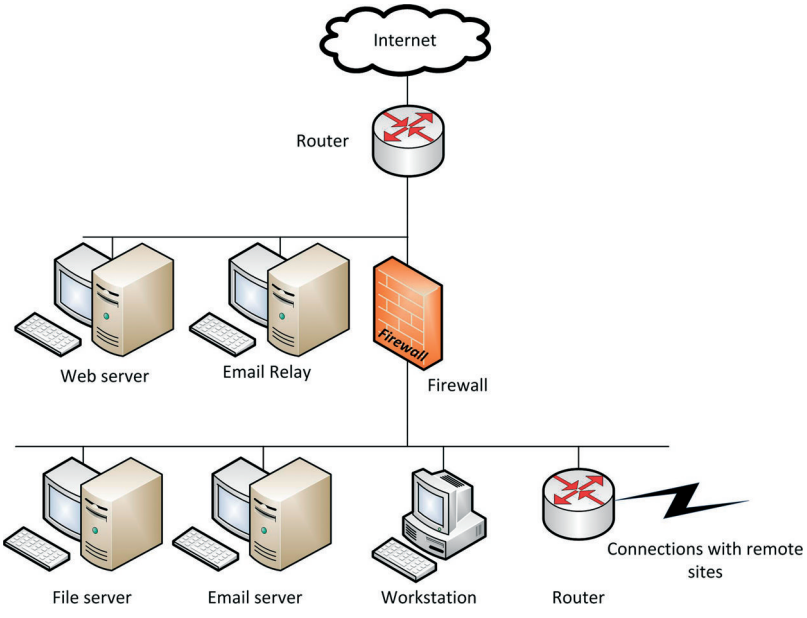


Figure 5.19 Correct positioning of a firewall.

packets, providing a security perimeter at multiple levels around the same network. In this case, if a malfunction becomes evident in one of the security devices, the second device can intervene to compensate for such malfunctions.

There can be many variations that are applicable to this basic scheme. For example, an additional firewall configuration can be added in order to increase the level of security. If this firewall is a dynamic packet filter, a proxy firewall can be placed behind it in order to increase the security of the connection to the external network.

5.3.11 Network security assessments

The US government, in order to better understand the importance of firewalls and network security in general, has produced a number of publications called *Green Book, Yellow Book* and *Orange Book*. These publications represent important references to focus the problems of its network security.

Until a few years ago, network experts used as classification what was indicated in the *Orange Book* where, for example, a class D network showed no security mechanism. A class A network, on the other hand, to be such, had to be characterised by a certain number of policies and security mechanisms.

The four security assessments as mentioned in the *Orange Book* are discussed below.

5.3.11.1 Level D

The lowest level is D, which offers no protection for files or users. The classic case is a local open network that is connected to the Internet.

5.3.11.2 Level C

At the top level, we find the C class, which is divided into subclasses that offer discretionary protection and ensure audit features that allow the activities of users to be followed. Subclasses are C1 and C2.

Subclass C1 offers a form of separation between users and data. A C1 system provides a form of control that can allow access restrictions on an individual basis. In practice, a C1 system allows users to protect private information and various data, preventing other users from reading or damaging these data. In a C1 system, all users can operate on the data at the same level of confidentiality. It should be emphasised that in this class, all computers connected to the network are characterised by a class D1 level of security because the computers themselves are subject to attacks. The minimum requirements for cataloguing a system such as class C1 are:

1. access defined and controlled between users and objects;
2. presence of a system of identification and password that users must comply with in order to be able to access network information.

Subclass C2 is represented by systems always equipped with discretionary control but characterised by a greater precision with respect to those systems provided by C1 class. Users of a C2 system are identifiable one by one for each action that they perform on the network. C2 systems ensure such control via login procedures, auditing of security-related events and through the isolation of resources. Systems classified as C2, in addition to the requirements held by class C1 systems, must meet the following requirements:

1. Extended control of access to groups of users and individual users.
2. Access control able to limit the replication of access rights.
3. Discretionary access control mechanism able to guarantee particular protection of specific objects against unauthorised access and at the request of users or as a default setting.
4. Access control to be able to allow or restrict access to certain objects by certain users.
5. Identification system able to uniquely identify each user connected to the network.
6. Operating system that is able to associate to each user of the network their actions.
7. The network's capacity to create and manage recording of all access to objects of the network.

5.3.11.3 Level B

Class B is divided into classes B1, B2 and B3. The difference between levels B and C is the mandatory nature of the presence of security tools, meaning with the latter term the fact that each level of access to the system must provide access rules. In practice, each object must be associated with a level of security and the system should not allow a user to save an object without first specifying the level of security.

Class B1 systems must be in accordance with that provided for class C2 systems and must comply with the following additional points:

1. Confidentiality label support for each subject under the control of the network. A confidentiality label is an indicator of the level of confidentiality of a given object.
2. Use of confidentiality labels as the basis for all the mandatory decisions of access control.
3. Identification of each object in input to the system by means of confidentiality labels and refusal to grant access to objects without a label;
4. Accurate representation by labels of the confidentiality of security levels associated with the objects.
5. Need for designation of each communication channel and each I/O device every time that the system administrator creates a system or adds new channels of communication. In addition, the administrator can only modify these designations manually.
6. Management, by the multilevel devices, of confidentiality labels of all the information that the network transmits to the device.
7. Lack of management, by single-level devices, of confidentiality labels for the information transmitted.
8. Generation, by direct information to users (display on monitors or printouts on printers), of labels indicating the confidentiality of the objects produced.

9. Use, by the system, of password and identification code for determination of security and the levels of user access. The system must also apply security and access levels to each object which the user tries to access.
10. Recording, by the system, within an audit trail, of every attempt to gain unauthorised access.

Class B2 systems must comply with all the specifications of class B1 systems. In addition, the system administrator must develop a network based on a security policy that is well defined and documented. Specifically, class B2 security model must ensure the discretionary and mandatory access control criteria of a B1 system to include all subjects and objects. A class B2 system is able to address all the elements, even those not visible, and is divided into critical and non-critical elements. It is also able to resist attacks. Class B systems are characterised by the following minimum requirements:

1. Immediate notification, for each user, of any changes in the level of security that the system assigns a user during each session.
2. Support for secure communication between the user and the system for login and initial authentication.
3. Research, by the system developer, aimed at hiding the storage channels and determination of the maximum bandwidth of each channel identified.
4. Support, by the system, of distinct functions of operator and administrator.
5. Identification of a configuration manager to ensure that the competent authorities approve every change in the design of the system starting from the upper level specifications.

Class B3 systems should be characterised by the same security requirements of class B2 systems. In addition, the authorisation of each access must be tamper-proof and small enough to be subject to analysis and testing. The system must exclude all codes that are not essential to ensure the security policy. The system must be characterised by reduced complexity in order to make the analysis easy. It must provide for a full-time administrator, extended auditing mechanisms and a system recovery procedure. Systems belonging to this class are characterised by high resistance to attack. The minimum requirements are as follows.

1. Generation of a list security of easy consultation, as well as control of access to an object by individual users, as provided in class B2 systems.
2. Creation of a list, for each object, which specifies which users have access to the same object.
3. Identification of the user before executing any action.
4. Identification of any user, not only internally but also through the use of external security protocols. The system does not allow access from the outside to users using the classification of external security systems, even if the internal profile of the user is correct. The system must also generate a record of every attempt at access rejected.
5. Logical isolation and distinction of all the secure communications with respect to all the other paths.
6. Creation of an audit trail information complete for each of the activities performed by each user on each object due to the secure communications base. The basis for secure communications generates complete audit trail information that can be analysed by the security administrator every time a user attempts to perform an activity.
7. Distinct function support for the security manager.
8. Safe recovery support: rebooting of the system must take place without any security risk.

5.3.11.4 Level A

Level A is the highest level. It contains only one security class, that is A1.

Class A1 systems are functionally identical to those of class B3 because this classification does not in fact have specific characteristics of security architectures or policies. The difference lies in the fact

that designers must analyse the system on the basis of formal project specifications. Specifically, a class A1 system must comply with the following points:

1. Receipt by the security manager of the formal model of security policies produced by the system developers. This model accurately identifies those policies.
2. Presence of a security officer.
3. Installation of the system by the security manager, formally documenting every step of the installation and demonstration that the installation complies with the security policies and the formal model.
4. Auditing of the operating system that can help to determine the level of damage that an intruder can cause. Auditing can also help to determine if an intruder is in the process of attacking the system or if the same has compromised the operation of the system. The audit trail records, moreover, all the changes that are made to secure files.
5. Presence of a compulsory access policy that can be, for example, the same policy provided for class B1 systems, which ensures that only certain users can have access to system files. Regular firewall users and most daemon processes can operate at a lower level of security, avoiding the same being able to modify the critical files of the operating system. The implementation of a firewall, without an operating system that achieves at least class C2, renders all the attempts to ensure a certain level of system security futile. Without these requirements, it is not in fact possible to understand if our network is under attack. In fact, the messages produced by TCP/IP proxies refer only to TCP/IP events and the fact of being able to follow these events may be of help for the detection of an attacker, but the management of an audit trail for operating system level activities may prove to be of greater utility to identify attacks against firewalls.

5.4 The S-HTTP protocol

The HTTP protocol, which has already been mentioned in Chapter 1, was developed for transferring multimedia information, graphics, audio, video, etc., and it was not expected that it could become the basis for a large number of commercial transactions on the Internet. It was soon realised that the protocol did not offer sufficient guarantees of security. Essentially the level of security used was suitable for normal applications on the network but was not certain for the more delicate business operations. To meet the demands of increased security on the Internet, in 1994, the Internet Engineering Task Force (IETF) produced a new proposal, represented by the Secure-HTTP (S-HTTP), which was subsequently developed in 1994 by Enterprise Integration Technologies (EIT).

It is characterised by the following points:

1. Extension of the set of instructions of HTTP to allow secure encrypted transactions.
2. Secure transmissions due to the use of headers and encryption according to HTTP.
3. Security of transactions using a signature method, an encryption method, verification of the identity of the sender and the authenticity of the message.
4. Transmission encryption using a system of symmetric key cryptography and a system of asymmetric key cryptography.
5. Support of the use of certificates and signatures.
6. Point-to-point encrypted transmissions support.

Up until a short time ago S-HTTP had an important role in the evolution of the Internet but, currently, it has been undermined by the Secure Socket Layer (SSL) protocol, which is further discussed below. S-HTTP is mainly aimed at the execution of secure operations on the Internet, whereas SSL is more suitable for a wide range of applications over the Internet (FTP, HTTP, Internet

Relay Chat (IRC), etc.). Both S-HTTP and SSL support private keys and public keys. Even if many users are more familiar with the use of S-HTTP, most users use SSL.

5.4.1 Introduction to S-HTTP

It has already been stated that S-HTTP is a version of HTTP that supports security features, including encryption of Web documents sent over the Internet and digital signature support. This protocol allows the client's web browser to verify the integrity of Web messages using the message authentication code (MAC).

It is able to offer HTTP transactions by ensuring the necessary security. The methods that are used to ensure the security of messages include signature, encryption, verification of the identity of the sender and authenticity of the message. Its method of working is shown in Figure 5.20.

HTTP messages are composed of two basic elements: the header, which specifies to the recipient how to process the message, and the body of the message itself. In a similar manner, an S-HTTP message is composed of a header and a body, encrypted, that contains the message. The header may include, among other aspects, information about the manner by which the recipient must process the body of the message after deciphering. The sender of the message can be both the client and the server.

The following will demonstrate the typical sequence of creation of an S-HTTP message such as, for example, the secure reply to the request of a client:

1. The server captures the message in plain text that it must send to the client, recovering it, generally, from local hard disk. The pure text message can be an HTTP message or by any other object. As S-HTTP conveys the message, including headers, in a transparent way, the software that sends the message can be any version of HTTP.
2. The server processes the cryptographic requests of the client and all the material related to the keys that the client has sent to the server during the procedure of initial connection. The server therefore uses the encryption system indicated by the client. The encryption method that uses the server depends on how the server administrator has configured it.
3. The server processes its own cryptographic preferences and the material concerning the keys.

The server, in order to generate an S-HTTP message, integrates its own security preferences with preferences received from the client. Using the cryptographic method supported by both the client and the server, the server itself encrypts and inserts the plain text message within the S-HTTP message. The server then sends the message to the client, in a manner similar to a normal response of an HTTP operation. The client, having received the message, decodes the message.

As S-HTTP uses encrypted communications, the client and server must, of course, agree, at an early stage, on an encryption key. In practice, the server and client must exchange in some way a key, ensuring that this key is not intercepted by a hacker. The typical sequence of key exchange takes place according to the following steps:

1. The client requests a secure page from the server. Within the request, the supported encryption schemas are listed.

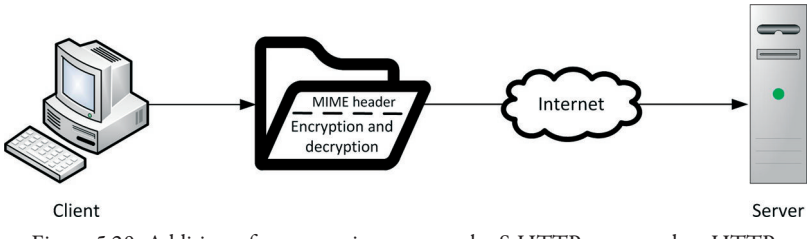


Figure 5.20 Addition of new security measures by S-HTTP compared to HTTP.

2. The server responds to the message of the client sending a list of supported schemas together with its public key.
3. The client responds to the server with a message encrypted using the server's public key. The message also contains a session key or a value that can be used by the server to generate a session key.
4. The client and the server start to communicate using an agreed encryption schema.

The client, in order to decipher the S-HTTP message, uses the session key. After the client has found a match in the encryption standards indicated in the header of the message, it decrypts the message using a combination of the keys of the sender and of the recipient. The client, after having decrypted the message, displays the HTTP content.

The server, in a normal HTTP transaction, after having transmitted the HTTP data to the client, should close the connection. On the contrary, in an S-HTTP transaction, the server does not terminate the connection until this operation is requested by the client's browser; if the latter does not send such a request, the connection will remain active and the encryption session key also remains valid.

It has already been seen in Chapter 2 that in a symmetric system, the subject transmitting and the receiving subject encrypt and decrypt the message using the same key. In the S-HTTP protocol, only the client and the server know the key, greatly increasing the level of security of the communications that occur between them. In Chapter 2, it has been seen that in an asymmetric system, two different keys are provided: one that is public, known by all, and a private one associated with the public key and known only by the owner. In a public cryptography system, the sender, in order to send an encrypted message, encodes the message with the recipient's public key that can decode using its private key associated with the public key.

In symmetric key systems, users must be sure to exchange the key securely, in such a way that nobody can intercept it, failure to do so risking the total loss of security of successive coded communications. S-HTTP uses two procedures for the transfer of keys: public bandwidth key exchange and use of externally managed keys. In the first case, the server encodes the private key using the public key of the client and sends the corresponding encoded message to the client. In the second case, the client and the server exchange the private key manually. The latter method is difficult to achieve and for this reason, the second method is the most popular method, which is preferred for corporate intranets and in some case at banking sites.

With regard to the bandwidth exchange, the S-HTTP transaction operates according to the following steps:

1. When the client visits the S-HTTP Web site, it sends the relevant server a connection request that initialises the transaction to which the server responds with an appropriate message in the case of a positive outcome.
2. Once the positive message has been received from the server, the client sends the server its own public key in plain text and the systems of encryption used by the client to generate the public key.
3. The server receives the client's public key and validates the acceptable encryption systems in its list of keys in order to check whether it is able to process the client's public key. In the event of a positive outcome, the server sends the client the session key that the server itself encodes using the client's public key. If, on the contrary, the server is not able to accept the cryptography system suggested by the client, it closes the connection.
4. Once a secure connection with the server has been established, the client browser encodes every communication directed towards the server using the session key and vice versa.

In addition to the encryption of the message, both the systems can verify the integrity of the messages and the authenticity of the sender by using the code of the message itself. S-HTTP calculates the code of the message as a hash key of the document using the session key that the client and the server exchange before initiating communication. In addition, both the client and the server may sign,

with a digital signature, the initial exchange to give each other methods of identity verification. If one of the two parties requires verification of the messages, the protocol adds the code of the message to each transmission, that is the digital signature. Because only the parties concerned have access to the session key, since the same was generated at the time (and therefore never communicated before) and transmitted in encrypted mode, both the subjects can check if the message has been modified during transmission. The protocol supports the digital signature using additional headers.

S-HTTP is compatible with HTTP, supporting all its commands. This is possible because S-HTTP encapsulates the HTTP commands in the encrypted message. S-HTTP extends the HTTP model to the bare minimum in order to add the functionalities of encryption and digital signature.

S-HTTP was designed to support different cryptographic standards including privacy enhanced mail (PEM), Pretty Good Privacy(PGP) and Public Key Cryptography Standard 7 (PKCS-7). Browsers and servers that are not compatible with S-HTTP can communicate with browsers and S-HTTP servers without the user realising it, except when the exchange of protected documents is prompted. The S-HTTP protocol does not require the existence of public keys of the client, and this means that users do not need to have public keys in order to participate in secure transactions. With regard to alerts, in most browsers, a key symbol usually appears (usually at the bottom of the browser) to indicate that it is within a secure connection or, however, the word "https" appears on the command line.

S-HTTP is able to provide security services for operations in general for e-commerce applications that require confidential transactions, authentication and message integrity. It supports secure transactions by incorporating extensions of cryptography at the level of the application. These extensions are in contrast with the authorisation mechanisms of HTTP. In fact, the HTTP protocol requires that the client tries to access the server and the server blocks the client until the server is able to use the security mechanisms. S-HTTP, on the other hand, activates the transactions of authentication and security starting from the initial request using systems of public key cryptography, passwords or systems based on Kerberos.

It has already been stated that S-HTTP is able to support a discrete number of security mechanisms for both the client and the server. EIT has provided various security methods to ensure the support of different Web applications. S-HTTP also provides symmetrical features both to the client and to the server, while preserving the transaction model and the characteristics of implementation of the HTTP protocol. S-HTTP is able to vary its process of cryptography depending on the client requests. In an S-HTTP client or server, it is possible to incorporate different standards of encryption of messages, including the PKCS-7 and PEM systems.

S-HTTP does not require public key certificates of the client as it is also able to support symmetrical session key work modes. S-HTTP session key modes allow private transactions to be performed without users being able to possess a public key. In addition, even if this protocol supports the standard digital certificates available on the Web, it does not require the use of these certificates.

S-HTTP is able to support secure point-to-point transactions unlike the authorisation mechanism of original HTTP that requires that the client tries to access the server and the server blocks the client until the same uses secure mechanisms. The server can ask the client to initiate a secure transaction. The server can use the initial messages of the client to support the creation of HTML modules without encryption and making the server secure.

It has already been said that S-HTTP is characterised by a degree of flexibility with regard to the algorithms, the modes and parameters of cryptography. It allows clients and servers to agree on a mode of secure transaction, a cryptography algorithm and a selection of the certificate. It does not apply a certain model of confidence even in the case where both of the parties involved in the communication are equipped with various public key certificates.

S-HTTP ensures the protection of the message in three ways:

1. via digital signature;

2. via message authentication;
3. via message encryption.

In this sense, it is possible to sign, authenticate and encrypt a message, or use a combination of the three methods described above. S-HTTP also supports the mechanisms for the management of multiple keys, including the secret password, the exchange of public keys and the distribution of Kerberos tickets. In particular, it ensures a preset symmetrical session, in a previous transaction, of session keys in such a way as to send private messages without having to use a pair of keys. In addition, it supports a mechanism that allows the communicating parties to check if the transaction is recent.

5.4.2 Digital signatures in S-HTTP

Digital signatures in S-HTTP use the signed data of PKCS-7 type. If the server asks for a digital signature, the client may attach the right certificate to the message or wait for the server to receive such certificate autonomously. It allows, explicitly, the use of certificates signed with the private component corresponding to that stated by the public component: in practice, a signed and personally verified certificate and not produced by a dedicated certification entity. This type of certificate is not verified. The importance that a server attaches to this type of certificate can vary from server to server. In every way, all signed messages are compliant with the standard PKCS-7.

The S-HTTP server encrypts all the transmissions that it sends to the client. If several clients connect to a server simultaneously, it is able to process and decode all the requests that it receives. It should be remembered that public key cryptography requires a processing time greater than private key cryptography, precisely because of the greater number of mathematical calculations that must be followed. To optimise the response times, it has already been stated that S-HTTP uses two mechanisms for the transmission of keys:

1. public bandwidth key;
2. externally supplied key.

With regard to the public bandwidth key, the sender encodes the symmetric session key by means of the server's public key and sends the result to the same server that decodes the message using the private key associated with the public key, to obtain the session key by which to encrypt transmissions addressed to the client.

With regard to externally provided keys, the sender encrypts the message using a session key exchanged beforehand and securely with the recipient. Users can also extract keys from Kerberos tickets, which is further discussed below.

It has already been stated that S-HTTP also supports verification of the integrity of the message and authentication of the sender. It also calculates the digital signature using a technique called MAC by means of a hash function applied to the document using a shared secret code. MAC can be created using manual distribution or through Kerberos. The technology for calculating the MAC does not require that both parties use public key cryptography. MAC is also useful for the mutual identification of two subjects in a reliable manner in a transaction without providing a code generated by a certification authority.

The S-HTTP protocol supports a verification mechanism of the degree of timeliness of transmissions. The S-HTTP server uses a mechanism for verification of the response to be certain that a transmission is genuine and has not been tampered with by a hacker. This mechanism prevents an attacker from copying a packet, modifying it, breaching the cryptography and inserting it back into the network.

An S-HTTP message is composed of a request, or status line (similar to HTTP), by a series of headers and encapsulated content. Once the recipient decrypts the encapsulated content, it may be another S-HTTP message, by an HTTP message or by simple data. To be certain that S-HTTP is

compatible with the HTTP implementations in place, S-HTTP requests and responses are separated by a designator of separate protocol. In practice, the multipurpose internet mail extension (MIME) header contains the string "Secure-HTTP/1.1" and not the traditional banner "HTTP 1.1".

The required S-HTTP are characterised by the same structure as HTTP requests. All secure requests that use the protocol S-HTTP version 1.1 include in the header the line "Secure * Secure-HTTP/1.1". A secure server that uses the S-HTTP standard is able to accept all the variations of the case, not making a distinction between lower-case and upper-case letters. The asterisk command above represents a wildcard, and clients that use proxy must replace it with the uniform resource locator (URL) address of the request and must provide the host section of the URL address and the port reference. The S-HTTP protocol, unlike SSL, which is further discussed in the following, uses communication port 80 of the server, in a manner similar to HTTP.

On the other hand, the browser checks all responses received from the server. In a way similar to HTTP that produces a response of the type "200 Ok", an S-HTTP server produces a response of the type "Secure-HTTP/1.1 200 OK". The first line of response must always have this form, regardless of eventual acceptance. This prevents the browser from analysing the positive or negative outcome of the request, as analysis of the refusal would provide a possible hacker with information relevant to the access to a secure server, and in this sense, the server will always return the code "OK" regardless of the acceptance of the initial request.

The S-HTTP protocol provides a series of new lines of the message header.

It also supports the use of digital signatures, as already stated above. The "MAC-Info" heading allows a MAC code to be provided that allows for authentication of the message and verification of the integrity of the message itself, using the text of the message as a base, the current time and a password that is shared. The browser or the server S-HTTP calculates the MAC code based on the contents of the encapsulated S-HTTP message. The mechanism used by MAC allows rapid verification of the integrity of the message as the parties share a key.

S-HTTP provides that both communicating subjects can express their requirements and preferences concerning the cryptographic extensions allowed and/or requested by the other party. The most appropriate choice depends on the possibility of the implementation or on the requirements of the application. To transmit the features available, S-HTTP uses a negotiation block, a sequence of specifications, each of which is consistent with the description of the item in an S-HTTP block.

S-HTTP allows the automatic repetition of the attempts of a client and this may allow a hacker to put into practice various forms of attack. In this sense, the client should never encrypt the data several times automatically, unless the server that requests repetition of the attempt is able to demonstrate already being in possession of the data. Situations in which it is permitted to repeat encryption are the following:

1. The server encrypts and sends the repetition response using a bandwidth key that the server itself has just generated for the original request.
2. The recipient of the original request signs the repetition response.
3. The original request uses an external bandwidth key and the server encrypts the response using this key.

The above list is not complete, for reasons of space, but serves to clarify the importance of paying special attention to the cases in which it is necessary to perform repetition of automatic encryption. In practice, if it is not entirely certain that the user is who it claims to be, it is vital to request permission from the same user to encrypt the data. In this sense, the S-HTTP guidelines prohibit browsers from supporting automatic functionality for automatic repetition of the signature. In any case, if it is assumed that a browser will follow the other specifications, it is possible to provide the automatic repetition of the MAC authentication.

5.5 Secure Socket Layer

We have seen previously that within the Web, clients and servers communicate using the HTTP protocol, which are entirely lacking in security features and, in order to counteract this lack, the S-HTTP protocol was developed. Most Internet servers and Web sites use another secure protocol called SSL, developed by Netscape in order to ensure reliable transmissions.

The SSL protocol is characterised by the following functionalities:

1. It is an open and non-proprietary protocol like S-HTTP.
2. It is able to support the features of data encryption, server authentication, message integrity and client authentication for a TCP/IP connection.
3. It is compatible with firewalls.
4. It is compatible with tunnel connections (which are telephone connections that allow users to access the WAN and the corporate intranet via the Internet).
5. It transmits secure data by using Secure-MIME (S/MIME).
6. It supports 40-, 56- and 128-bit encryption. There are many algorithms characterised by keys with length greater than 128 bits, but, because of the restrictions on international cryptography technologies, SSL is limited to 128 bits.

In recent years, SSL has most commonly used protocol for secure transactions over the Internet. Its evolution is transport layer security (TLS) that does not unfortunately support SSL, generating an incompatibility between the two protocols.

SSL was developed by Netscape to respond to demand for reliable transmissions on the Internet. It is able to support functionality of security between application protocols (HTTP, Telnet, Network News Transport Protocol (NNTP), FTP, etc.) and TCP/IP protocols. It supports data encryption, server authentication, message integrity and authentication of the client for a TCP/IP connection. It is located between the application layer and the transport layer, as shown in Figure 5.21.

SSL is an open and non-proprietary protocol. This means that it is available for both businesses and individual subjects that need to use it in their Internet applications.

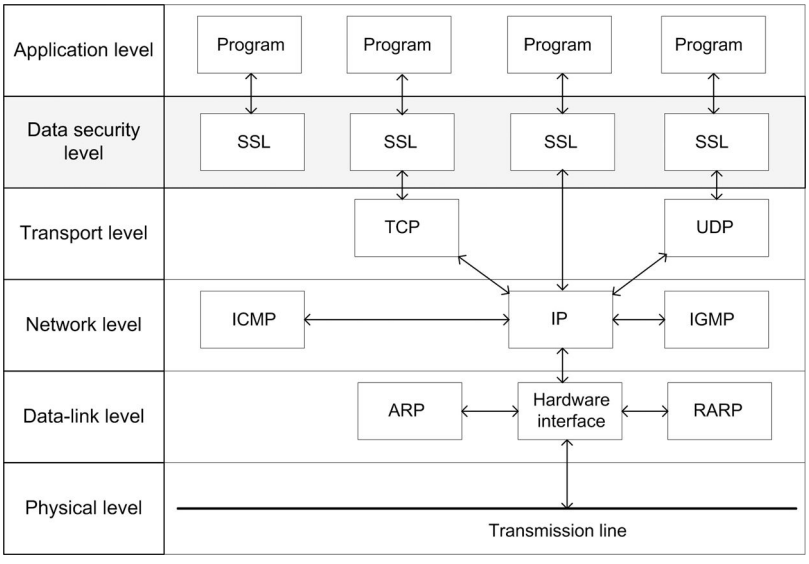


Figure 5.21 Position of SSL level within the other Internet protocols.

The manufacturer has also requested from the World Wide Web Consortium (W3C) the use of SSL as a security standard for web browsers and Web servers and in this sense, the manufacturer is working with W3C for the development and standardisation of security mechanisms and the IPs proposed.

SSL allows Web servers and users to encrypt and protect transmissions in the transactions on the Internet. It requires that a browser and an SSL server take part in the connection. Most Web browsers support this protocol.

At the dawn of the Internet, developers took no account of the need to provide secure connections on the Web. For this reason, neither TCP/IP nor HTTP is able to provide encrypted connections or to protect communications from voluntary attacks. One of the most popular protocols, in addition to SSL is, developed by Netscape, S-HTTP, developed by EIT. SSL allows servers and users to protect communications on the Web using the following services:

1. Server authentication using digital certificates, in order to discourage any imposters.
2. Security of transmissions using cryptography, in order to avoid the unwanted reading of transmitted messages.
3. Data integrity in the connections, in order to reduce the effects of a possible attack.

It is well known that data transmitted via the Internet, without any protection, is subject to manipulation by the intermediate bodies. Information that is transmitted from our computer passes through a series, sometimes high, of intermediate systems. Each of these systems represents a potential danger for the message that passes through them as the computer intermediates are able to access information in transit. Security features prevent any hacker on computer intermediates from being able to attack the information that passes through them, either by observing them, copying them, damaging them or replacing them.

In TCP/IP protocol stack, the SSL level is below the level of applications, represented by HTTP, SMTP, Telnet, FTP, Gopher, NNTP, etc., and above the transport level, which contains the TCP module, and the network layer, which contains the IP module. This location allows them to use the existing standards for communications, without SSL being limited to operating for only one application protocol.

If an SSL browser connects to an SSL server, both can exchange the encrypted information. In this way, security increases, for both subjects, allowing our message to reach the destination securely and reliably.

The authentication of SSL servers uses Ronald Rivest, Adi Shamir, Leonard Adleman (RSA) public key cryptography, together with an independent certification authority. When connecting to a secure server, it is possible to check the server certificate in such a manner as to check if we are connected to the desired server. When a browser and a server establish a secure connection, the server sends the browser a session key that both subjects will use to encrypt communications. To ensure maximum confidentiality for communications, the client and the server must exchange the key securely. To protect the session key, SSL uses public key cryptography at the beginning of the communication. When a browser attempts to connect to a secure server, it sends the server a “Client.Hello” message, which has the same meaning as an HTTP connection request, as shown in Figure 5.22. In addition to the message itself, the browser sends the server different information on the technologies of supported cryptography.

After the server has received this message, it evaluates the information contained in the message itself: if the browser supports the same protocols of encryption supported by the server, and if the other server protocols correspond to the browser, the server sends the browser the “Server.Hello” response, as shown in Figure 5.23. The response also contains the public key of the server and the information concerning the connection established by the server.

Once the client receives the “Server.Hello” message, it uses the information previously exchanged with the browser to generate a session key that is used to encrypt in direct messages to the server and vice versa. The client sends the server the session key encrypting the relevant message with the public key received from the server. The server, in turn, sends the client a message containing a second request, in such a manner that the client and the server can create the session key to be used, in the following, to

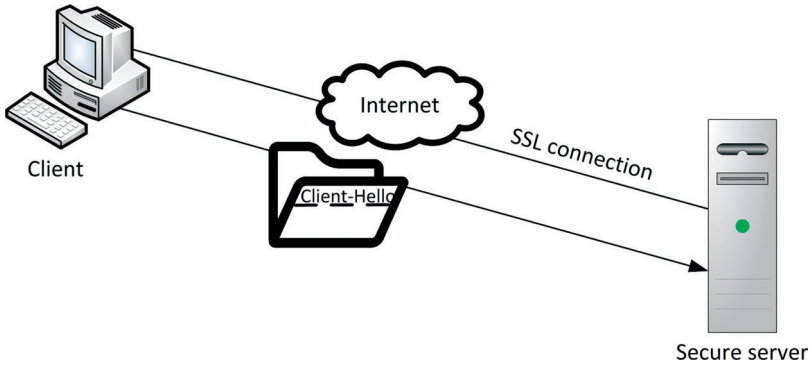


Figure 5.22 Commencement of SSL communication between browser and server.

encrypt the messages. Once the client receives the encrypted session key, it continues with subsequent requests. If the server uses HTTP under SSL, the client encrypts the HTTP request to send to the server using the session key generated by SSL.

SSL uses a technology for authentication and encryption developed by RSA Data Security Inc., which owns the RSA encryption algorithm. SSL encrypts messages in such a way as to ensure that the encrypted connection established between the client and the server remains valid for multiple sessions and with a high level of security. Because the session key is generated for each new connection, a hacker will not be able to use an old key, which has come to his knowledge, in a new connection. To breach a message encrypted with a 128-bit key, 225 MIPS/year is required, an enormous figure to reach with the resources available to a hacker, and this makes the system extremely secure. To use an SSL encryption, a version of the browser is required that fully supports encryption.

We have seen in Chapter 1 that TCP/IP supports two types of connections:

1. point-to-point;
2. section – section.

A service point-to-point ignores the intermediate steps, whereas a section – section service performs a function for each section, which goes from the transmitter to the receiver. SSL uses point-to-point connections to ensure a high level of security for transmissions.

It has already been seen that HTTP connections take place in four steps and conclude after the server has sent the response. Connections that use SSL, on the contrary, remain open until either the client or the server explicitly ends the connection, usually when the browser requests a different URL address.

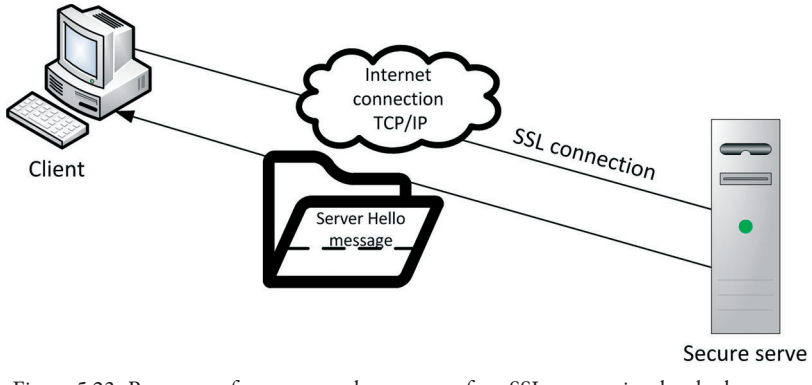


Figure 5.23 Response of a server to the request of an SSL connection by the browser.

Secure transmissions do not integrally avoid security problems for Internet users. In fact, even if we send, for example, our credit card data in a secure manner, using, for example, SSL, sensitive data could be sent to unsafe or suspicious subjects that could make unlawful use of it. Server administrators should primarily aim at avoiding security breaches and, in order to protect the information of their customers, should manage as best as possible the physical security of their servers, by controlling access to the software, the password and private keys. It is very important, when communicating confidential information, not only to ensure that the information is transmitted securely over the Internet, but also to verify the integrity of the person to whom we are communicating such data and their authenticity.

5.5.1 Features of browsers and SSL servers

There are many products that use SSL to ensure reliable transmissions and server authentication using digital certificates signed by a certification authority.

To understand if a document originates from a secure server, attention must be paid to the URL address. If it begins with "https", it means that the document comes from a secure server. To connect to an HTTP server that supports SSL, it is sufficient to add a final "s" to "http".

In the case of Internet Explorer, we are within a secure connection by looking at the icon that appears in the bottom left-hand corner of the browser window represented by a padlock. The browser itself, however, advises us every time we enter and exit a secure connection by means of an appropriate dialogue box.

To create a secure connection, the two parties involved in the communication must use the SSL protocol. Most of the browsers available on the market support this protocol. The code used to generate secure servers can vary from product to product, but the protocol uses particular conventions for names. Table 5.4 shows the correspondence between secure and non-secure protocols.

The majority of UNIX servers that support SSL are equipped with a daemon (management program for specific applications referred to above) SSL called SSLD. It is used as a proxy SSL for TCP not SSL applications. It can create a secure connection to an SSL server from a non-SSL client, as shown in Figure 5.24. If SSLD is run from a network server, it may allow connections between SSL clients and non-secure servers. To enable a secure communication channel between two non-secure processes, two SSLD processes can be used.

It is very important to pay particular attention to the configuration of SSLD because if it is not carried out carefully, security flaws in the server defence system can be opened. In this sense, it is very important to secure the ports that have been defined as auth-client as a port configured in this mode allows anyone to connect to the same, to authenticate in place of another and to use SSLD certificates. Therefore, it is necessary to verify that nobody, except those requiring access, can access the port. To avoid fraudulent connections, the value of the network must be configured in other ways. The problem can be overcome by avoiding creating an auth-client port if it is not strictly necessary. It is also necessary to secure all the client ports because a port configured in client mode allows all individuals who are able to connect on the same port to transmit to the other computers, as if they were operating directly on this computer. In this sense, the other hosts on the network that use a host authentication mechanism, by controlling the certificate, may grant connection to unauthorised parties if they access from a client port; access of this kind will invalidate the existence of SSL.

Table 5.4 Correspondence between non-secure and secure protocols

Non secure protocol	Secure protocol
HTTP	HTTPS
FTP	FTPS
NNTP	NNTPS
Telnet	Telnets

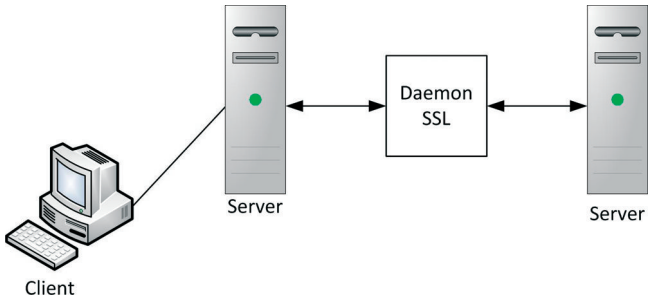


Figure 5.24 Example of connection between non-secure client and secure server operated by the SSLD.

5.5.2 Tunnels in firewalls and SSL

It has already been illustrated what firewalls are and their use to increase the level of security of a network. In many cases, it is necessary to create a tunnel within a firewall to allow authorised users to access internal resources that are normally inaccessible. In practice, outward FTP transfers and internet users can be blocked, but at the same time, internal users can be allowed to connect within the system from outside. This process is called tunnel. Given the intensive use of SSL in secure servers, SSL must extend the proxy protocol in such a manner that a client that is located beyond the firewall can connect to an SSL server. The connection diagram is shown in Figure 5.25.

To perform the connection, the HTTPS protocol is used, that is the SSL hypertext protocol, in the same manner other protocols connect to the proxy. Secure Hypertext Transfer Protocol (S-HTTP) connections use the shttp protocol connector, whereas SSL http connections use the https protocol connector. In most cases, an attempt is made to create the tunnel by asking the proxy to commence a secure connection with the remote HyperText Transfer Protocol over Secure Socket Layer (HTTPS) server and then running the HTTPS transaction. To operate correctly and to meet the request, the proxy must incorporate complete implementation of SSL. The following are two disadvantages of connecting first with a proxy and then with a secure server:

1. The connection between the client and the proxy server is not secure because it uses a normal HTTP connection or any other protocol. The connection is not secure, and, in most cases, acceptable if the client and the proxy are located within a secure network. Obviously this does not exclude the possibility of attacks originating from within the network.
2. It is not possible to manage SSL tunnel without having to use SSL compatibly with the current protocol of the proxy.

Since there is no alternative to the use of SSL tunnel within a proxy, it is essential to generate the proxy exchange remembering to avoid a situation where the proxy has access to the data transferred in

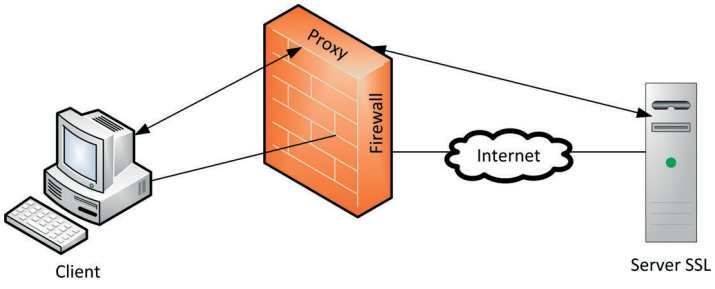


Figure 5.25 Connection diagram of a client with a proxy server and an SSL server.

both directions; the proxy must only be aware of the origin and destination and if the same requires user authentication, also the user name that requires a connection. The connection between the client and the remote server that is connected via the proxy is determined by a handshake between the client and the proxy. To allow compatibility from top down, both the client and the server must generate the handshake in the format of HTTP/1.1 request. Proxies that are not able to transmit data without being accessed are not able to satisfy the client request and must provide the user with a proper warning message. Tunnels do not represent a topic of SSL, but they are a method by which a third party establishes a connection after which it simply copies the data in both directions.

When a tunnel is created to enable an SSL connection, the client must obviously connect with the proxy server. The SSL command format is the same as the format of http commands because SSL supports connections to the Internet of various kinds and was created, originally, to generate secure HTTP connections, given the need to respect the HTTP command convention. The client connects to the proxy using the CONNECT method to indicate the name of the host and the port number to which to connect. In the CONNECT header, the client must indicate the name of the host and the name of the client separated by a comma.

The security aspects of the CONNECT method are very important. In fact, CONNECT operates at a lower level than the other HTTP methods and is characterised by a greater ability to control the way in which the proxy manages the information that follows the command. The proxy does not need to know everything in the URL address that the client is attempting to access. The only information required by the proxy is the name of the host and the port number. The proxy is not able to check if the protocol currently operating is SSL; therefore, configuration of the proxy must restrict connections to only those ports usually used by SSL. These ports are 443 for HTTPS and 563 for Secure NEWs Server (SNEWS).

All the information provided for the proxy is used by the same for managing traffic, ensuring that the proxy can read the minimum necessary. It is also very important that the connections pass through the known SSL ports, indicating the port in the URL, because the proxy is not able to know if the connection is secure.

Sometimes use of SSL tunnel technologies are required for the mutual connection of two SSL servers, as in the case of double firewalls by both parties. When there are two firewalls, SSL treats the most inner proxy as a client of the most outer proxy.

5.5.3 S/MIME: secure extensions

It has already been seen above that the HTTP protocol, in order to transmit data via the Internet, uses MIME extensions. RSA Data Security has developed S/MIME, which represents a secure version of MIME, useful for email messages and for encrypted and digitally signed transmissions. S/MIME allows email clients to send encrypted messages and to authenticate messages upon receipt. Servers and SSL clients use S/MIME to transmit from ciphers on the Internet. S/MIME uses authentication and encryption of messages in most Web browsers that are currently used. It includes the following features:

1. Cryptography for the privacy of messages.
2. Server authentication using digital signatures.
3. Detection of tampering.
4. Interoperability with other S/MIME software.
5. Transparent integration in other packets.
6. Inter-platform messages.

S/MIME encryption ensures confidentiality of messages. S/MIME authenticates senders' messages by reading their digital signature and using a hash function to check if the messages themselves have

undergone modifications due to voluntary attacks. S/MIME is an open standard that can, therefore, communicate with any S/MIME client. It is able to support X. 509 certificates, enabling users to send signed and encrypted messages both within and outside organisations to which they belong. S/MIME is easily integrated into the software, allowing the straightforward signing and encryption of messages. This is important for browsers and SSL servers, because both parties use S/MIME instead of MIME extensions for the transmission of messages.

5.6 Intrusion detection

Intrusion detection system (IDS) is a relatively recent technology that provides very valid performance with regard to the security of networks. In some cases, it is alleged that this technology can eliminate the presence of firewalls, but this statement is not entirely true as an IDS can be a valuable aid for firewalls to increase the overall security of networks.

IDSs, in practice, analyse and control the traffic that passes through the network and are aware of possible attacks that a potential attacker can attempt: through constant examination of the data stream, they are able to recognise any suspicious traffic and block it, promptly informing the network administrator. An IDS continuously compares the packets in the network with the models of known attack, readily identifying one of them, taking all the necessary countermeasures in such case.

The countermeasures depend on the IDS used and its configuration. All the IDSs are able to identify suspicious events. Some IDSs acquire samples of network traffic to allow later analysis by the network administrator. Other IDSs can be equipped with advanced warning features, being able to send emails or SMS messages to the network administrator in the event of critical events. Most IDSs are able to communicate with the firewall and the router to vary the filtering rules and to intervene in a possible attack in progress, blocking it. An IDS is composed of two parts: a sensor to read and analyse the traffic; a console through which to manage the sensor and on which all the information about the traffic is shown.

IDS are very demanding in terms of the use of resources and must work on relatively powerful and dedicated platforms. In addition, as they analyse all the traffic, they need a considerable amount of memory capacity to store all the work data.

However, IDS have limitations. In particular, they are not able to tolerate high traffic volumes and are not capable of detecting attacks when traffic is encrypted, leaving Web servers without protection. In many cases, they are not able to respond in time to an attack as shown in the following, in an example of attack called teardrop, which belongs to the category of DoS attacks, which is further discussed below. To fully understand a teardrop attack, it is necessary to understand the operation of the offset field and the field length that make up IP packets. The offset field is used by routers which, if they receive a packet that is too big for the next segment, must fragment it before transmitting it. The offset field is used together with the length field in order to allow the recipient to recompose the datagram in the proper order. If a system receives a fragmentation offset equal to 0, it deduces that this is the first packet of fragmented information or that fragmentation has not been used. If fragmentation has been used, the receiver uses the offset to understand the correct position of data within each packet when the received data reassemble. IP fragmentation offset, in practice, tells the receiving system at what distance from the beginning of the datagram the received data must be placed. If everything works correctly, this system allows the data to be placed in the correct order of transmission. The length field is used as a final check to ensure that there are no overlaps, and that the data received have not been corrupted during transmission. For example, if by positioning fragments A and C, it turns out that the intermediate fragment B, received afterwards, is too large to be positioned between A and C because it overwrites part of the fragment C, this means that there is a problem. In this case, the receiving system will try to reassemble the data received and, if it fails, it will send the sender a request

for resending the data. Most IP stacks are capable of managing overlaps or the blocks of data that are too large for their segment. Now that operation of the offset and length field has been explained, we can examine teardrop attack. This attack begins with the sending of a normal size data packet characterised by a fragmentation offset equal to 0. For this reason, the above-mentioned type of attack is not distinguishable from a normal data transmission. Packets characterised by modified offset and length fields are subsequently sent in order to generate a crash of the receiving system, fully blocking it. When the system receives the second packet, the fragmentation offset is checked in order to determine at which point in the datagram its information should be placed. Specifically in a teardrop attack, the offset of the second package requires its information to be placed within the first fragment. In practice, the second fragment does not overlap with the first fragment but appears to be entirely contained within it. Since this error condition is not provided by the receiving system, there is no program of management of the same, with the result that the buffer memory is overloaded generating a crash of the receiving system. In many operating systems, just one altered packet is required to generate a system crash, whereas in many others, a certain number of these packets must be received.

An IDS, in order to detect this type of attack, from examining the first packet might not deduct anything since it appears as a normal data packet. From examination of the second package, the IDS would deduce that a teardrop attack was under way and would activate its alarm program, also alerting the network administrator. If the operating system is able to withstand more than one packet of this type, the system does not crash otherwise the system becomes blocked from the first packet. To avoid this last condition, this operating system must be suitably programmed to avoid a crash situation upon receipt of the first packet, allowing the IDS to be able to intervene in time. We might think it an idea to block the IP address of the attacker but, in most cases, the latter, in order to avoid identification, has more than likely used the technique of IP spoofing, which consists of making the attack seems as if it has originated from a place different from where the aggressor is actually located. This attack technology will be discussed below in more detail. In this case, unless the IDS is located in the same collision domain of the attacker, it is not possible to deduce that the IP address used is false. In this situation, the attacker would be able to randomly vary the source IP address from time to time in order to generate new attacks.

However, it is common belief that it is not possible to prevent all types of attack based only on observation of traffic (sniffing). This is due to the fact that almost no IDS reassembles the IP packets in the same manner in which the systems communicate via IP. This causes inconsistencies between what the IDS captures and what the receiving system is capable of processing. One of the problems is that some IDS do not verify the checksum field in the IP header, an operation that is performed by the recipient; alteration of this field causes the recording by the IDS of a valid data block different from that which the receiver processes. One example is the so-called PHF CGI attack, where the IDS seeks to identify such an attack by looking for the "phf" character string within the necessary data of all the HTTP requests. If this string is found, the IDS can deduce that this type of attack is taking place. An attacker that has been detected could send a series of packets, each containing one of the characters that make up the "phoof" string, suitably manipulating the checksum in such a manner that the packets containing the letter "o" are not valid. As a result, the recipient would process the "phf" string while the IDS, not checking the checksum, would then read the "phoof" string.

Rather than attack the system, it is also possible to attack the IDS, inhibiting its ability to detect intrusions. Once the IDS is deactivated, a potential attacker can continue his/her attack against the network without running the risk of being detected. This can be obviated by not making the IDS addressable by any host in the network since analysis of the traffic does not necessarily require a valid IP address. The only systems that need connectivity are sensor; console; DNS system, if we need to break down the IP addresses into host names; firewalls or routers, if we want to allow the IDS to vary the filtering rules. To isolate the communications of the IDS from the network, a separate private network with a private IP address can be used. The sensor, on the contrary, needs an IP stack protocol and thus an IP address on the main network, but this does not mean that it must be a valid address. In the same

manner as a firewall, an IDS sensor that uses an IP address on a public network must be properly reinforced before use, ensuring that it has installed all the latest security updates (patch) and that the same does not execute services that are not strictly necessary. In this case, the system will be more resistant to attacks and will be able to check, with greater efficiency, the security of the network being monitored.

The fundamental difference between a firewall and an IDS is that a firewall works as a keeper, so that all inbound and outbound traffic passes through it for the necessary security checks. If the firewall is attacked, its services may be interrupted and the same may open, blocking all traffic. In this case, the attacker would not be able to disable it to then continue the attack against the hosts on the internal network. An IDS, on the contrary, is not installed between the sections of the network (preferably at the point of connection between the internal network and the external network), the latter operating as a discrete controller within a collision domain. If the IDS is deactivated, the network traffic is not interrupted, and attack on the internal hosts can be continued, as the traffic has not in any case been analysed and stored.

A type of attack that can be conducted in respect of the IDS is the internal attacks. In this sense, it should not be forgotten that the sensor and the console are the parts which are most vulnerable to attack from within. If an attacker managed to discover the IP address of the IDS on the main network, he/she could change the local address, addressing such systems onto the main network. Such systems are protected as long as no one knows where they are hidden, making them completely inaccessible from the outside network, limiting the environment from which attacks can come and facilitating the process of interception of the attacker. To protect IDS from attacks from within, an IDS can be used that does not require an IP stack; without an IP address, the system can be attacked by any type of IP address-based attack. It is clear that special precautions should be taken in this case, to protect the control console. In this sense, the IDS console can be installed on the same sensor system or a second network card installed in the sensor in such a way that the latter is able to communicate with a console via a private network.

IDS, in addition to recording and warning, has following two additional features: interruption of the session and manipulation of the filtering rules.

Interruption of the session refers to the easiest countermeasure to use. It can be implemented in different ways but the simplest way to do this is interruption through reset or closure, by the IDS, of each terminal of an attack session. This operating mode may not prevent the attacker from continuing with attacks, but would prevent the same from causing further damage in the current session. Suppose, for example, that the IDS sensor detects an attacker that is trying to send an appropriate string of characters during an FTP session in order to gain the root level access (event that can occur in older systems). This level of access does not require any password authentication and the attacker would gain access to reading and writing for all the system files. If session discontinuation were enabled, the IDS sensor would be able to identify such an attack and, subsequently, transmit false ACK-FIN packets to both terminals of the session to close the connection. The IDS sensor could do this because it would impersonate because of the system that is located on the other side of the connection. It would transmit an ACK-FIN to the attacker using the source IP address, port numbers and sequence numbers of the FTP server. By doing this, the connection would be closed, preventing the attacker from accessing the system files. Depending on the capacity of the IDS sensor used, the same could attempt to stop all communications of the attacking host indefinitely or for the time defined by the administrator. Session discontinuation is an extremely powerful functionality but one that is also characterised by limits. For example, in the event of a teardrop attack, the IDS would not be able to prevent it by analysing a single packet. It could prevent it from analysis of the second packet, but in this case, it may be too late for some systems that might crash following receipt of the first packet.

With regard to manipulation of the filtering rules, this is the capacity of an IDS to intervene in the operation of a router or a firewall to avoid repetitive attacks. This feature prevents the attacker from transmitting additional traffic to the host that is the objective of his/her attack. In this case, the IDS

adds a new filtering rule to the firewall that stops all the traffic coming from the IP address listed as suspect by the IDS. This operating mode is characterised by positive and negative points. With regard to the positive aspects, the same can prevent an attack by generating an amount of traffic that is less compared to the session discontinuation previously seen. As soon as the IDS changes the filtering rules, the attack stops immediately. On the contrary, if the session is interrupted, the IDS must close in continuation every attack session, and in the case of particularly aggressive attacking, a significant amount of traffic could be generated.

The negative aspects are non-integral protection offered by this mode of operation. In fact, it is not able to deal with attacks originating from within, because even the traffic rules had been altered, the same, coming from an internal IP address, would not pass through the firewall and could not, therefore, be blocked. In addition, an expert attacker may use a false IP address instead of a real one, changing it gradually as the firewall blocked the traffic coming from that IP address. Session discontinuation, on the contrary, counteracts the type of the attack where the original IP address is not used, and in this case, the latter could address the attack in a continuous manner, which the manipulation of filtering rules could not do. In the latter case, the IDS would be forced to constantly chase and block the new IP address from which the attack was coming but would not be able to do this with high speeds because the time necessary to react, to transmit the information to the firewall and change the rules of the firewall itself, would be around 20 to 30 s.

The ability to change the filtering rules should be used with caution and only against attacks that are considered particularly dangerous.

5.6.1 Installation of an IDS on a host

The IDS seen so far are designed to be installed on a network server to constantly monitor the traffic. They are used for controlling the traffic within a collision domain. In addition to IDS that run on servers, there are also IDS that run on hosts, designed to protect an individual computer.

The IDS that runs on a computer operate in a similar manner to an antivirus as the process is run in the background, controlling the occurrence of suspicious events. Suspicious activity includes attempts to transmit unknown commands through an HTTP request or even changes to system files. When a suspicious activity is detected, the IDS usually ends the suspected session or sends a message to the system administrator.

Such systems are, however, characterised by certain disadvantages. One of the first disadvantages is the capacity to control only a small number of systems. In addition, even if most of them control the main functions, such as changes to access rights, an attacker could find a way through which to disable the IDS before attempting to perform modifications to the system. Another disadvantage is that IDS that operate on hosts run in the background and do not have access to the main functions of communication of the computer itself; as such, the IDS is unable to cope with attacks aimed at the protocol stack. For example, on an NT server without a stack, 10 or more teardrop packets would be needed to generate a system crash; an IDS on servers would have sufficient time to react and counter the attack, whereas an IDS on hosts would not even be able to detect the attack.

From what we have seen so far, it can be inferred that installing an IDS on a host is not an optimal solution from the point of view of security as an attacker that managed to enter the system might also disable the IDS, leaving the computer requiring protection completely vulnerable.

Often attackers forget to erase the traces of their raids within the computer, forgetting to delete logs and suspicious processes. For this reason, it is always recommended for system administrators to forward all logs to a remote system; in this way, if an attacked computer is altered, the logs will remain unchanged. The same principle should be used for IDS.

However, the IDS on hosts are characterised by certain advantages. For example, suppose we have a Web server that needs to be protected and that is located on a demilitarised network section (DMZ).

This DMZ is located behind a firewall, but on a network section that contains the only Web server. The firewall is configured to allow HTTP traffic to the sole server. In this situation an IDS on host will not be able to protect the full Web server because most protection is guaranteed by the firewall that only allows HTTP requests to pass. This means that we do not have to worry about breach of the other services on the Web server, and the IDS need to only deal with ensuring that HTTP requests are suspect and do not require access to the file or be alert for known CGI or Java bugs that might breach the Web server.

IDS on hosts are very useful in all environments governed by switches, such as that shown in Figure 5.26.

As can be seen in Figure 5.26, all the computers are connected directly to one main switch, generating individual collision domains for each computer; in this condition, the switch isolates the unicast traffic in such a manner that it becomes visible to only two computers that, in turn, communicate. Because the switch isolates the communication sessions, any IDS that runs on network servers is not able to see all the traffic in transit; if an internal computer tries to attack the Web server, the IDS would not be able to reveal and combat it. In addition, traces of such an attack would not be present in the logs of the IDS and the event itself would not be recorded. In this sense, better protection may be provided by an IDS that runs on hosts, because, the latter being executed by the system itself, it is not isolated from the switch and is able to see all the traffic of the Web server, protecting the latter from HTTP-based attacks. The vast majority of switch producers allow configuration of one of their ports as the control port: this enables the switch to transmit a copy of all the traffic that passes through it to the indicated port, allowing a possible network-based IDS and connected to this port to monitor all the traffic passing through.

5.6.2 IDS fusion

Currently, new IDS devices generally offer the integration or fusion of data, in order to allow a greater degree of protection. If server and host information packets are combined with the other information from other sources, these IDS are able to intercept a possible attack with greater accuracy. Following are the additional sources:

1. Simple Network Management Protocol (SNMP) that allows network devices to transmit their own operating state to a central control system. A possible example is a router that updates a central

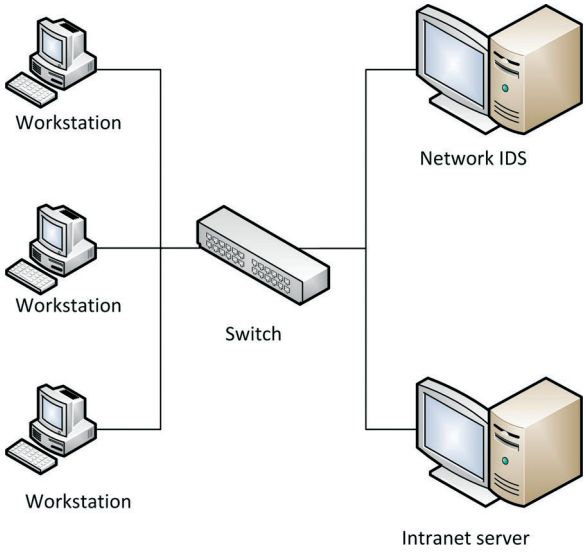


Figure 5.26 Example of IDS operating in an environment governed by switch.

system on the state of its traffic, to determine if a hacker is attempting a DoS attack (which is discussed below).
2. System log that allows the recording of a considerable amount of information on the state of overall operation and the state of operation of the individual components. For example, an email server can provide information not only on messages but also on the IP source address; this information can be used by a possible IDS to search for the origin of messages that are harmful for the network, containing viruses, advising all email servers of the protected network to filter all messages coming from the identified and reported IP address.
3. System messages that are used to have a general framework of the entire network in order to combine data and retrieve information on the status of the network;
4. Commands that allow IDS to exceed the generic limit of all operating systems do not record every command issued by each user. In this sense, the IDS stands out from among the operations performed that may not be considered by the system log that lingers only on transactions involving security. Consider, for example, a command designed to erase confidential information of an organisation; even if, the same can be extremely harmful to the organisation itself if performed incorrectly, the command itself does not violate or influence the integrity of the system as a whole;
5. Behaviour of the user that allows, if appropriately controlled, the creation of a behavioural model of the same; from comparison of this model with the current behaviour of the user, an IDS can understand if our account has been breached or not and anticipate possibly even more dangerous breaches of the security of the entire system.

Even if, the concept of data fusion may at first sight seem very simple, this operation is not immediately applicable given the enormous amount of factors and quantities to be acquired, checked and reported. It does, however, allow the performance of a significant step forward in improving the security and performance of IDS.

5.6.3 Configuration of an IDS

It has already been mentioned that, normally, an IDS is composed of a sensor that records all the network traffic on a given section and compares it with the types of attack: from a server, which controls the system log and the interface traffic, and a console, which controls the entire system, including the sensor network and server, and stores the main database used to generate reports.

If the system needs to check a reduced bandwidth connection, it is preferable to resort to the use of a single computer with good capacity rather than several computers characterised by poor quality. If, on the contrary, a network backbone and an area characterised by a high amount of traffic need to be checked, then two or more specially equipped machines can be used, as the reception and processing of each single packet requires a powerful microprocessor (CPU). It is recalled that the IDS examines each packet to search for more than, on average, 100 suspicious circumstances and at the same time, record the log and possibly activate countermeasures, for which reason the work load is quite high.

Positioning of the IDS within the network is very important in order to ensure maximum security to the network itself and to suitably protect the desired systems. In this sense, it is very important to focus the security objective well before selecting and proceeding with the purchase of IDS software or hardware. A potential installation is shown in Figure 5.27.

As can be seen in Figure 5.27, both the DMZ and the zone within the firewall are controlled, allowing verification of all the traffic that originates from the Internet, and strengthening the defences of the existing firewall. Two sensors are installed in the identified areas, without IP address as the same are connected to public network sections. The IP address is only used on the network card that connects the sensors to the console, allowing the sensors themselves to remain completely invisible to all systems present on the sections of public network being controlled.

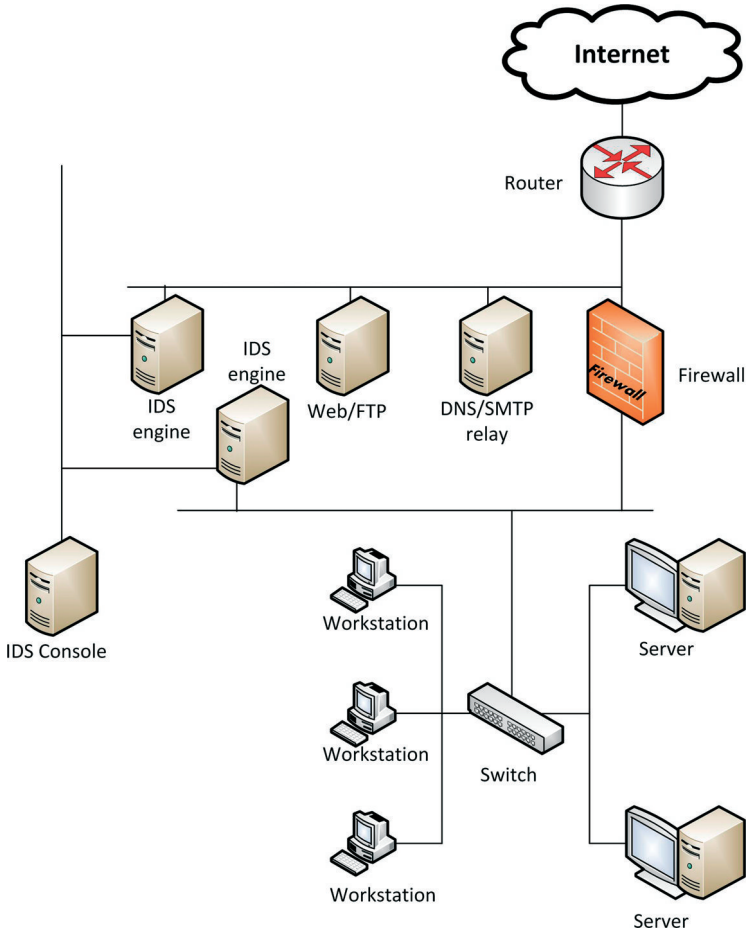


Figure 5.27 Potential installation of an IDS.

This configuration, however, has limitations. First of all, it is not possible to check the traffic coming from the Internet and intended to attack the firewall. The firewall should be able to detect and record this activity but, in this case, without exploiting the advantages of the packet capture, manipulation of the rules of dynamic filtering and other features that an IDS could ensure. If connection with the Internet is characterised by a relatively high speed and we want to check all the traffic, it is necessary to use a relatively powerful server and run the IDS functions outside the firewall. As an IP address is not required, the system installed on that section of network should not be subject to attack and should, therefore, be relatively safe.

Another limitation of the configuration of Figure 5.27 is the impossibility of controlling the unicast traffic exchanged between internal systems, for the reasons discussed above. If all the network traffic needs to be checked, the internal IDS sensor should be moved onto a switch port, properly configuring the aforementioned port for monitoring function, permitting the IDS sensor to achieve its objective.

If the goal is to protect the network at the highest level, an IDS sensor could be installed outside the firewall and a second IDS sensor on a monitoring port, connecting both the sensors to the console through a private subnet. This configuration allows all the network traffic to be checked through the control of a central console.

It is clear, therefore, that once the areas to be checked have been identified, IDS sensors required can be selected along with the relevant software and hardware.

5.7 Network attacks

A hacker can attack systems using different types of attack. This section describes the most widespread technologies directed against networks. As a hacker can intercept transmissions that occur over the Internet or through any network that is accessible to the hacker, the risks posed to our network from the different attacks can be recognised and will be described. The main attacks that will be described are:

1. DoS that consumes the resources of a server, preventing the server from being able to provide the service to other users;
2. attack in anticipation of the TCP/IP protocol number sequence;
3. TCP session hijacking attack;
4. interception (*sniffing*) of packets in transit on the network;
5. hijacking or *spoofing*, in which the hacker replaces an existing IP address in order to emulate a trusted server on an existing network connection;
6. passive attack, which makes use of pre-emptive *sniffer*;
7. *hyperlink spoofing* attack directed against SSL server installations;
8. *Web spoofing* attack that allows hackers to intercept all the transactions that a user and a server exchange during http transactions.

5.7.1 Denial-of-service attack

The DoS attack is able to bring whole systems to their knees, and for this reason systems are much feared by network administrators. It is intended to consume system resources, thus preventing other users from using the same. This type of attack can be simple or complex.

In many cases, a firewall is able to detect this type of attack by recognising that there is a repetitive request from the same remote host.

There are more complex DoS attacks that resort to written programs using different programming languages.

5.7.2 Number sequence anticipation attack

We have seen previously that a computer connected to a network is equipped with a unique IP address and that the same, when it sends packets, includes the destination IP address as a unique number called sequence number. When a TCP connection is active, the recipient accepts only packets with correct IP address and sequence numbers. It has also already been shown that most of the security devices, including routers, allow network transmission only to and from computers characterised by certain IP addresses.

A TCP/IP number sequence anticipation attack exploits the manner by which networks address computers to create packet sequences.

In practice, a hacker's attack is divided into two stages. During the initial phase, the attacker must learn the IP address of the server. In that case, if the attacker learns the domain name of the server, he/she can use the ping command to also learn its IP address. Once the IP address of the server is known, the attacker can find out the IP address of the other computers connected to the server network that share certain parts of the server address and using related addresses, the attacker tries to

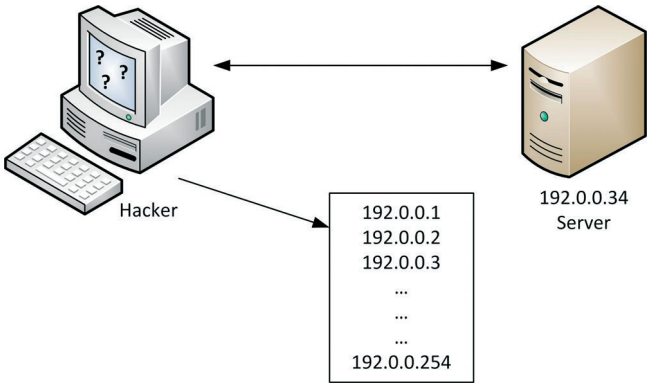


Figure 5.28 Example of an attempt to search for the IP addresses on a network given the address of the server.

simulate a number of IP address that allows it to pass through the router and enter the system as an internal user.

For example, if a system is characterised by an IP address and connected to a class C network, where there may be up to 256 computers, the attacker may try to guess their address by varying only the last bytes, as shown in Figure 5.28.

Once the hacker finds one or more network addresses, he/she also begins to check the sequence numbers of the packets that these computers exchange with each other. After a certain period of time sufficient to acquire the greatest number of information, the attacker attempts to predict the next sequence number, which is generated by the server, and will generate a packet with a sequence number provided in order to be inserted between the server and the host. As the attacker already has the IP address of the server, he/she can generate packets marked with the correct sequence numbers and IP addresses that allow him/her to intercept communications with the host. Figure 5.29 shows the manner in which the attacker can deceive the server posing as a host on a network, simulating an IP address and a packet sequence number.

Once the attacker has managed to gain access to the system through the provision of a sequence number, he/she may have access to information that the communication system transmits to the server, including password files, confidential data, login names and any other information that is transmitted across the network. In general, an attacker uses number sequence anticipation to prepare an even more dangerous attack against the server, or as a starting point to attack another server on the network.

A straightforward and extremely functional method, which avoids number sequence anticipation attacks, is an activation of the audit trail protection (which is described below) on the routers, firewalls

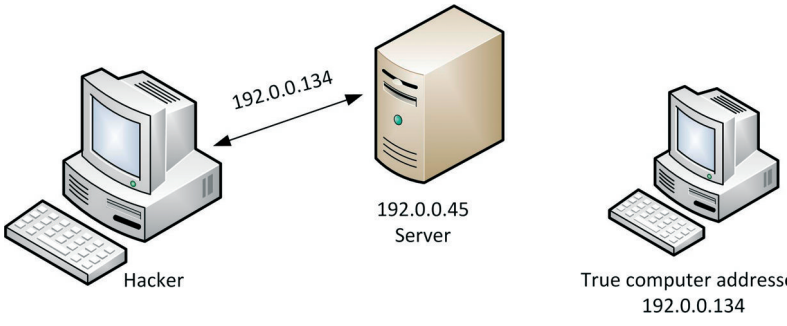


Figure 5.29 Example of how the attacker simulates a TCP/IP transmission with the server, deceiving it.

and on each server in the system. Due to the audit trail, it is possible to find out when an attacker is trying to cross a router or firewall or when trying to access a server.

5.7.3 TCP protocol hijack

TCP protocol hijack is one of the greatest threats to servers that are connected to the Internet. It is also called *active sniffing*. Number sequence anticipation and TCP hijack attacks are very similar. Their difference is that in the second type of attack, the attacker gains access to the network as the attacker manages to get it to accept his/her IP address as if it were a valid network address; in this case, the attacker need not try different IP addresses to find one that is valid. In TCP hijacking, the attacker gains control of a host that is connected to an objective network. Then, the same disconnects the host from the network and deceives the server, replacing the host. Figure 5.30 shows an example of this type of attack.

Once the attacker has taken the place of the disconnected host, he/she changes the IP address within each packet, replacing it with his/her own IP address and also varies the sequence numbers. This simulation of sequence numbers is also called *IP spoofing*. Due to the replacement of the IP address, an attacker can simulate with his/her computer the IP address of a host authorised to transmit on the network. This technique is discussed in detail below. Once the attacker has managed to fool the destination host, he/she uses a suitable sequence number to become the new server destination.

TCP hijack attacks are more dangerous than *IP spoofing* attacks because in the former case, an attacker can gain a higher access level than the second case because the attacker intercepts transactions in progress and does not simulate a given host by the activation of a new transaction.

5.7.4 Sniffer attack

Regarding the sniffer attacks, the attacker must have access to packets that cross the network. These types of attacks can be carried out in respect of all the computers connected to the network, if they are not protected, as the traffic passes through all network cards; if a dedicated programs is used, called *sniffer*, it is possible to have access to all the packets in transit.

This type of attack is usually passive because the attacker does nothing other than read the content of the packets on the network. It is usually a prerequisite to other types of attacks, such as hijacking or *IP spoofing* attacks, which are generally more dangerous. For this reason, reference is often made to passive sniffing, which represents one of the most common attacks on the Internet. Figure 5.31 shows a diagram of this type of attack.

If the attacker has in some way physical access to the network, via an internal computer connected to the network, it is relatively easy to install a program to sniff and run a control of all the traffic in transit.

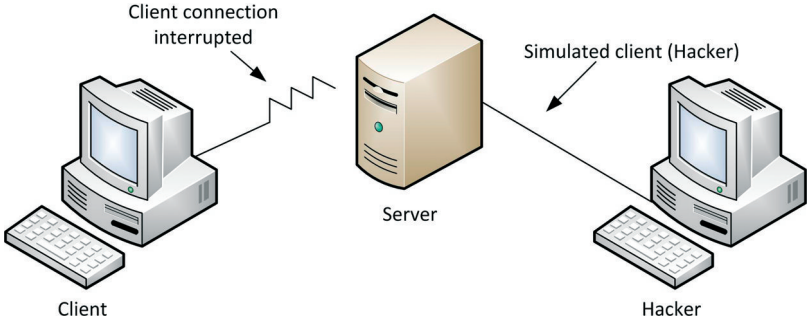


Figure 5.30 Example of TCP hijack attack.

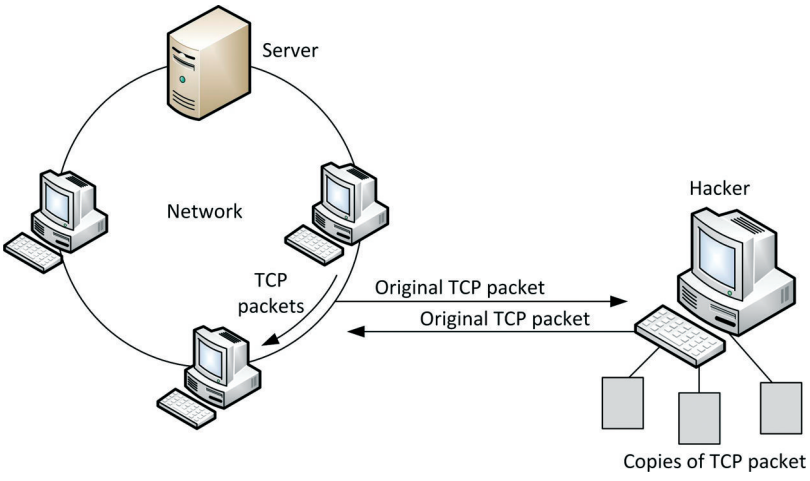


Figure 5.31 Example of passive sniffing attack.

If the attacker fails to gain physical access to the network, he/she must first breach the system. Once the system has been breached, it is possible to run the dedicated program inside the network and the contents of all the packets in transit can be checked, in order to prepare targeted and more dangerous attacks.

5.7.5 Active desynchronisation attack

It has already been seen that a TCP connection is effected through an exchange of synchronised packets. If, for whatever reason, the recipient receives a packet with a sequence number different from that expected, the packet is rejected while waiting for the correctly numbered packet. An attacker can guess the sequence number of the TCP protocol to intercept the communication.

To attack a system, the attacker, by resorting to a desynchronisation attack described above, forces both subjects in TCP communication into a state of desynchronisation in such a manner that they are no longer able to exchange data. Immediately afterwards, the attacker uses an external host to intercept the packets and to generate replacement packets that may be acceptable by both the subjects of the original communication. These packets must obviously simulate packets that the two subjects would have exchanged.

A subset of desynchronisation attacks is hijack attacks with post-desynchronisation. In this case, it is assumed that the attacker is able to access all the packets that two computers exchange in a TCP communication. It is assumed, moreover, that the attacker can intercept each packet and that he/she can replace every packet with his/her own specially created packets. Generated packets allow deception of both the client and the server. If the attacker has the properties stated, he/she is able to force transmission between the client and the server that will therefore take place between the client and the attacker and vice versa, and between the server and the attacker and vice versa.

If the attacker is able to desynchronise the TCP connection and the client sends a packet containing both the sequence number of the next packet and the next value of acknowledgement to assign to the packet, because the connection is desynchronised, the sequence number of the client packet is not equal to the sequence number previously communicated by the server and the server itself is not able to accept the packet, and refuses it.

After a short period of time, which allows the server to transmit the packet again on the network, the attacker sends the server the same packet sent to the client suitably changing the content. Because the current packet contains the correct sequence number, it is accepted by the server. At the same time,

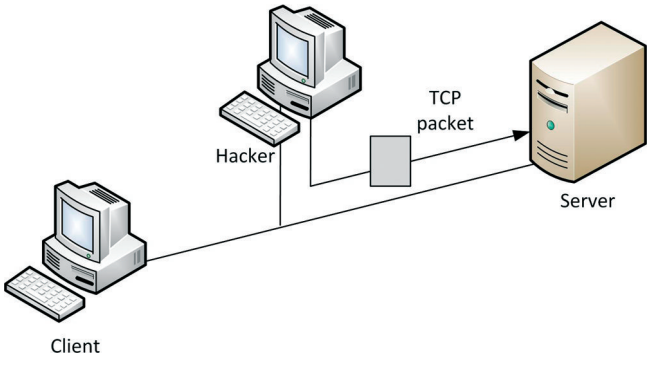


Figure 5.32 Connection status after the attacker has transmitted the modified TCP packet.

depending on the number of packets transmitted from the client that are in any case not accepted by the server, the client can be engaged to transmit other packets or to transmit packets of acknowledgement. Figure 5.32 shows the connection after the attacker has transmitted the suitably modified TCP packet.

As all communications pass through the attacker's computer, the latter can add or remove any packet from the network. Figure 5.33 shows how the attacker can add commands into the connection that connects the client to the server.

Once the server receives the packet, it responds both with the data requested by the attacker and with the data requested by the client. The attacker can filter the packet, eliminating all of the responses that the server has generated as a result of the command sent by the attacker and send the filtered response to the client. In this way, the user of the client will have no way to suspect the presence of the attacker within the connection with the server. Figure 5.34 shows how the attacker intercepts the return transmission and deletes the information that he/she has requested.

The hijack attack with post-desynchronisation is characterised by the disadvantage of generating a large amount of TCP packets of ACK type. This exaggerated presence of ACK packets is also called TCP ACK packet storm (TCP ACK *storm*). Each time a host, client or server, receives a packet that is not acceptable, the attacker performs acknowledgement of the packet by sending the expected sequence number to the sender; this packet is an ACK packet.

During an attack of the type described above, the first TCP ACK packet includes the server sequence number. The client may not accept such a packet because the client has not sent the amended request packet. As a result of this, the client generates an ACK packet to the server, forcing it to

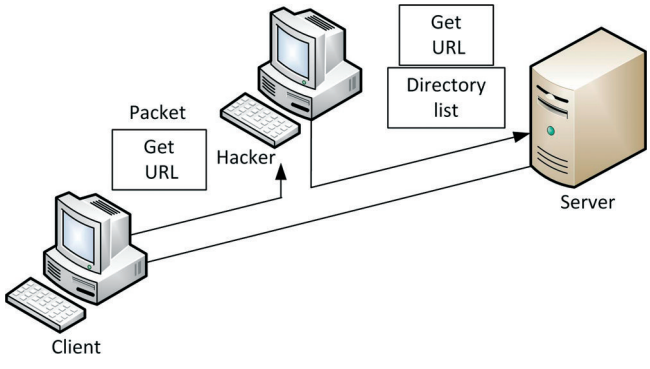


Figure 5.33 Schematic diagram of the process of adding commands to the packet.

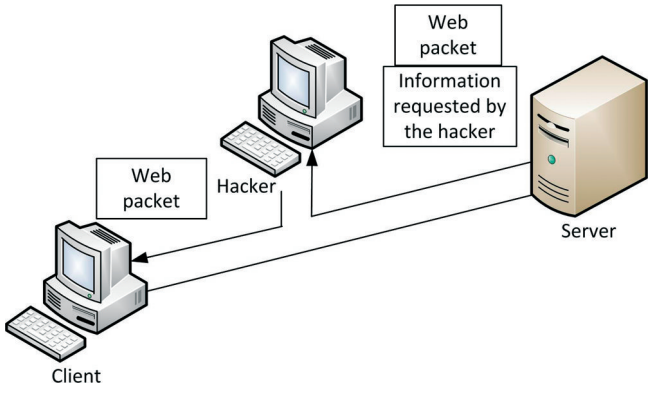


Figure 5.34 Elimination, by the attacker, of the packet that contains the server's response to information requested by the same attacker.

generate, in turn, an ACK packet directly to the client and so on, in a virtually infinite cycle. This situation is represented schematically in Figure 5.35.

As the ACK packets do not contain data, the sender of the same does not send them again in the case where the recipient loses them. In practice, if one of the two hosts loses a packet during an ACK storm, the cycle is interrupted and the storm ends.

The TCP connection generates a cycle every time the client or the server exchanges data. If the client or the server transmits data and no attacker retrieves it, the sender transmits the data again. Once the sender has transmitted the data again, the TCP connection generates an ACK storm for each retransmission and finally, the two subjects in communication close the connection, as neither the client nor the server are sending an ACK packet. If the attacker retrieves the data transmission, the TCP connection generates only one storm. In practice, the attacker loses data packets due to excessive load on the network and uses the first retransmission; this is also the reason why the attack causes at least one ACK storm every time the hacker transmits.

In a TCP connection, virtually all of the empty packets containing the activated ACK flag are classified as invalid packets. In practice, on whatever type of networks, and in particular on the Internet, a certain number of retransmissions take place in order for communication to be successful. In a network subject to active attacks, a truly high number of retransmissions due to the load introduced take place on the network and on the host by the attacker with the ACK storm. A record of the server that counts the retransmission packets, including the ACK packets, during a TCP ACK storm, can contain thousands of ACK packets. In particular, a packet containing data during an attack can produce from 10 to 300 ACK empty packets.

A subset of the desynchronisation attacks is the initial desynchronisation attacks. This, unlike in the case of the hijack attack, which intervenes once the connection has been established, stops the connection between the client and the server at the initial phase of connection and not after the connection is already operational. This type of attack stops the connection from the server side. After having terminated the connection, the attacker generates a new connection using a different sequence number.

An initial desynchronisation attack operates according to the following steps:

1. The attacker reads the SYN/ACK-synchronised connection acknowledgement packet that the server sends the client at the initial phase, as shown in Figure 5.35.
2. The attacker extracts the SYN/ACK packet, sends to the server an RST packet (Reset request) and subsequently in SYN packet (Synchronised response) with the same parameters of the SYN/ACK server packet and, in particular, the TCP port to be used to synchronise the connection. The packet, however, contains a different sequence number. This packet is called "attacker 0 acknowledgement packet". The situation is shown in Figure 5.36.

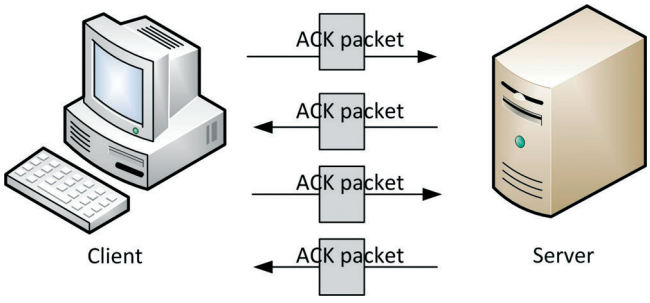


Figure 5.35 Generating a TCP ACK packet storm after an attack.

3. The server, when it receives the RST (ReSeT) packet, closes the connection and, when it receives the SYN packet, reopens the connection on the same port using a different sequence number. The server sends the client a SYN/ACK packet.
4. The attacker intercepts the SYN/ACK packet and sends the server his/her ACK packet that activates the state of synchronised connection, as shown in Figure 5.37.

The client has already enabled the synchronised connection state when it received from the server the first SYN/ACK packet. The attack is successful, if and only if, the attacker has chosen a correct value for CLT_TO_SVR_OFFSET. If the attacker chooses an incorrect value, both the client packet and the attacker packet are not accepted, generating unpredictable effects including the almost certain closure of the connection. (Figure 5.38)

A subset of the desynchronisation attacks is the null data desynchronisation attacks, which refers to the data that does not affect anything on the server side apart from the TCP acknowledgement number. In this case, the attacker performs the null data desynchronisation attack via the simultaneous sending of an enormous amount of null data both to the server and to the client. The data sent by the attacker are not visible to the client, but the same null data pushes both the client and the server forming part of the TCP connection into a state of desynchronisation because the high volume of this null data greatly affects the ability of computers to maintain the TCP connection.

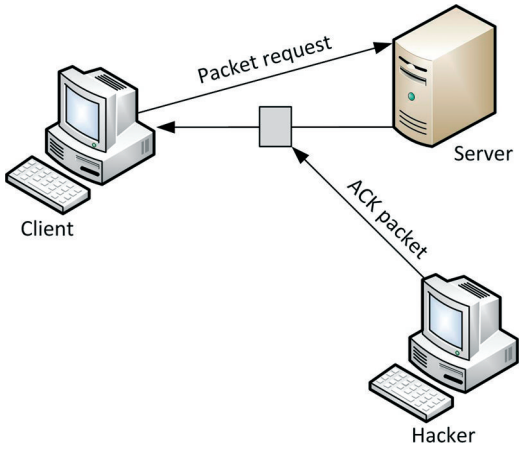


Figure 5.36 The server sends the ACK packet to the client.

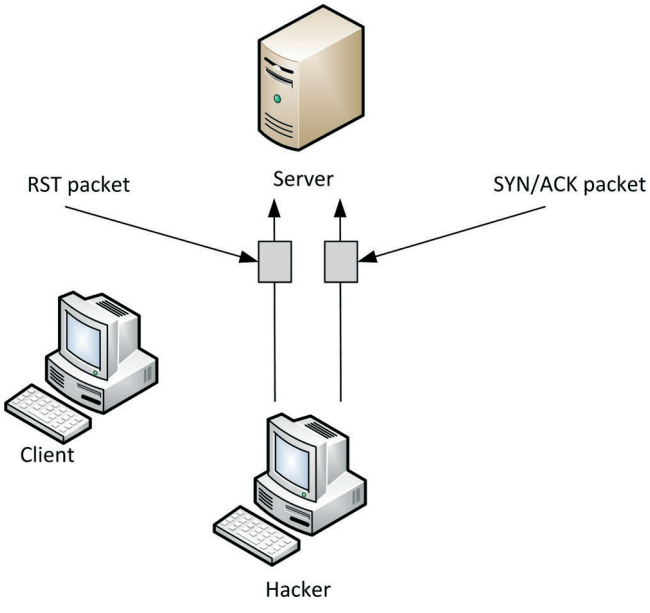


Figure 5.37 The attacker sends two packets to attack the connection.

5.7.6 Spoofing attack

It has already been stated that sniffing is a passive type attack through which the attacker reads only the packets crossing the network. Spoofing, on the contrary, represents an active type attack in which the attacker tries to convince another system that the messages sent by him/her come from an approved system. In practice, in *spoofing*, the attacker is impersonating, simulating being another system or another user.

The services of the TCP and Uniform Datagram Protocol (UDP) assume that an IP address can be valid and, therefore, give the maximum degree of confidence to this address. An attacker can use a

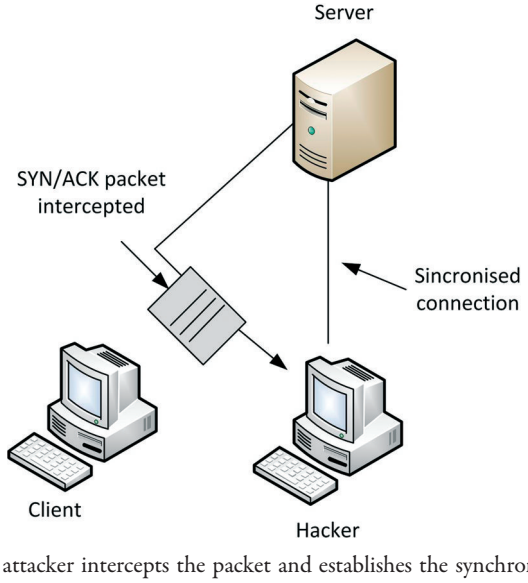


Figure 5.38 The attacker intercepts the packet and establishes the synchronised connection.

routing of the source IP address to specify a direct path to a destination and a return path for the source. The path can also involve routers or hosts that do not usually send traffic to that destination. In this way, the attacker can intercept or amend communications without having to meet packets directed to the true host.

A typical attack takes place according to the following steps:

1. The attacker varies his/her IP address in such a manner as to use the IP address of a valid client.
2. The attacker creates a path that leads to the server using the valid client as the last leg of the route to reach the server.
3. The attacker uses the source path to send the server a client request.
4. The server accepts the request of the attacker as if the same had originated from a valid client and then returns the answer.
5. The valid client forwards the packet to the host of the attacker using the source path.

The diagram of attack referred to the above steps is shown in Figure 5.39.

Most of UNIX-type hosts accept the original transfer of the packets and therefore can pass the packet of interest in the manner indicated by the new path. There are also many routers that accept the re-transfer of packets while some can be programmed to stop the transfer of packets to other systems.

The easiest method to implement a *spoofing* attack against a particular client is to wait for the client in question to turn off his/her machine, allowing replacement on the network with the attacker.

In many organisations, computers are employed that use TCP/IP protocol to connect and use UNIX host, such as, for example, local servers. Computers are usually authorised to access the directories and files of the server using the NFS system () of UNIX or Linux, as NFS uses IP addresses to authenticate the client. An attacker might be presented as the true client and configure a computer with the same name and the same IP address of another authorised computer and begin to test the connections with the UNIX host. An attacker can perform this type of *spoofing attack* with relative ease because the same is performed from within as only an internal user knows when the other computers are turned off.

Spoofing can also be performed by means of email because it is relatively easy to intercept email messages on the Internet, and it is not possible to ensure the reliability of messages without the use of digital signatures. When hosts exchange correspondence, the same use ASCII commands, with a simple attack being carried out in the following manner:

1. The attacker connects to the SMTP port, using Telnet, and enters the corresponding input commands in an ASCII format.

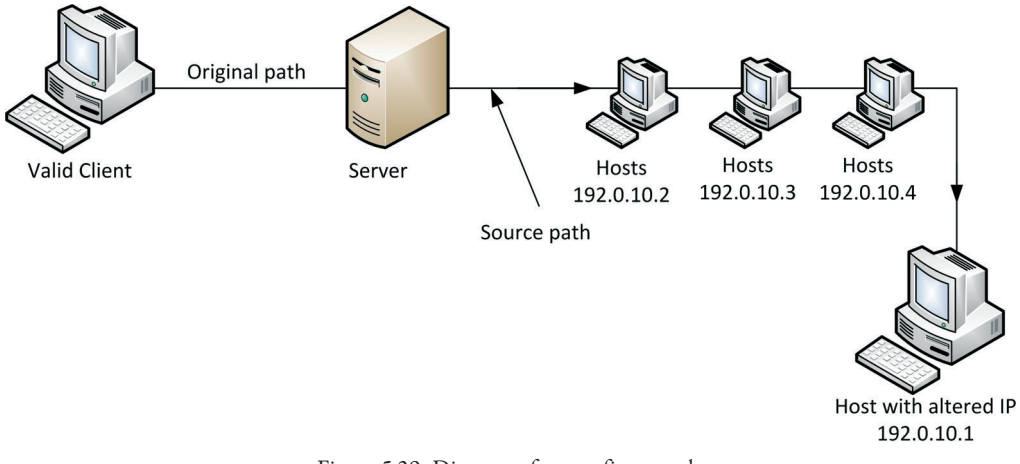


Figure 5.39 Diagram of a spoofing attack.

2. The receiving host verifies the identity of the sending host, which is an identity that has been suitably falsified by the attacker and valid for the receiving host.
3. The attacker enters the email system and can falsify, read or remove messages.

IP spoofing attacks are relatively difficult to detect, unlike desynchronisation attacks that tend to leave characteristic signs on the network in terms of traffic carried out. If the network can check incoming traffic on the router connected to the Internet, authentication of incoming traffic should be performed. When the audit trail traffic is performed, recording of the traffic itself is always available. By using audit recording, we can check all incoming traffic in order to identify packets in which both the source and destination addresses are within the local domain. In this sense, there should be no packets coming from the Internet characterised by an internal source and from an internal destination because this would reveal the presence of a probable *spoofing attack*. Within a packet, which is part of a *spoofing attack*, usually both the source and destination addresses refer to internal computers while the packet itself comes from outside. The best defence, therefore, consists of the filtering of these packets through the router connected to the Internet, which is a function that is supported on most routers. If the inbound router does not support this functionality, it is possible to provide the installation of a second router, located between the first router and the internal network, dedicated only to packet filtering.

5.7.7 Hyperlink spoofing

The attacks seen so far require the use of TCP and Telnet. The attacks described in the following, on the other hand, use the HTTP.

We have seen previously that HTML pages are formed of hyperlinks that inform the browser of very specific Web page views. An attacker, to conduct a hyperlink *spoof attack*, must change a link on a Web page to direct users of that page to the page of another site through which it performs an operation that is not lawful. Just imagine an attack of an online sales site focussing on the link that leads to payment; in this case, when we go to make a payment, we are diverted to another site that is exactly the same as the original, with the difference being that our credit card data necessary to make the payment is provided to a subject attacker who can use it fraudulently.

If an attacker is able to access a Web server, he/she is able to change a link in such a way as to point to a page that is located on another site, a page on which the same has displayed possible forms for the fraudulent collection of information.

A very simple method by which an attacker can change the connection is to look for the pages on a server containing typical phrases of purchases and then change the HTML file in such a manner that the links point to another site. Because Web pages are changed, the administrator of the site can detect the problem due to the decrease in traffic on the site itself, with particular reference to the payments. The administrator, at a later date, can learn the identity of the hacker through the altered connection.

To attack the site in a less visible manner, the attacker can use a sniffer to read the messages between the server and the users and decide which is the most opportune moment to intervene in the attack (perhaps in the case of large payments), replacing the payments Web page; in this case, the probability of being intercepted by the administrator is undoubtedly greatly reduced.

In addition, if the attacker is able to intercept the packets sent to the user before the user has selected a secure page, he/she is able to generate a secure connection with the user itself, which will be further deceived, believing that he/she is within a secure connection.

5.7.8 Web spoofing

Web *spoofing* is another type of attack. In this case, the attacker generates a false and very similar, if not, even equal copy of a website. The website has all the same features as the original site with the

difference being that it is under the total control of the attacker and all traffic between the user and the website is read without difficulty by the attacker. Figure 5.40 shows the schematisation of this type of attack.

In this type of attack, the attacker can read and edit all the information that is transmitted by the victim to the website and vice versa. Once the attacker is in possession of the desired information, he/she may launch a more appropriate attack.

Methods that are more used for this type of attack are *sniffing* and *spoofing*. In Web *spoofing,* the attacker records the content of the Web pages that are being read by the victim. When the victim fills in a form on an HTML page, the browser of the same transmits these data to the Web server. As the attacker is positioned between the client and the server, he/she is able to read all the data being transmitted from the client to the server. In addition, the attacker can read the content and the responses that the server sends to the client. Because the vast majority of business services on the Internet use Web forms, the attacker is able to learn sensitive and confidential information such as credit card numbers, numbers of bank accounts, passwords and all other data that the victim writes in the relevant areas.

The attacker can intercept the victim even if the latter thinks he/she is in a secure location, because the attacker can simulate the Web page of a secure connection, whether he/she is using SSL or S-HTTP, showing the relevant icons on the page, represented by a key or a padlock.

A Web *spoofing* attack *can* be conducted only if the Web server of the attacker manages to come between the computer of the victim and the rest of the Web, ensuring that the striker represents the interface between the victim and the Web. Once the server of the attacker has recovered the document requested by the victim from the Web, the attacker rewrites the URL addresses within the document, returning the changed page to the victim. As all the URLs point towards the server of the attacker, any address follows the victim, the latter always remaining within the Web server of the attacker, without being able to leave.

When filling in a form of a page of a fake website, we do not have the slightest suspicion that the information we have entered does not reach its correct destination. In this sense, Web page *spoofing* works without any problems as the basic protocol of the Web integrates the modules very closely with the actual pages. The browser encodes the transmissions of Internet modules within HTTP requests and a Web server responds to the requests of the modules using HTML. In the same way in which hackers replace URL addresses, they can also replace Web addresses and in the same way a Web page makes reference to the server of the hacker, also the Web forms that are compiled by the victim are sent to the server of the attacker. In this way, the server of the attacker can read and edit all the information

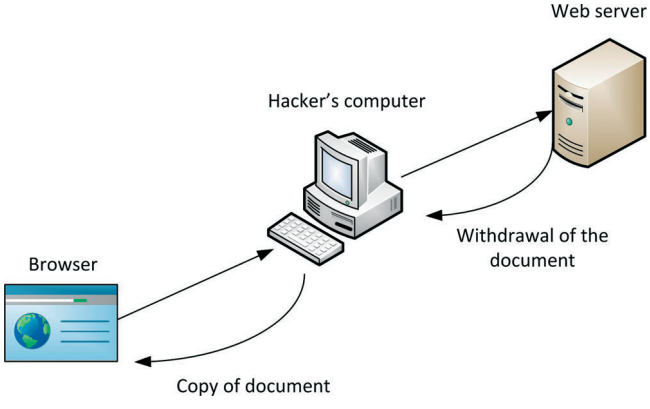


Figure 5.40 Schematic representation of a Web spoofing attack.

entered in forms before sending them to the real server. The attacker can also operate against the data that the true server sends to the victim.

Web *spoofing* is further deceptive because it can also use secure connections, helping to increase the sense of security that the victim has with the site that he/she is using. Figure 5.41 shows the relative connection scheme.

It is very difficult to escape from a Web *spoofing* attack once the latter has begun. In any way such an attack, for commencement, must request preparatory actions from the victim. To begin the attack, the attacker must somehow attract the victim to a false website, encouraging the victim to select a false link. The attacker can make the false link easily accessible in various ways:

1. He/she can insert the false link on a popular and notoriously secure Web page.
2. He/she may also send the victim an email message containing a pointer to the false link, hoping that the victim selects it.
3. He/she can send an email message containing a page of the false website.
4. He/she can fool a Web search engine and have the false website indexed.
5. He/she can write an appropriate command (such as ActiveX if the victim uses Internet Explorer) that is run every time the browser is used. This command can replace a correct URL address with a URL address of the attacker.

In practice, the attacker must explore all possible roads to lure the victim to his/her false website to then begin the attack.

As the attacker must convince the victim to enter the false website, the Web *spoofing* attack should be very convincing. If the attacker has not been attentive or if the victim has disabled some of the options on the browser, the false Web pages will show certain information to a careful observer, which is useful for discovering the deception, on the status bar. For example, most browsers show the destination URL in the status line every time that the mouse pointer is over a link contained in a page and this could reveal the actual address of the false link. An attacker that has been discovered could eliminate this operating mode in order to best deceive the victim. An attacker could use Javascript, Java and VBscript to manipulate the status bar of the Web browser in order to disguise the attack as far as possible.

The Web *spoofing* attack can leave specific signs on the status bar that can be considered as very important clues. It has already been said that when the mouse passes over a link, the corresponding pointed address appears on the status bar, which might reveal to the victim the true link being pointed at. In addition, when the browser is extracting a Web page, the status line momentarily displays the name of the server that was contacted by the browser and this could provide additional clues for the victim. In this sense, the attacker can make use of Java, Javascript or VBscript programs on every false page and the above-mentioned programs would eliminate each of the above actions. In addition, the attacker could ensure that his/her programs always show the status line of the true website, making the false site very convincing.

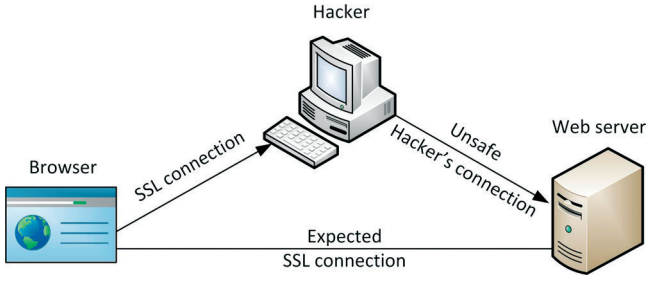


Figure 5.41 Diagram of secure connection during a Web *spoofing* attack.

The address line can also provide valuable information about an attack in progress. Once the victim has written the URL address of the site that he/she intends to visit, if the attacker has not made any modification, this line will display the new URL address to which the attacker wishes to divert the victim. If the attacker has suitably designed the attack, he/she can hide the true address and replace it with the URL address that the victim expects to visit. In practice, the program of the attacker writes the address that the victim wrote, before hijacking the latter to the new site from which to lead the attack.

An expert victim, if suspicion is aroused, can make use of the most advanced controls and may want to examine the source code to verify if the latter contains rewritten URL addresses. To avoid this, the attacker can use another program that hides the menu bar of the browser by replacing it with a menu bar that is similar, if not equal, to the original one. When the victim selects the option *View Document Source* from that bar, the attacker can show a window that displays the original HTML source code.

Further information about an attack in progress can be seen when accessing information on the document. If the victim selects the option *View Document Information*, the browser shows the information in the document that also contains, among other things, the URL address. But, in the same manner as seen before, the attacker can provide a suitable program that shows an altered menu bar from which to display, upon request, the original information. In practice, the attacker can suitably hide all sources of clues that can arouse suspicion in the victim that he/she may be on a false website. The only way to defend ourself from such manipulation is to disable any form of language script in our browser.

From what we have seen so far, it is evident that Web *spoofing* attacks are very dangerous and difficult to identify. However, there are precautions that can be taken by users to prevent this type of attack, which include the following steps:

1. Disable Java scripts, JavaScript and VBScript in the browser in such a manner that the attacker cannot hide information that would reveal to the victim that he/she was on a false site.
2. Ensure that the address line of the web browser is always clearly visible.
3. Always pay close attention to the URL address that is shown in the browser, in order to check that the same always points to the expected address.

This technique of counter-attack dramatically reduces the chances of Web *spoofing* attack. A big problem is the possible presence of Java scripts, JavaScript, VBScript and ActiveX that, even if in general can ensure the advanced features in display, in cases of attack can be deadly tools in deceiving the victim. The best solution is to disable these languages in normal navigation and then to reinstate them when surfing on a secure site.

To solve the problems that this type of attack can cause, it would be necessary, for browser producers, to perform changes to their code in such a manner that the program always displays the line of the address or makes it impossible to create false status bars, false menu bars and so on. In any case, these solutions always require the user to pay a certain degree of attention to the status of his/her browser during navigation and thus a certain level of competence on his/her part. In any case, a browser that is not influenced from the outside through special programs may represent a good starting point for secure navigation and solid defence against Web *spoofing* attacks.

5.8 Authentication

Authentication, together with the cryptography already illustrated in detail in the relevant chapter, ensures the protection of data. Authentication is the process that ensures that both subjects in communication are who they claim to be. Authentication concerns not only the entity that tries to access a service but also the device that provides the service.

Both authentication and cryptography have specific purposes in the protection of a communication session and the highest level of protection can be achieved by combining them together. For this reason, most security protocols are characterised both by components aiming at authentication and by components aiming at cryptography.

When in the 1970s IP version 4 (IPv4) was created, which is still in use, security was not a design specification and for this reason, the same was pushed to the sidelines. Even if the security of a system was considered to be important, no particular attention was given to the process of communication and thus the first version of IP contained no security measure and did not envisage that users would protect their data during its transmission. The new IPv6 specification, which is currently replacing IPv4, contains, on the contrary, security specifications.

IPv4 transmits data in plaintext, that is the same is not coded but remains as it is. The same is true for authentication. This means that with a normal network analyser, it is possible to read all the communications that occur on the network itself. Network analysers can be dedicated hardware tools or by software tools that run on a particular operating system. In the latter case, there are network analysers for Windows or for Macintosh at relatively low cost and even free for UNIX. Network analysers operate as passive instruments, not transmitting any data on the network but only reading what passes. However, there are network analysers that, in order to locate a particular device on the network, are also able to transmit on the actual network. A network analyser usually does not even need an IP address, and can carry out his/her work without being detected. A potential attacker can also attack a given system by loading software of a network analyser over it: in this way, the attacker does not need physical access to the system to control the traffic, being able to use the system under attack to read the traffic. For this reason, it is extremely important to perform periodic checks on all the systems: in this way, *sniffing* activities can be avoided that are introductory to more dangerous attacks.

In order for a network analyser to be able to capture a given communication session, it must be connected to any point of the path of the session. This situation can be created thus compromising, as a minimum, one of the systems of communication that can be found on both sides. For this reason, the attacker cannot acquire the network traffic from outside the network, passing through the Internet, but must physically connect the network analyser within the same.

There are many services that operate in plaintext. Most of them are not the owners and were not designed to provide services for authentication and encryption and for this reason they operate in plaintext. They are: POP3; FTP, in which authentication is in plaintext; Telnet, in which authentication is in plaintext; SMTP, in which the contents of the messages are delivered in plaintext; HTTP, in which the contents of the pages and the form fields are transmitted in plaintext; IMAP, in which authentication is in plaintext; SNMPv1, in which authentication is in plaintext. In particular, for the latter service, significant security problems can become apparent as it is used to manage and query network devices such as routers, switches, servers and firewalls. If our SMTP password is compromised, a potential attacker can do whatever they feel is best on the network. SNMPv2 and SNMPv3, on the contrary, guarantee a higher level of security and integrity of data with respect to SNMPv1. The problem is that not all devices support later versions of SNMP, exposing the network to significant risks.

Good authentication is of primary importance to ensure the security of a system. Plaintext authentication (logon) should be avoided at all costs because the data flow can be easily intercepted and could become more dangerous in all situations where the frequent replacement of passwords is not expected. In such a case, an attacker would have all the time necessary to prepare a more aggressive attack. Another even more critical aspect is the fact that most users have a tendency to use the same login name and password for access to different accounts: this is very risky because a potential attacker, that has fraudulently acquired credentials for access to an account, may use the latter to gain access to all of the user's account, including those that are most critical and sensitive.

Validation of the resource that tries to access a service is very important and, in particular, we should check that the resource has not been replaced by an aggressor host during the communication session, giving rise to what is called session hijacking, already illustrated in the previous sections.

In addition to authenticating the origin before and during a communication session, it is also very important to check the destination server. Most users take for granted the connection to the requested server, without thinking for a moment that the same server may be under attack.

5.9 Virtual Private Networks

VPNs are not only considered to be a remedy to the high costs of a WAN but also an element of security vulnerability of the perimeter of a network.

A virtual private network session or VPN is an authenticated and encrypted communication channel on a public network, such as the Internet. As the public network, by definition, is not secure, both authentication and cryptography is used to appropriately protect the transmitted data. Usually a VPN does not depend on the service and thus all the information exchanged between two hosts, regardless of whether they are Web, FTP, SMTP, etc., are transmitted through an encrypted channel. Figure 5.42 shows an example of VPN between two Internet sites.

As can be seen in Figure 5.42, the two networks are connected to the Internet and intend to exchange data in a secure manner as the same are private. To do this, a VPN is established between the two sites.

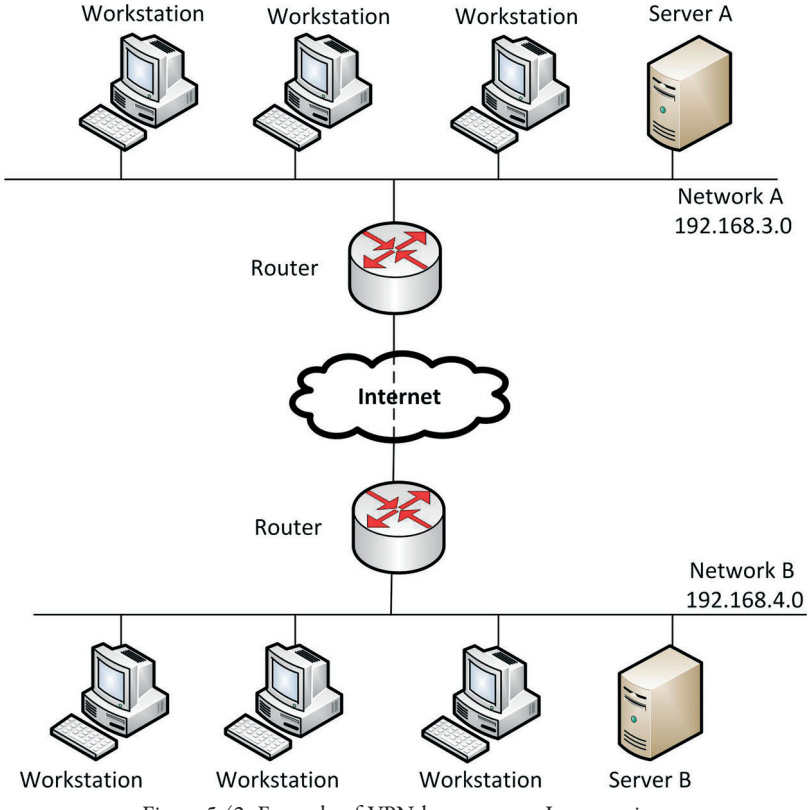


Figure 5.42 Example of VPN between two Internet sites.

VPN requires a minimum of preliminary design. In fact, before making decisions on a VPN, the following points should be considered:

1. Each site must have a device compatible with VPN on the perimeter of the network. This device can be represented by a router, firewall or a device dedicated to the VPN activity.
2. Each site must know the IP addresses in the subnet present at the other site.
3. Both sites must use the same authentication technology, using, as the case may be, the exchange of digital certificates.
4. Both sites must use the same technique of cryptography, using, as the case may be, the exchange of cryptographic keys.

As can be seen in Figure 5.42, at each end of the VPN tunnel, routers are positioned that are used to connect to the Internet. The router of network A must be configured in such a manner that all outbound traffic towards network B is encrypted, using, for example, Data Encryption Standard (DES). This is called remote domain encryption. This router must, in addition, decode all incoming traffic from the router of network B. In a similar manner, the same is applicable to the router of network B. On the contrary, all data transmitted to other hosts on the Internet, are transmitted in plaintext.

In some VPN configurations, it is possible to enter a network analyser between the two routers in order to see all of the packets that use as source and destination IP addresses those of the interfaces of the same router. It is not possible to see the IP address of the host that has transmitted the data or the IP address of the destination host as such information is encrypted with the data of the original package. Once the original packet is encrypted, the router encapsulates the same in a new IP packet using its own IP address as source address and as destination IP address of the remote router. This procedure is called *tunneling*. Thanks to *tunneling*, a potential attacker cannot know what traffic to attack on VPN, because all the packets use the IP addresses of the two routers. Not all the VPN methods are able to support this feature, but the latter must be used whenever available, in order to decrease the level of risk.

Since a virtual tunnel between the two routers is available, a private address space can be used on the Internet. For example, a host that is connected to the network can transmit to a host of network B without having to resort to the translation of network addresses because the routers encapsulate information about the header when the data are inserted into the tunnel. When router B receives the packet, it extracts the encapsulated packet, decodes it and sends it to the destination host.

We have already said that VPNs are independent of platforms and services, and to perform secure connections they do not need computers on the network to have encryption software installed as such work is performed by routers. This means that services such as SMTP that are in plaintext can be used securely as long as the destination host is on the remote cryptography domain.

VPNs are currently used intensively; however, there are two specific applications for which they are most commonly used. They are: replacement of battery of dial-up modem and replacement of dedicated WAN links.

A VPN can safely perform the replacements referred to above in an integral manner or only in specific situations. A VPN requires, in general, a certain amount of time to dedicate to manual tuning of configuration and communication parameters.

With regard to the replacement of remote access modem stacks, VPNs are able to significantly reduce the cost of implementation; in fact, groups of remote access modems are a very complex element to manage by network administrators. It often happens that we have to deal with auto-response modems, with wiring characterised by low-quality, research groups not correctly configured and access issues. Thanks to VPN, there is no longer any need to rent phone lines, perhaps with free access to internal users, and there is no need to upgrade hardware every time modems characterised by new standards are used or whenever the communication technology on the line is changed. Thanks to VPN, all incoming traffic is managed through connecting to the Internet, which in most cases already

exists, to be used by other applications. Thanks to access via the Internet, it is no longer necessary to maintain access numbers free of charge for employees.

Thanks to VPN, it is also possible to reduce support costs to end users. In fact, in the past, it was necessary to have internal staff available to resolve all the issues of access via modem while if we pass through the Internet and any connection problems will be resolved by the ISP while any remaining problems relating to internal connections will be resolved, as usual, by internal staff that in any case will be found to have a reduced work load.

If a firewall is being used, it is very important to choose the products that support the VPN. The majority of firewalls come with a client packet that allows the creation of VPN within the firewall itself.

However, there are disadvantages that can occur when remote access service via VPN is provided.

The first disadvantage is the integrity of the remote location that can, in any case, be compromised by external attackers through the use of appropriate tools, the majority of ISPs having no firewall for remote users. For this reason, remote access systems are particularly vulnerable and an attacker can enter the remote client and use the tunnel to enter the internal network and attack it.

The second disadvantage is the opening of the firewall that must be performed for VPN access: this opening represents a vulnerable passage that can be used by attackers to penetrate within.

With regard to the replacement of dedicated WAN links, it has already shown above that a VPN can also be used to connect two networks via the Internet that are geographically separate, whether they are near or far.

Now consider the network diagram shown in Figure 5.43.

Figure 5.43 shows the diagram of a network using dedicated links to safeguard the security of important services. As can be seen, there is obviously a firewall and a part of the network of the type DMZ in which there is a Web server and an SMTP relay. A further dedicated network card for remote connections that are used to transmit data to remote locations without having to pass through the Internet is also present on the firewall. Such a configuration, which is relatively simple, may,

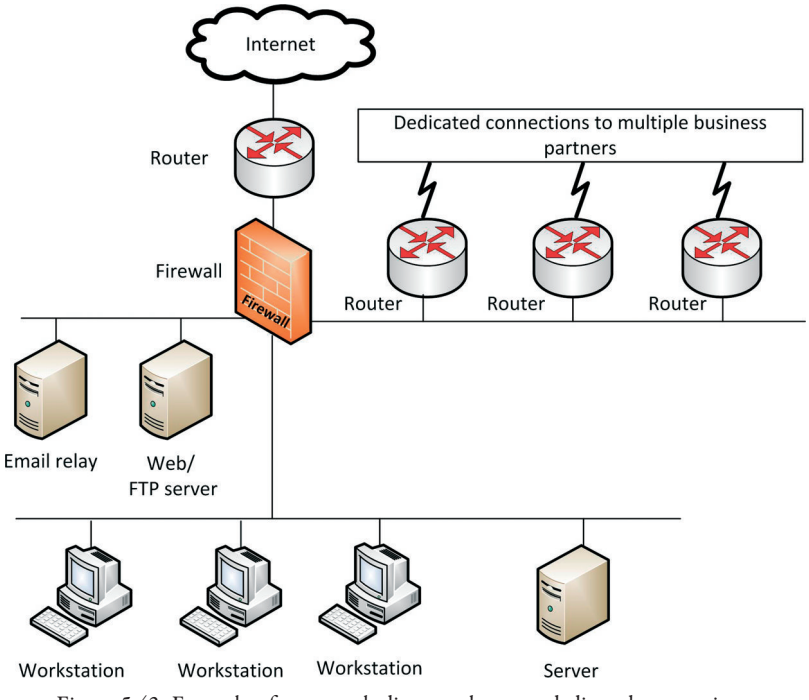


Figure 5.43 Example of a network diagram that uses dedicated connections.

unfortunately, present security problems. The first is represented by routing, since the firewall must be properly programmed with all the necessary information to allow connections to every remote location because, if the same were referring to the default settings, all traffic would be sent over the Internet. In this case, for any variation in the characteristics of remote users, it is necessary to reprogram the firewall. Problems can also become apparent with IP addresses if one of the remote networks were to use NAT with a private address space. In this case, if we were performing a DNS search on these networks, we would receive as a response the public IP address and not the private address of the remote user that was intended, and in this case routing problems would arise that can be solved by registering the DNS entries locally. Further problems would arise if the remote networks were using the same private address space and it would be necessary to perform NAT on the router to the end of a network connection in such a manner that the hosts would be able to see the networks as separate.

A further problem is the possibility that a remote attacker, passing through that network, would be able to penetrate the network of another remote user: this would also entail conflict of allocation of responsibilities of system security flaws.

From the reasons given above, it is clear why the use of VPN is recommended, which supported the ease of managing multiple VPN compared to several dedicated circuits, referring to the fact that, even in this case, it is necessary to operate a further opening in the firewall to ensure that the VPN traffic, even if it may be protected by a good authentication system, is always a possible passage through the perimeter of the network.

When intending to gain access to the network via VPN, particular attention must be paid to the ability of the system taking into account the following factors:

1. Numbers of concurrent users, in that an increase of the latter must also increase the capacity of the system.
2. Time for the connection of the users, in order to guarantee the necessary services at the same time.
3. Services to which remote users access as services characterised by a significant bandwidth consumption require particularly fast Internet connections and hardware.
4. Type of encryption that is used, as the algorithms vary so does the ability to request calculation and therefore the consumption of resources that needs to be carefully guaranteed to prevent an unacceptable slowdown of the performance of the system.

5.9.1 The choice of a VPN

In choosing a VPN, it is very important to pay attention to the following characteristics: excellent level of authentication, excellent level of cryptography, adherence to standards and integration with the other network services.

With regard to authentication, this is very important because it allows us to know and verify the identity of the subject that is located on the other side of the VPN. In this sense, the Diffie–Hellman algorithm is greatly used to identify the two subjects that are located at either end. As this algorithm allows the generation of a shared secret through the exchange of public keys, this element avoids the need to exchange confidential information by resorting to channels of auxiliary transmission.

With regard to cryptography, it is very important to choose the level of protection desired before proceeding with the selection of cryptographic method. For applications with low security, a 40-to 56-bit DES key may be enough, but for information that requires a high degree of security, it is necessary to use at least a triple DES or Advanced Encryption Standard (AES) with a much longer key. The level of encryption must be chosen by also taking into account the performance of the communication channel, as the greater the power of the cryptography algorithm, the greater the time required for encryption and decryption of information: if the channel is too slow, there may be a risk of occurrence of the various timeout of applications. It is evident that the greater the length of the key, the higher the level of security, but, even in this case, a cost/benefit analysis must be performed. Also the type of

algorithm used influences, of course, performance, since, as has already seen, public/private key cryptography, for example RSA-based, for the same length of the key, can be from 10 to 100 times slower than symmetric key cryptography as the first requires greater processing time. Most VPN solutions use the technology, now consolidated, of exchanging a symmetric key at the initial phase by means of a public/private key algorithm, and then continue the communication by exchanging and encrypting the information using the symmetric key exchanged.

With regard to compliance with the standard, it is very important, as has already been stated above, that the cryptography algorithms that are used are both well known and publicly verified. It is very important to use algorithms for cryptography that have been in use for a long time, the vulnerabilities of which are known. For example, it has already been seen that the vulnerabilities of DES are represented by the length of the relatively short key that does not ensure a high level of protection from brute-force attacks that can be carried with current computing resources: in this sense, to avoid this problem, it is possible to resort to a triple DES or an AES with relatively long keys. It is also very important to ensure that the VPN is compatible with other VPN solutions present on the network.

With regard to integration with other network services, new generation VPN can be integrated with other devices and systems, such as firewalls, user directories and control software. A very important function is the ability to centrally manage the authentication of VPN connections and to check the amount of bandwidth allocated to each connection. In addition, the VPN should be able to integrate with devices produced by different manufacturers.

5.9.2 Various VPN solutions

There are many solutions available for VPN that can in any case be categorised into three groups: firewall-based VPN, router-based VPN and software- and hardware-based VPN.

With regard to the firewall-based VPN, the configuration most used is the integration with firewalls. In this case, a firewall must be directly selected that supports VPN, in this case being able to count on a central reference both for the protection of the network and for the realisation of VPN. However, there is one disadvantage, that is performance, in that if there is a lot of Internet traffic and there are many encrypted VPN connections, the device can easily become overloaded. To avoid this, there are devices equipped with dedicated expansion cards engaged in the activities of traffic encryption and decryption.

With regard to router-based VPN, they are very useful in that the transmissions are decoded before reaching the firewall, leaving the latter free to carry out its main activity. There is always the problem of high workload that can be remedied by resorting to devices that use dedicated hardware. The downside is security, since the routers are less suitable than firewalls for the purposes of the protection of a network and a potential attacker could carry out *spoofing* beyond the router, leaving the firewall to believe that this traffic is coming from a VPN, allowing the attacker to enter and access services that are not normally available to external Internet users.

With regard to software- and hardware-based VPN, they are very useful when we already have a firewall or a router that does not support VPN and we want to implement them in our own network. We must obviously use independent products in such a manner that no incompatibility problems will occur. The disadvantage is the creation of another control point that must be administered and managed. If the solution is positioned outside the firewall, there are the same problems of possible vulnerability to spoofing of the router-based VPN. If, on the contrary, the solution is placed inside the firewall, problems of conflict with the firewall security policy may be an issue, not allowing the latter to create VPN. The vast majority of the solutions in question encrypt all the data packets, and the firewall therefore lacks information about the IP addresses for the purpose of traffic control and the latter moves from one side to another of the tunnel using the same encapsulated header packets. In this way, the firewall is not capable of making a distinction between an encapsulated telnet session in the tunnel

and an SMTP session and should rely on the options to control the traffic that are offered by the VPN solution.

Many remote access solutions do not require a VPN with full functionality. The disadvantage of these solutions is the necessity of running on the client dedicated software to initialise the connection and there is no more independence from services like a true VPN solution.

5.9.3 Setting up a VPN

In the following, we illustrate the basic procedures for setting up a VPN. These procedures are applicable to most products.

First, the VPN must be configured by defining the domains of cryptography at each end of the VPN tunnel in such a way as to identify with which remote networks the firewall will be required to exchange encrypted data.

Then it is necessary to configure the network objects requested and the first step consists of the creation of the requested objects. Each of the firewalls must have the following items:

1. a network object or a group of networks for the local cryptography domain;
2. a network object or a group of networks for the remote cryptography domain;
3. own location object;
4. a location object for the remote firewall.

Once we have created the network objects, we need to exchange keys and for this activity we will need the functionality offered by the selected solution.

Once we have completed these tasks, we must define a set of criteria in such a manner as to allow the firewall to use the VPN.

Once we have done this, we can check the VPN tunnel and check the flow of data. To check with efficiency the flow of data, we should make use of a network analyser, because the same allows us to read the contents of the packets and, since the traffic is encrypted, it should not be possible to read anything.

5.10 The exchange of Kerberos keys on distributed systems

The exchange of Kerberos keys represents an extremely functional and secure way to introduce useful functionality to a network. This system has been partially illustrated in Chapter 2 and will be further explored in this section. It was developed in 1980 by the Massachusetts Institute of Technology (MIT) in the context of the Athena project in order to manage access to distributed systems of a certain size using various security measures and the one trusted system.

A Kerberos system uses a secure computer, possibly located in a secure environment, and guarded 24 h a day on which is found information relating to the password and to access levels of the system users. All of the computers on the network use the information on this computer that is usually a server. This server, also called trusted server, is the only one that can provide the network programs with access information to resources and servers that are administered by Kerberos and users consider the rest of the network as characterised by a lower level of security. Kerberos is based on the principle of simplicity, in which it is assumed that it is extremely easy to have a server that is highly secure and more difficult to make the computers on the network secure.

Before proceeding further, it is very important to illustrate, albeit briefly, distributed systems. A distributed system consists of a network to which are connected one or several servers and one or several work stations. It is understood that this system should use secure computers, even if this is not explicitly stated. As computers that are part of the distributed system are positioned in different areas,

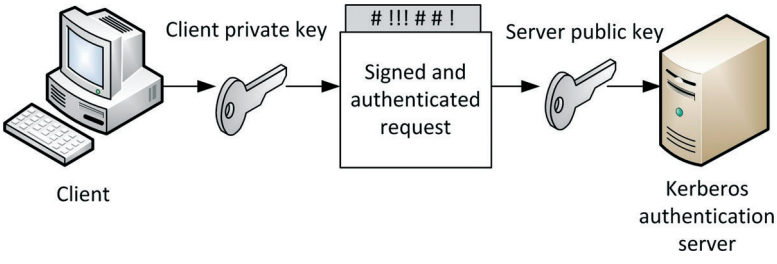


Figure 5.44 The client sends the encrypted message to the trusted server.

it is very difficult to ensure that each computer is practically secure. In this case, an attacker can physically access one of these computers and use it to attack the network. Kerberos is based on the fact that by making a single computer secure and using the appropriate security procedures, it is possible to make the whole system secure.

If a computer needs to access a file server or another server belonging to the network, it must first of all ask the trusted server for permission. Below is illustrated an operation operating mechanism, assuming that a computer wants to access a file on a server connected to the network. If the computer, or client, is already connected to the network, the software will perform the following operations:

1. Access request signature for the requested file with the client private key and encryption of the message with the server public key. Use of the server public key ensures that the request can only be read by that server. The trusted server uses digital signature to verify that the message was actually sent by the client and not by an attacker.
2. The client software sends the message to the trusted server, as shown in Figure 5.44.
3. The trusted server analyses the login message, verifies the identity of the client by using the digital signature and checking at the same time if the client is authorised to access the desired file.
4. If the verification succeeds, the trusted server connects the client to the file server and informs the latter that the client has access to the file.

To begin the communication between the client and the file server, the trusted server performs the following steps:

1. The trusted server sends the client a unique key called ticket. In order to avoid interception, the trusted server encrypts the ticket using the client public key. The ticket contains the login information and a session key. The session key is constituted by a simple encryption key that is used to contact the file server. This step is shown in Figure 5.45.
2. The trusted server sends a copy of the ticket to the file server. To ensure maximum security, the trusted server encrypts the second copy of the ticket using the server public key. The second ticket is substantially equal to the first ticket. The situation is shown in Figure 5.46.
3. The client and the file server containing the requested file connect and compare their tickets with

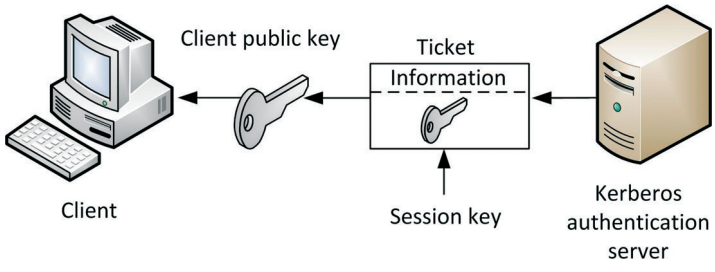


Figure 5.45 The trusted server encrypts and then sends the ticket to the client.

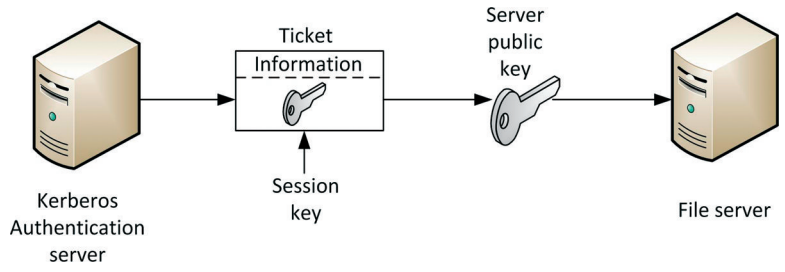


Figure 5.46 The trusted server encrypts and then sends the client ticket to the file server.

each other to prove their identity. The client network software encrypts its own copy of the ticket by using the public key of the file server and transmits the encrypted ticket to it. The file server decrypts the ticket using its own private key. If the copy of the session key that is transmitted corresponds to the copy sent to the trusted server, the file server authenticates the ticket. If the tickets match, the client is allowed to connect to the file server. Failing this, the connection will be rejected. The situation is shown in Figure 5.47.

4. If the tickets match, the trusted server has completed its task and the client communicates with the file server using a secure channel in order to access the requested file. The file server, depending on its settings, can send the encrypted file with the client public key, which is available at the trusted server, or with the session key or in plaintext. This situation is shown in Figure 5.48.
5. Once the file server has completed the transmission of the requested file to the client, it sends to the trusted server an informational message about the completion of the access and to authorise the same to destroy the ticket. In this way, if the client tries to use the same ticket for a new access, the trusted server will prevent it.

The Kerberos server is the only trusted server of the network. It may also be considered as a method for verifying the identity of the persons operating on an open, unsecured network. Kerberos verifies the identity of these subjects without recourse to the authentication that is performed by the underlying operating system, without trusting the host addresses and without requiring the host to use specific mechanisms of security. Moreover, Kerberos assumes that any person can read, modify or insert

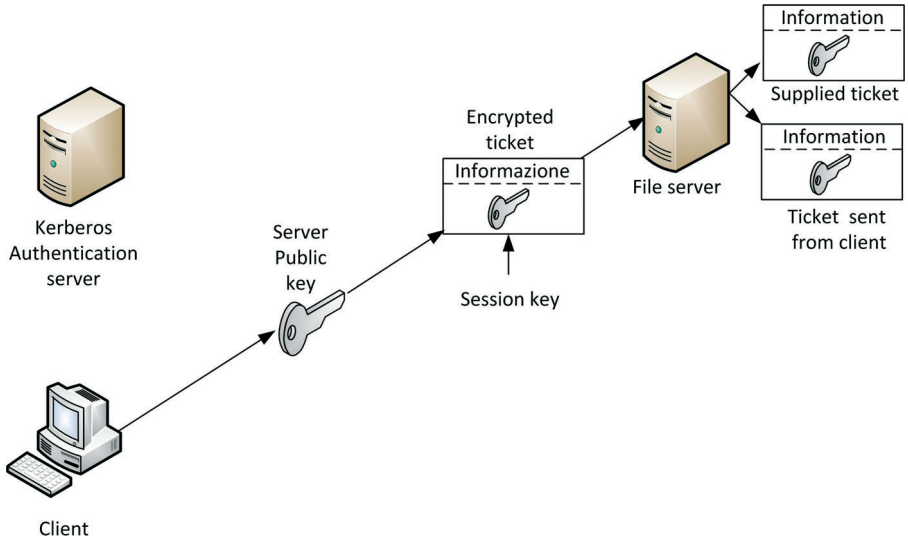


Figure 5.47 The client and the file server compare tickets.

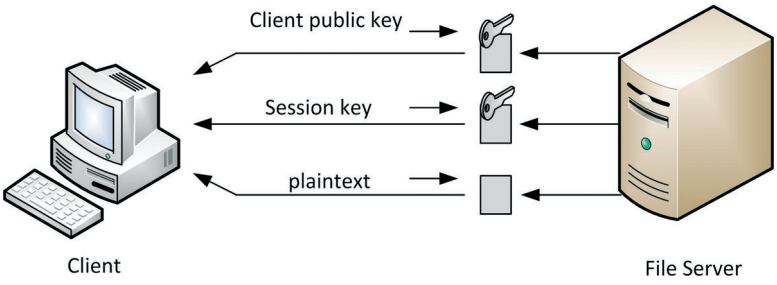


Figure 5.48 Transmission mode of the file to the client.

physical packets into the network, considering the entire network as a single area of risk. The same assumes that an attacker can intercept all transactions and that the latter is constantly trying to breach the network. Given this, Kerberos authenticates all transactions, representing a secure authentication service. It uses traditional methods of cryptography to authenticate transactions.

The trusted server authenticates all transactions that take place within the controlled network. It is the only computer in the network that can authenticate transactions. The authentication process takes place according to the following steps:

1. The client sends a request to the trusted Kerberos server requesting the credentials of a certain server.
2. The trusted server controls the level of access of the client to check if it is authorised to access the server and the resource that has been requested.
3. If the client does not have access to the server or the resource, the trusted server rejects the client's request and provides no ticket to the client.
4. If the client has access permissions, the trusted server transmits to the client the ticket encrypted with the client public key. This ticket also contains the temporary encryption key.
5. The trusted server encrypts the ticket using the server's public key of the resource and sends to the latter the encrypted ticket. The server decrypts the ticket and waits for the client's transmission.
6. The client encrypts the session key with the server public key of the resource and the ticket itself transmits it. This ticket contains the identity of the client and the encrypted copy of the session key.
7. The server of the resource decrypts the session key using its private key. Then, it compares the session key received from the client with the key received previously from the trusted server: if the keys match, the server of the resource authorises the client. The client can also use the session key to authenticate the server. At this point, the two subjects can use the session key to symmetrically encrypt communications or may exchange a sub-session key that is used to encrypt the communication.

When Kerberos is implemented, one or more trusted servers are used that operate on physically secure computers. The trusted servers maintain an archive of Kerberos subjects, public key and private key of each subject.

The Kerberos protocol consists of several sub-protocols. It has already been said that a client must request credentials (ticket and encrypting key) from the trusted server every time the same attempts to access another server. A client can choose two different ways to ask a trusted server for credentials.

The first method consists of the client sending a request encrypted with its own public key to the trusted server requesting access to another server. The trusted server sends the client a limited ticket for access to the requested server that the same client then sends to the server. Even if this method is relatively simple to manage, it is not as secure as the second method that will be illustrated.

The second method consists of the client sending a plaintext request for a ticket. This usually applies to a ticket for the request of a ticket that consists of a sort of super-ticket for an entire

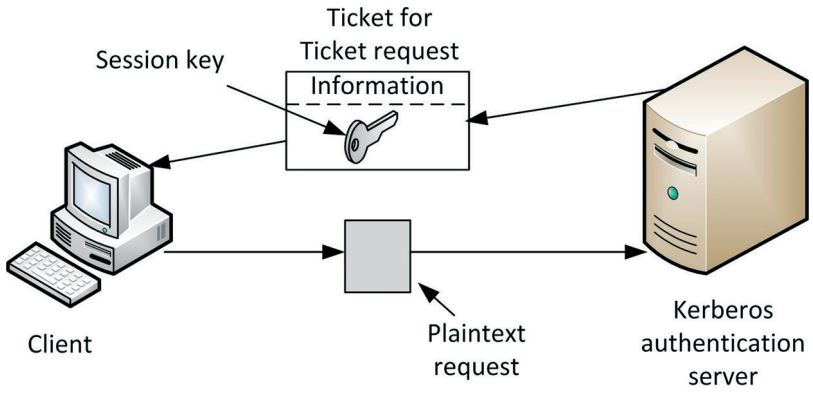


Figure 5.49 The client requests and receives a ticket for a request for a ticket.

connection session. The server verifies the identity of the client using the shared secret key, such as a password, and sends the client the ticket for the request of the ticket, as is shown in Figure 5.49.

The client can use the ticket for the request for a ticket instead of its public key to receive from the trusted server a ticket for access to the resource that it wishes to use. Figure 5.50 shows how a client sends a copy of the ticket for the request of a ticket to the trusted server receiving as a response an access ticket.

Once the trusted server responds to the ticket request of a client, it encrypts the access ticket using the session key that is located inside the ticket for the ticket request. The trusted server subsequently sends the access ticket. The access ticket contains the client's credentials and a dedicated session key for the rights that guarantee the ticket.

The first method, relating to the request of the public key, ensures lower security with respect to the second method as the ticket for the ticket request is not persistent, being valid only for a certain period of time and is therefore more difficult for an attacker to breach compared to a message encoded with a public key.

After the applicants have obtained the credentials, they can use them to verify the identity of the other subjects from the transaction in such a manner as to ensure the integrity of the messages exchanged by the computers or the privacy of the messages that are exchanged. The application decides on the type of protection.

In the management of a distributed network system such as Kerberos, one of the major problems is the theft of ticket or, in short, by the problem of *replay*. A *replay* occurs when an attacker is able to copy a ticket, to breach its encryption, decrypting it to impersonate the client, presenting the ticket again. Usually, tickets contain additional information that tends to discourage *replay*. To prevent *replay*, the

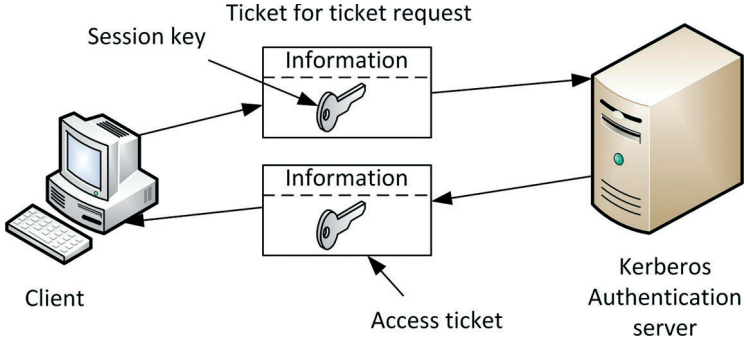


Figure 5.50 The client sends a ticket for the ticket request and receives an access ticket.

client sends new information to verify the origin of the message, encoding such information using the session key and including the date and time. The date and time are used to demonstrate that the client has produced the message containing the ticket recently and it is not therefore a *replay*. The fact of using the session key ensures the client's identity because the same key is known only to the receiver and to the server because the Kerberos server never sends the key in plaintext on the network. The client and the server are also able to ensure integrity of the messages exchanged using the session key. The same is used by Kerberos to detect both replay attacks and attacks that seek to disrupt the normal flow of messages. To ensure the integrity of messages using the session key, each client or affected server generates and transmits a hash or digest of the message that the client has encrypted with the session key.

It has already been said that Kerberos has been developed to operate in distributed environments. Since distributed environments can be composed of parties located at a great distance from each other, Kerberos is able to ensure its services even in very large and complex environments. When a Kerberos application must be designed, in order to check the areas in which a given trusted server is the only authentication subject, the concept of realm is used: every Kerberos server has its own well-defined realm. Network administrators can generate *inter-realm keys* in order to authenticate clients in remote realms. The *inter-realm key exchange* records each ticket supply service as a client or a server of the other *realm*. A client can receive a ticket to access tickets for the trusted server of the remote realm from their own local realm, as shown in Figure 5.51.

When users send the trusted remote server the ticket for the ticket request that they in turn have received from the local trusted server, the remote trusted server uses the *inter-realm key* to decode the ticket for the ticket request and to be sure that the ticket has been issued by the trusted server of the client. For the final service, the remote authentication ticket indicates that the service authentication server has used an *inter-realm* key to authenticate the client. This situation is represented schematically in Figure 5.32(Figure 5.52).

*Realm*s are able to communicate with each other if they share an *inter-realm* key or if the local *realm* shares an *inter-realm* key with an intermediate *realm* that is able to communicate with a remote *realm*. The intermediate *realm* sequence that an authentication request must follow in order to communicate from one *realm* to another is called authentication path. Figure 5.53 shows an authentication path between three *realm*s in the situation where the *realms* are not connected to each other.

Network administrators usually organise *realm*s in a hierarchical manner, that is each *realm* has a unique parent *realm* and may have several children *realms*. Each *realm* shares a unique key with its parent and a unique key with each of the children. If there are two *realms* that do not share an *inter-realm* key, it is still possible, by passing to the higher hierarchical levels, to create an authentication path between the two. If an installation does not use a hierarchical organisation, the trusted remote server

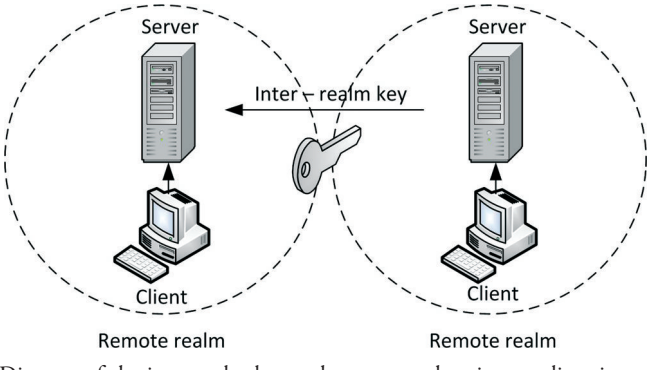


Figure 5.51 Diagram of the inter-realm key exchange to authenticate a client in a remote realm.

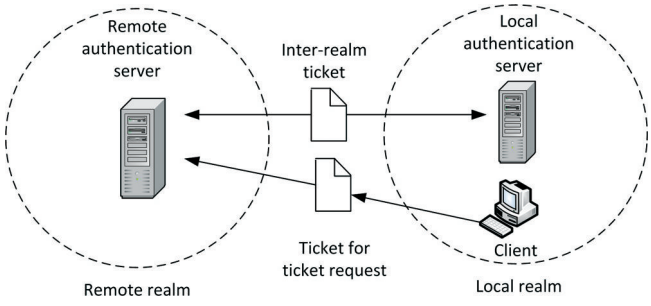


Figure 5.52 Schema of authentication by the trusted server of a remote client using an *inter-realm* key.

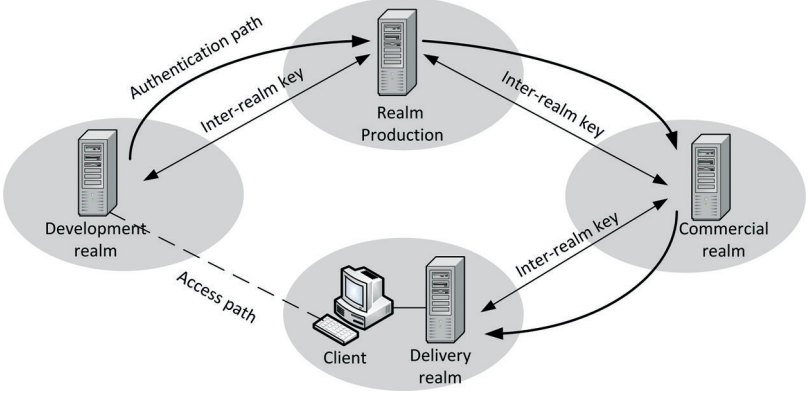


Figure 5.53 Example of authentication path between several *realms*.

must be able to access a trusted archive to create the required authentication path. Figure 5.54 shows how an authentication path between two children within a Kerberos hierarchical organisation is generated.

Realms are characterised, in general, by hierarchical relationships but a Kerberos server can bypass the intermediate realm in order to find an inter-realm authentication via alternative routes. These paths are determined by the administrator in such a way as to make communications between the two

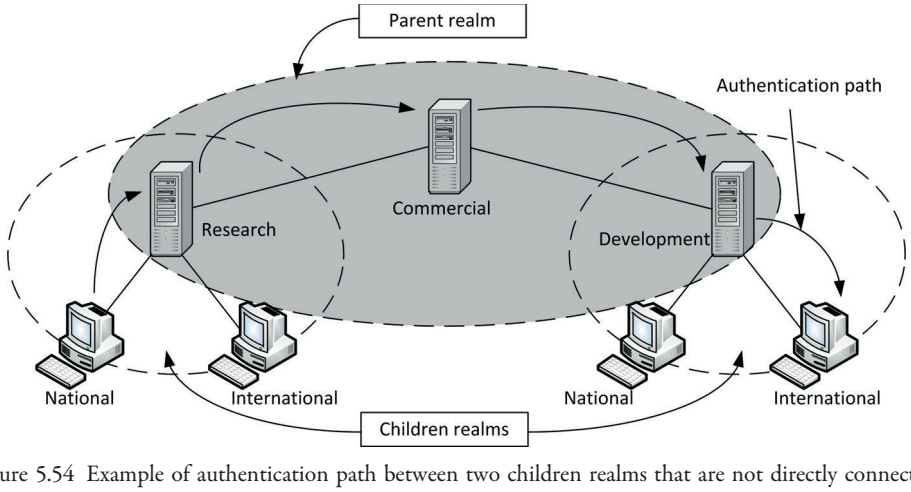


Figure 5.54 Example of authentication path between two children realms that are not directly connected.

realms faster. When the final service must decide on the degree of trust of the authentication path, it must know which *realms* are crossed by the path itself. To facilitate this operation, a dedicated field of each ticket that passes through several *realms* stores the name of the *realms* that are part of the authentication process.

5.10.1 Ticket flags

Each Kerberos ticket is characterised by a series of flags that are used to indicate the various attributes that the ticket itself may have. The Kerberos server automatically enables and disables various flags depending on its real needs and on the manner in which the client has received the ticket. Flags are:

1. initial;
2. pre-authenticated;
3. invalid;
4. renewable;
5. post-dated;
6. proxiable and proxy;
7. FORWARDABLE.

The initial flag indicates that the ticket was generated by a trusted server and that the server has not produced the ticket on the basis of a ticket for the ticket request. The application servers that intend to request the proof of knowledge of the secret key from the client can ask the Kerberos server to enable this flag, ensuring that the client has received the key recently.

The pre-authenticated flag provides the server with informational applications on the initial authentication irrespective of whether the trusted server has generated the ticket directly (in this case, in initial flags it is active) or whether it may have generated it as a result of a ticket for the ticket request.

The invalid flag is effective to indicate that the ticket is invalid and the same is, consequently, a refused application by the servers. Trusted servers raise this flag when they produce post-dated tickets (which will be discussed below). To allow the users to use their ticket, the trusted server must authenticate an invalid ticket. The client can validate their invalid ticket by making a special request to the trusted server. The trusted server cannot authenticate the invalid ticket if the validity start time has not elapsed. Validation ensures that the system can permanently render invalid each post-dated ticket that has been stolen before its validity period.

The renewable flag is very useful when there are applications that need to keep the ticket valid for long period of time. However, there is a contraindication since tickets of excessive duration are at greater risk because if an attacker manages to steal the ticket, they remain valid at least until expiry. In any case, if short-duration tickets are used, the client should access the Kerberos server and its shared key on a frequent basis, running greater risks than the theft of a single ticket. In this sense, renewable tickets are very useful. They have two deadlines: the first, which is long term, when the current instance of the ticket expires while the second represents the last chance for renewal of the ticket. Operationally, a client periodically presents its renewable ticket to the trusted server before the copy of the ticket expires: when the trusted server issues the ticket, it must also activate the RENEW option otherwise the ticket will not be renewed. In this way, the trusted server produces a new ticket with a new session key, and a new expiry date and the renewal process does not alter the other fields. Once the last expiry permitted has been reached, the ticket expires definitively. After each renewal, the trusted server can consult a particular list to ascertain whether a complaint of theft of the tickets after the last renewal has been submitted. In the event of theft, the trusted server does not renew the tickets concerned. Usually, the ticket RENEW flag is taken into consideration by only trusted servers and not by application servers. It is, however, possible to ascertain the case of a particularly rigid application server that does

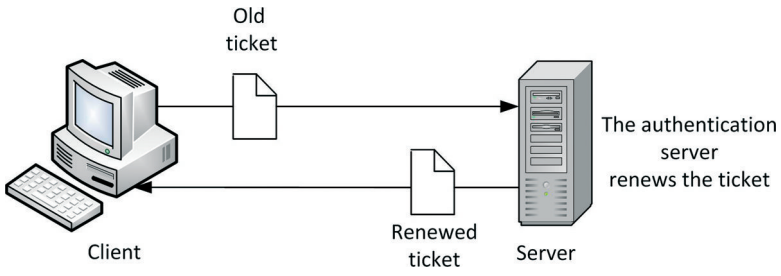


Figure 5.55 Ticket renewal process.

not allow the use of renewable tickets. If a client has failed to renew a ticket prior to its expiry, the trusted server will not proceed with renewal of the ticket. Figure 5.55 shows the ticket renewal process.

The post-dated flag is very useful when applications must obtain tickets that they will be able to use later, after a significant period of time. The operating mode of post-dated tickets is shown in Figure 5.56.

The proxiable and proxy flag is very useful when the client and the server must ensure that a service performs a given operation effectively replacing the client. In this sense, the service must be able to assume the identity of the client exclusively for the requested operation. A client or server may allow a particular service to take its identity for a certain purpose by assigning a proxy to the service itself.

Usually, the trusted server is the only one that can interpret the PROXIABLE flag of a ticket while the application servers can ignore it. When this flag is active, it communicates to the trusted server that it can produce a new ticket with a different network address and based on the current ticket. In this sense, the Kerberos server activates the PROXIABLE flag within the tickets for the request of the ticket, as the default. The PROXIABLE flag allows a client to pass a proxy to a server in such a manner that it performs a remote request in its place. To make it difficult to use stolen tickets, Kerberos tickets are only valid if they come from network addresses that are listed on the ticket itself. Since the original owner that performed the proxy does not contain a local address, a client wishing to assign a proxy to a remote service must request a new ticket that is valid for the service network address from which the trusted server has received the proxy. In this case, the trusted server activates the PROXY flag of the ticket. When the trusted server emits a proxy tickets, it activates the PROXY flag of the ticket. The

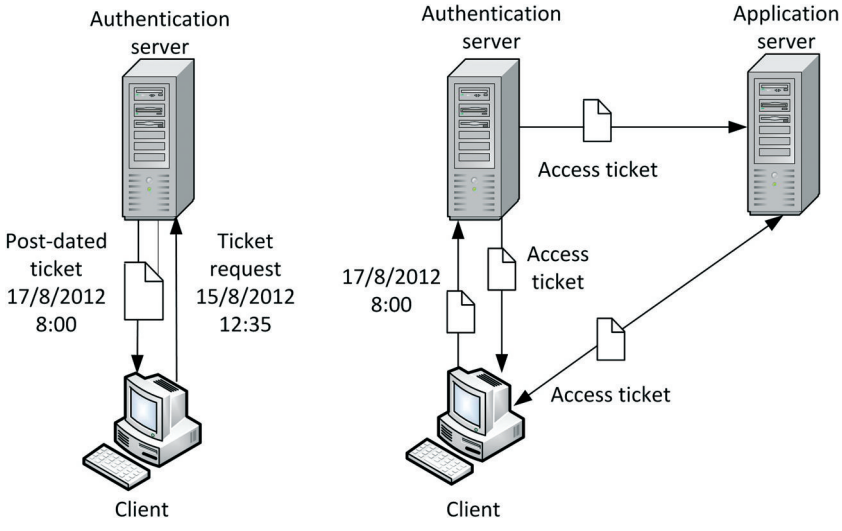


Figure 5.56 Means of managing post-dated tickets.

application server may check this flag and request further authentication by the subject that submitted the proxy in such a way as to provide an audit trail (which will be described later).

The FORWARDABLE flag represents an instance of a Kerberos ticket where Kerberos itself provides the full use of the identity of the client to the service. Forwarding of the authentication usually takes place when a user accesses the remote system and wants the authentication to act as if the user were connected locally. Usually, only the trusted server reads the FORWARDABLE flag of a ticket and consequently operates while application servers can ignore it. The FORWARDABLE flag has an interpretation that is similar to that of the PROXIABLE flag with the difference being that the trusted server may also issue a ticket for the ticket request with different network addresses. Tickets are not FORWARDABLE by default. In any case, users can request that the Kerberos server activates this flag using the relevant option. Figure 5.57 shows how a pair of Kerberos servers can process a FORWARDABLE ticket on different *realms*.

The forwardable flag allows forwarding of authentication without requiring the user to indicate his/her password. If the server has not activated this flag in the ticket request, this means that it does not want to allow forwarding of the authentication. The trusted server activates the FORWARDED flag every time that a client presents a ticket with the FORWARDABLE flag enabled and requests the trusted server to activate the FORWARDED flag by specifying the FORWARDED option of the server and providing a group of addresses for the new ticket. The FORWARDED flag is also activated in all the tickets that a Kerberos server issues starting with tickets with the FORWARDED flag enabled. The application server can process FORWARDED tickets differently from non-FORWARDED tickets.

5.10.2 Kerberos archive

The Kerberos server, in order to perform authentication, must be able to access an archive that contains the identifiers and secret keys of servers and of the client that the server itself must authenticate. Implementation of the Kerberos server does not have to store the archive on the same computer.

When the key of an application server must be changed and this change is not due to breach of the old key, the server needs to maintain the old key until all the tickets that use this key have expired. Since the server also uses the old keys, a particular server or client can have several active keys.

Kerberos always marks the ciphertext that it encrypts in the key of the server or of the client with the version of the key that was originally used by Kerberos to encrypt the key, in such a way as to help the recipient find the correct decryption key.

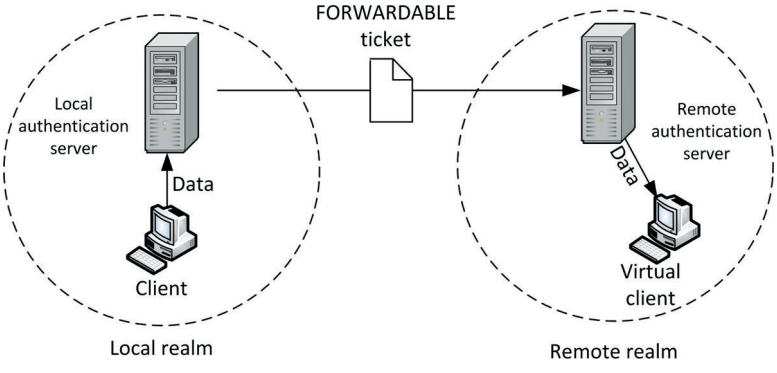


Figure 5.57 Schematic diagram of how a pair of Kerberos servers that are able to process a FORWARDABLE ticket on different *realms*.

When a server or a client has more than one active key, it will have several records thereof in the Kerberos archive. The keys and their version numbers vary from one record to another, and each time that a Kerberos server issues a ticket or responds to a request for authentication, the server itself uses the most recent encryption key and this key will be characterised by a larger version number.

5.10.3 Vulnerability of Kerberos

Kerberos assigns its authentication based on the login of the user by means of a shared key. This procedure is, however, subject to vulnerability as an attacker can always perform *spoofing* of the trusted server having the latter believe it is a legitimate user: once the attacker has succeeded in its intent, he/she has acquired the rights to full access.

To avoid this, the Kerberos system introduced *timestamp* that ensures that all messages that are received from the trusted server contain a date and that the server deletes all the messages that are more than 5 min old, complicating the task of a possible attacker. The operating pattern for the elimination of expired tickets is shown in Figure 5.58.

In any case, Kerberos remains vulnerable in offline decryption: an attacker can intercept a response (which contains the encrypted ticket with the recipient's public key) that the trusted server has sent to a server or to a client, and the same attacker can give rise to a dictionary attack on the response, attempting to decrypt the message by guessing the value of the private key. An attacker could also try to read the transmissions and use time attacks to look for the private key: if, after such an attack, the same obtains recognisable information, such as stamping or network address, then the attacker has succeeded in his/her intent to breach the private key.

Kerberos requires further precautions to avoid being vulnerable to attacks. These precautions are as follows:

1. Since Kerberos guarantees no solution to avoid service disruption attacks, and an intruder can, with extreme ease, prevent an application from taking part in an authentication process, it is very important that, when using Kerberos, administrators and users pay attention to the detection of this type of attack that can appear as a normal system malfunction.
2. As the Kerberos server assumes that the shared secret key is really secret, it is very important that the server and the client ensure the confidentiality of their own key. If an attacker manages to find a key, he/she can replace the legitimate user without any great difficulty.
3. Since Kerberos is defenceless against attacks that attempt to guess passwords, if a user uses a password that is easy to breach, an attacker can attempt a dictionary attack to try to decipher the

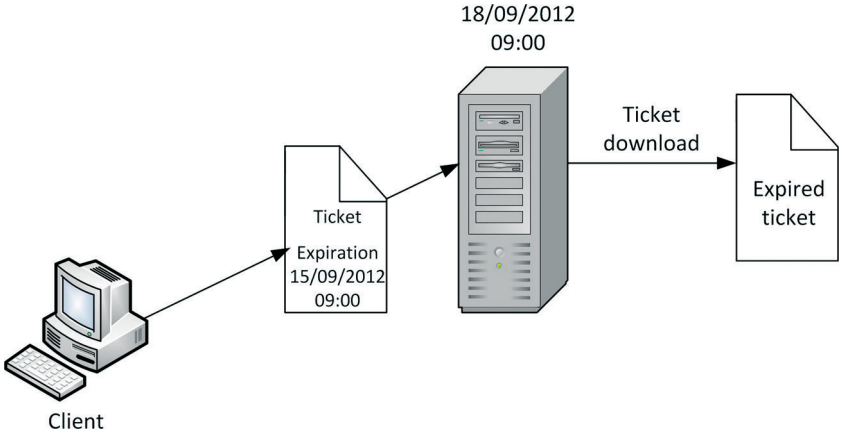


Figure 5.58 Diagram for the removal of expired tickets.

messages that the attacker itself manages to copy. Because the user encrypts messages using a key that Kerberos generates from the password, an attacker may be able to find a user's password after a certain number of attempts.

5.11 Security of commercial transactions on the Internet

Commercial transactions on the Internet are, currently, a significant part of the overall commercial transactions. The problem is that, in actual fact, the Internet was not designed to safely perform these types of transactions that normally take place on it and for this reason, it is necessary to take appropriate precautions. From the point of view of a single user or an organisation, it is essential to be certain that potential attackers are not able to interrupt, modify or copy transactions that take place on the Internet. Currently, if our transmissions on a network are not suitably protected, transactions that occur on it can be easily attacked, as shown in Figure 5.59.

As the data in the transmission are transmitted in the form of plaintext, the attacker need not decipher the transmission to see the data it contains and if such information is composed of credit card numbers or other confidential information, the attacker has easy access to such information that it can use fraudulently at a later date. For this reason, it is absolutely necessary that in the case of e-commerce, use is made of strong cryptography characterised by inviolable or otherwise difficult to attack keys. There are, in this sense, secure protocols for the transmission of commercial information on the Internet such as the already illustrated S-HTTP and SSL.

Another significant problem of commercial transactions on the Internet is authentication. In fact, when we use a credit card in a normal shop, an attentive shopkeeper always asks for an identity document to authenticate the card holder, but this does not happen on the Internet. In this sense, having the digital signature that can be used to verify and authenticate a transmission appears to be very helpful, as shown in Figure 5.60.

The problem of digital signatures is that our digital certificate is located on our computer. If an attacker manages to get access to this computer, he/she can steal the certificate and assume the identity of the respective owner. For this reason, it is particularly recommended, if we use a laptop computer containing a digital certificate, to use protection passwords at least at the Basic Input-Output System (BIOS) level, even if an expert attacker manages to evade even these protected passwords. In any case,

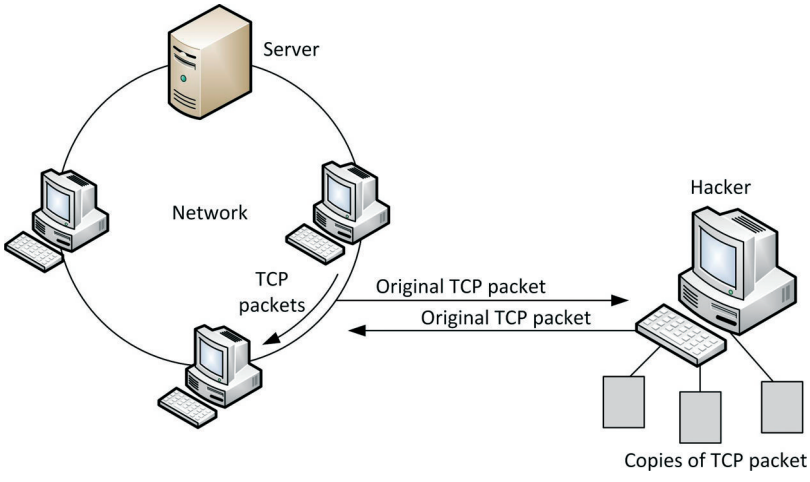


Figure 5.59 An attacker can, with extreme ease, copy the transmissions that occur on the Internet if the latter are not suitably protected.

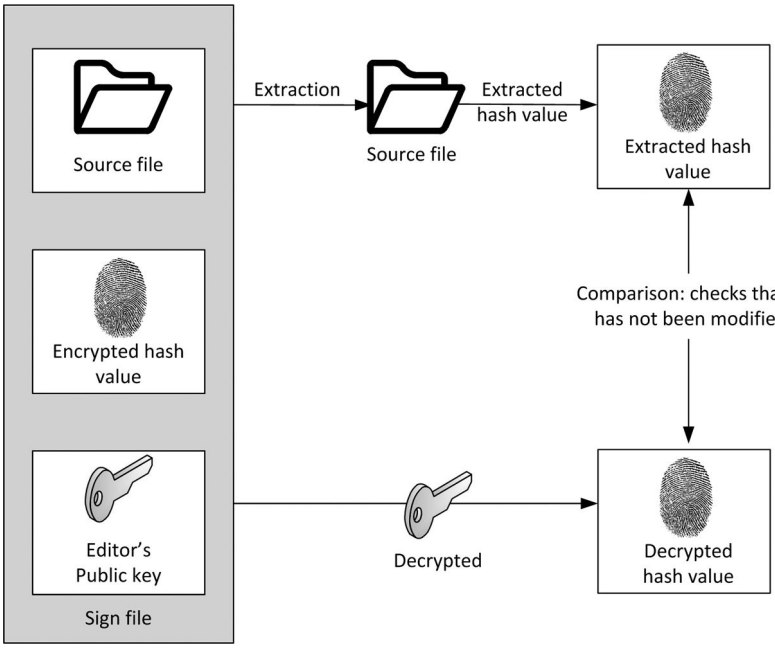


Figure 5.60 Schema of digital signature that allows the recipient to verify and authenticate transmissions.

there are different protocols that allow us to encrypt and sign transmissions, guaranteeing a high level of protection from network attacks.

Another problem is the confidentiality of transactions. In fact, when we are making a purchase in a shop, there is no recording of personal identity. On the other hand, when we make a purchase with a credit card, the ship records the identity of the purchaser, the type of purchased goods, etc. These records are also available to the company that owns the credit card, allowing the same to analyse the buying habits and to deduce other confidential information. Since on the Internet purchasing takes place mainly via the use of credit card or with bank payments, each operation leaves a trace, thus reducing the level of confidentiality.

SSL and S-HTTP address the security of transactions over the Internet and will not be discussed further as they have already been illustrated. This section discusses aspects of the protection of confidentiality and, in particular, a new technology called digital cash. Digital cash may take various forms, from smart cards (a type of credit card containing a computer chip) to electronic certificates that can be issued by banks or other owners.

It has already been said that the main drawback of the forms of electronic payment used currently is that they leave a trace that inevitably reduces the level of confidentiality. These traces are of relevance to many subjects: the forces of law and order, the companies that carry out market research, etc.

The great advantage of digital cash is its greater discretion with respect to credit cards, cheques and bank transfers, as they can offer the same level of confidentiality as cash. It allows consumers to buy what they want without leaving significant traces, offering the same possibilities of cash. However, there is a contraindication that the lack of recording can facilitate money laundering via the Internet, feeding the activities of organised crime.

Electronic cash bases its operation on digital signature cryptography algorithms already illustrated. The algorithm used to a greater extent consists of a pair of numeric keys that operate as two halves of a secret code and the messages that are encrypted with a key can be decrypted only with the other key. One key is published while the other key must be kept secret. In practice, a public/private key algorithm is used. Because of this algorithm, if all users have a public key, a bank may allow them to

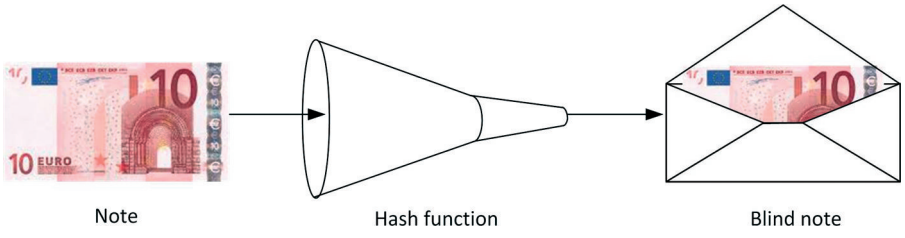


Figure 5.61 Calculation of the hash function by the user before transmission to the bank.

transmit messages of any kind, including those related to financial transactions, by encrypting them appropriately through their public key. Public/private key cryptography allows users to sign the message, allowing the bank to decipher the message with the certainty of obtaining valid and authenticated content.

In electronic cash systems, the user's computer generates a random number that is used as a banknote. Then, the computer hides the banknote using a random number and sends it to the bank. In practice, the user applies a hash function to the banknote before transmitting it to the bank, as shown in Figure 5.61.

Subsequently, the bank debits the user's current account or account that is indicated by the user and uses its own private key to sign the hidden banknote and transmits the banknote to the user, as shown in Figure 5.62.

Once the user has received the note signed by the bank, the computer or the user's device reveals the banknote, passing it in hash function, obtaining the banknote signed by the bank, as shown in Figure 5.63.

At this point, the user can use the note signed by the bank for the desired purchases. When the user transmits the banknote to the seller, the latter verifies the digital signature of the banknote that allows it to be certain that the bank has signed the note, as shown in Figure 5.64.

The seller then transmits the banknote to their bank to deposit it in their own account. The bank, in turn, verifies the signature of the debiting bank and credits the amount to the current account of the seller.

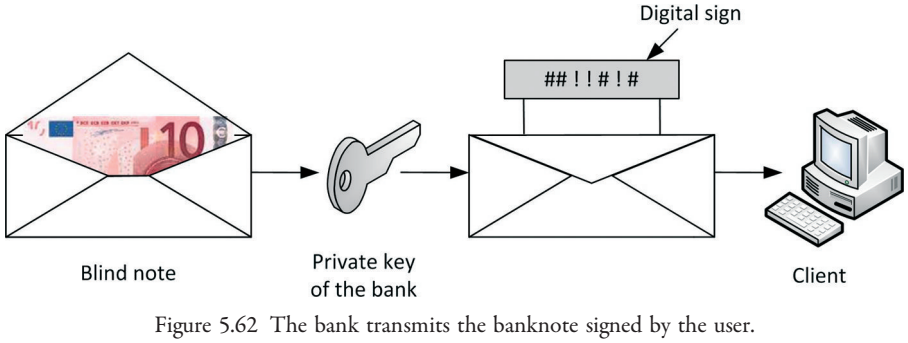


Figure 5.62 The bank transmits the banknote signed by the user.

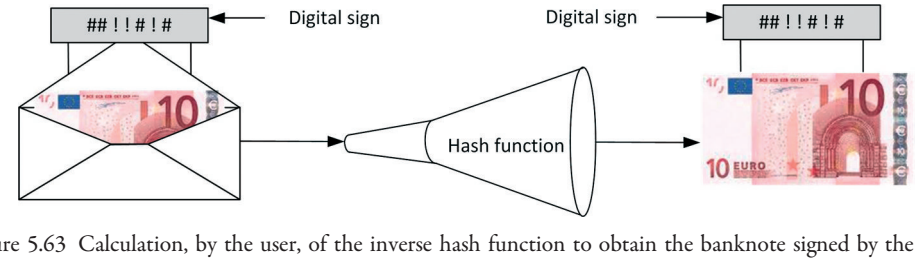


Figure 5.63 Calculation, by the user, of the inverse hash function to obtain the banknote signed by the bank.

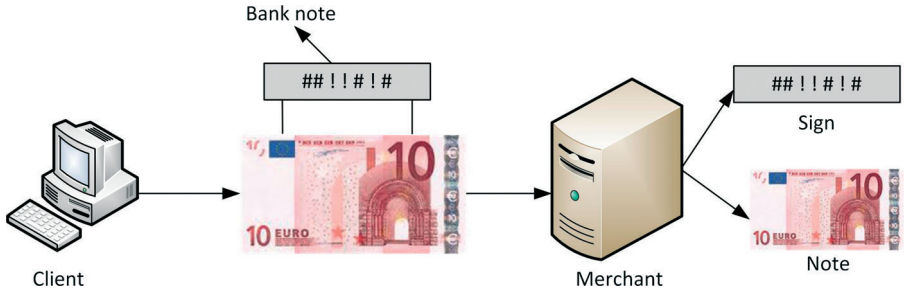


Figure 5.64 Verification by the vendor of the validity of the banknote.

Digital cash not only ensures confidentiality of the buyer but can also ensure the safety of all those concerned in the transaction. In fact, neither the buyer nor the seller can falsify the signature of the bank because neither of the two subjects have the bank's private key. They may, however, check that the payment is valid, as they both know the public key of the bank. The user can also demonstrate that payment has been made and can make the blind code available. As the user hid the number of the original banknote when he/she sent the same to the bank, the bank is not able to relate the signature to the payment. The combination of blind code, signature and encryption allows banks to protect digital cash from attempts at counterfeiting, sellers from being refused by banks when paying with legitimate banknotes and the user against unfounded accusations and against the invasion of their privacy.

A problem that may arise is multiple use of the same digital banknote. In fact, when buying goods from a real seller, what is being done is delivering a number of paper banknotes according to the selling price: the physical passage of the banknotes does not allow the buyer to use them several times. In the virtual case, the buyer might be tempted to use the note several times with different sellers. To avoid this, many systems use a response system, in which the seller's computer issues a question that is not foreseeable to which the buyer's computer must reply, providing information regarding the number of the relevant banknote and allowing the seller's computer to accept payment via digital cash. The information provided by the computer of the buyer does not provide the computer with any personal information on the identity of the buyer. In any case, if the buyer tries to use a certain banknote a second time, the information provided by his/her computer will reveal that the banknote has already been used. On this occasion, the bank can reconstruct the identity of the buyer and accuse him/her of the illicit use of a digital banknote. This situation is represented schematically in Figure 5.65.

Unlike traditional money that is printed by the mint of a country, digital cash is not absolutely bound to a particular national currency and this could cause political problems. In addition, there is the significant problem of the breaking of banknotes (a large payment can be made with a large banknote or with several small denomination banknotes) and the change, which requires a transmission operation, from the seller to the customer, of a certain number of small notes. To avoid this, the concept of electronic cheques can also be used, represented by a single number that contains sufficient

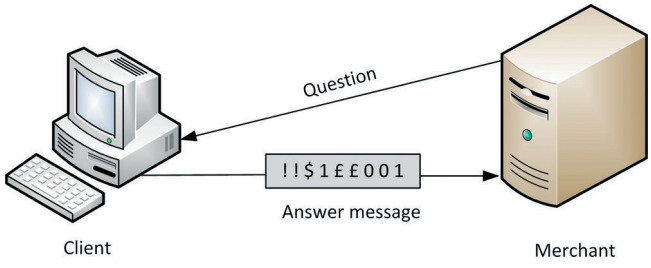


Figure 5.65 Schematic diagram of the question and response system to obtain from the buyer additional information on the digital banknote that is to be used.

times to perform a transaction valid within the limit of the banknote, and to which the program of management of the digital cash assigns the value required at the time of payment. The buyer's computer has a certain number of cheques that relate to a certain bank digital banknote: if a cheque exceeds the value of a certain banknote, the buyer must return to the bank (virtually) and purchase further digital cash. Furthermore, the cheque is characterised by having the digital signature of the bank, in a manner similar to the banknote, and by a system of double checking that prevents multiple uses.

When a user buys a digital banknote from a bank, the same credits the user the corresponding sum. In this manner, the bank has a trace of the sum covered as well as other information on the banknote. As a result, when the user uses the banknote to make a purchase, the seller delivers the banknote to their bank that may transfer the corresponding economic value deposited in the issuing bank. It is clear that it is possible to follow the path of the banknote to understand what purchases were made with it. To ensure the right to privacy, many banks guarantee anonymous banknotes. An anonymous banknote is similar to a cash order and when a seller deposits this order, the relevant bank performs the payment in cash. In this way, the bank is not able to trace back to the subject that has used the banknote unless via comparison between the figures of the purchases, an operation outside of the scope of a bank, but within the scope of the law forces.

If a user wants a bank to sign one of its banknotes without providing further details about the type of purchase that the user intends to make, the same user can use an anonymous banknote (*blind note*), which may be obtained by selecting a number of banknotes that his/her program will multiply by a random number. At this point, the user performs encryption of the value obtained using the public key of the bank and sends the encrypted banknote to the bank. This situation is shown in Figure 5.66.

Once the bank has received the encrypted and blind banknote, it decrypts it using its private key and signs it using a one-way hash function, suitably encrypting the result by using its own private key. Later, the bank adds the result to the banknote and sends it to the applicant. Depending on the applicant and the bank, the latter can send the banknote either in plaintext and signed or in plaintext and signed but encrypted. Figure 5.67 shows how the bank signs and encrypts the signed banknote using the public key of the user.

If the banknote is encrypted, the applicant deciphers the banknote using its own private key, obtaining as a result the signed and blind message. At this point, the program of the user computes the original number of the banknote by simply dividing by the random number chosen to make the banknote blind. After this transaction, the user will have their banknote signed by the bank, as shown in Figure 5.68.

As the blind factor is represented by a random number, the bank is not able to determine it and in this way may not trace back to the signing of the subsequent payments.

5.11.1 Use of credit cards on the Internet

Transactions that use normal credit cards obviously require three components: the buyer, the seller and the company issuing the credit card. Transactions on the Internet are substantially similar with the

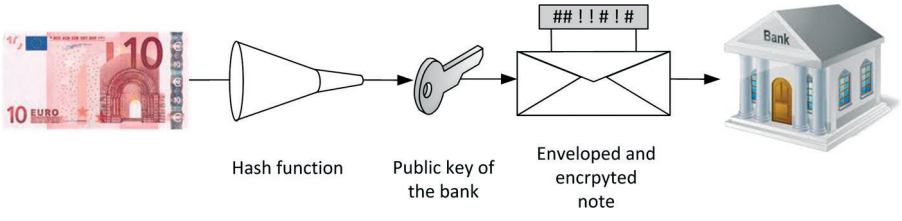


Figure 5.66 Diagram of anonymisation of the banknote, encryption and sending of the same to the bank.

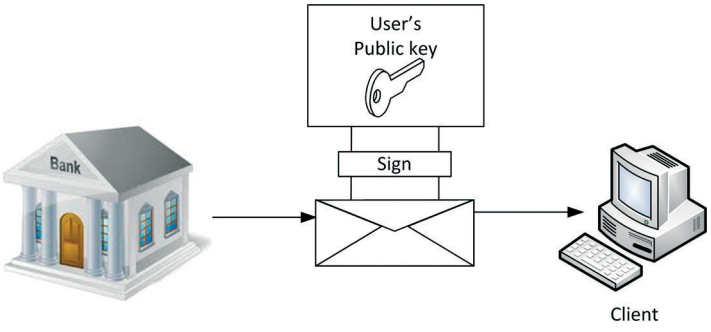


Figure 5.67 Scheme of sending to the user, by the bank, of a banknote signed and encrypted with the public key of the user.

only difference being that our credit card number is provided to a seller that may be at great distances from the buyer. When the data relating to our credit card are transmitted via the Internet, we are in any case exposed to risks. In this sense, besides being very sure of the identity of the seller, attention should be paid to whether the Web browser displays all the security indicators and that the connection used for transmission of the credit card data is secure.

5.11.2 The Secure Electronic Transmission protocol

It is clear that it is in the interest of all lawful parties involved in a purchase via the Internet to ensure that the same is performed in a manner that is secure for all. For this reason, various protocols were created including Secure Electronic Transmission (SET), developed by a number of credit card companies, which allow us to enjoy secure transactions using encrypted connections between the buyer, the seller and a server for processing of the payment.

When SET is used, the buyer, the seller and the server for processing the payment must hold a digital certificate and public/private encryption keys. To perform a secure transaction, SET carries out its activities by following steps:

1. The buyer makes a purchase on the Web.
2. The seller sends the user a message that contains a certificate, a unique code for the transaction, a certificate for the exchange of the key for the server for processing the payment and, finally, a certificate of key exchange for the seller.
3. The program of the buyer, using the key exchange certificate of the server for processing the payment, performs encryption of a message that contains its certificate and its information on the purchase. This message is sent to the payment processing server.
4. The program of the buyer, using the seller's key exchange certificate, performs encryption of a message that contains its own certificate and information on the purchase. This message is sent to the seller.
5. The seller's program performs encryption of the message that contains the information on the

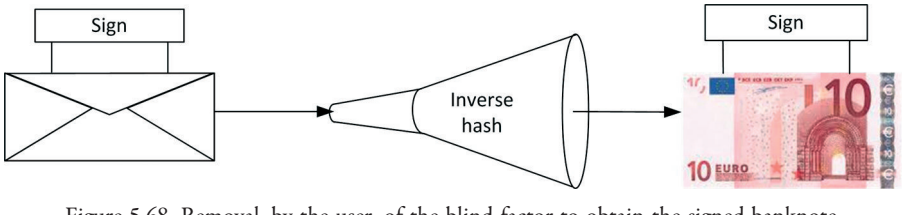


Figure 5.68 Removal, by the user, of the blind factor to obtain the signed banknote.

purchase and the certificate of the buyer and sends it to the payment processing server.
6. The payment processing server, once the messages of the buyer and seller have been received, authorises the operation, covering all the parties concerned.

To be sure that we are in a secure connection, it has already been stated that special attention must be paid to certain graphic elements that are shown in the browser window. Most of the navigators, when they are within a secure connection show a padlock in the lower right-hand corner. But it is important to avoid coming under attack, as already illustrated previously, situation for which these elements can be generated by special programs.

Once we are connected to a secure website, we must also be certain that it is the correct site. As has already been shown previously, a possible attacker can generate a false site that is equal in all respects to the original site except for the fact that it is designed to ensnare the end user. To avoid this, we should always check the digital certificate of the site to which we are connected. If the site uses SSL, the latter requires that every website that uses it is equipped with one or several digital certificates to verify the identity of the organisation that owns the website. To view this certificate, a double selection on the padlock icon that appears in the browser window at the bottom on the right must be performed. The digital certificate must indicate the name of the organisation, the place of the activity, the authority that issued the certificate, and, finally, its expiry date. If one or more of the pieces of information mentioned appears to be missing or if the certificate has expired, we should be wary of the use of this site and of the sending of our personal data.

5.12 Audit trails

Audit trails are one of the best ways to detect the infiltration of an external attacker. An audit trail represents an almost permanent recording that is stored in the operating system of a computer to have a trace of tasks that users perform on a computer. The audit trail is of vital importance not only after an attack, in order to reconstruct the dynamics and to inspect the unlawful activities performed, but also during the same attack.

The audit trail is an essential network administration tool. The majority of network administrators use the audit trail to identify internal and external attacks, to monitor the activities of employees and to control access to a server. In the latter case, audit trails are capable of controlling access to the server as all the times that the server issues a request to identify the user or permission to access it, the computer that is trying to access the server issues a response. This response also includes the IP address. The response information allows network administrators to check the identity of the subject that has attempted access to the server, the way in which the same has obtained access and the activities performed during access. In practice, the audit trail allows detection of an attacker that has gained access to a system. It also allows the blocking of an attacker in his/her attempts to compromise a system, giving the administrator more information relating to the attack. Audit trails, moreover, allow internal users to be monitored to be certain that the latter do not perform activities that could compromise the system. In practice, according to the definition of the National Computer Security Centre (NCSC) of the United States, an audit trail represents a historical record of system activities, sufficient to permit the reconstruction, the revision and examination of the sequence of situations and activities relating to or that have resulted in a transaction, a procedure or an event in a transaction from its beginning to its final outcomes.

Figure 5.69 shows the way in which an audit trail records attempts to access the system.

The vast majority of servers are equipped with an auditing packet as a basic provision in terms of equipment. In most cases, this package is activated during initial installation and if we want to uninstall it, this requirement must be expressed explicitly. Systems that use audit trails are usually more difficult to attack because every time that an attacker attempts to perform an authorised task, the system

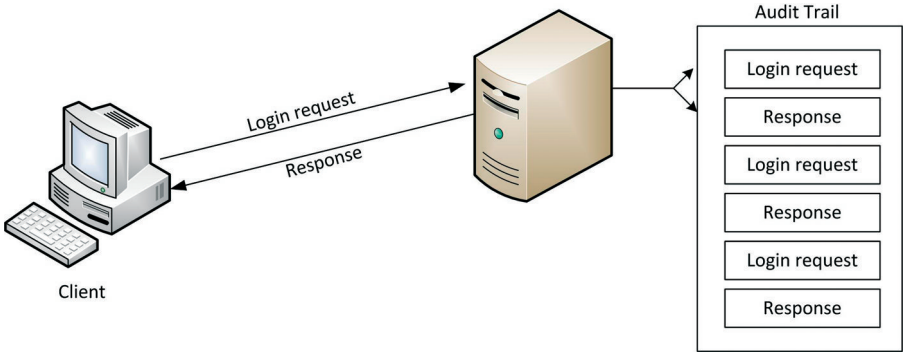


Figure 5.69 Control mode of access to the system by audit trails.

performs the corresponding recording. To know how it is possible to activate (if not already active) the audit trail on our computer, our operating system manual must be consulted as this mode varies from operating system to operating system. Verification of the activities performed is a significant task and must often be performed by several people, given the high quantity of data to be controlled even in a system characterised by a small number of users.

It should however be pointed out that the auditing, maintenance and control of audit trails do not guarantee protection against any attack on the system. In fact, if an attacker manages to perform *spoofing*, its activities will not be highlighted by auditing as they are part of the normal activities that the user being attacked performs, unless the latter is performing activities that are not permitted. In addition, if an attacker performs *sniffing* activities, these activities may not be recognised by auditing as the attacker performs no access to the system but is limited to observing transactions on the network.

If the attacker performs an active attack, he/she will almost certainly provoke a TCP ACK storm, generating a high and sudden number of packets and such activity cannot go unnoticed even to the less expert administrator.

In practice, audit trails represent an additional instrument to the other tools seen so far to locate an attack and to protect the network in the context of an organic security plan and that uses several resources. In this sense, an audit can in no way represent a substitute for a firewall, a screening router or a correct security policy.

As it is not possible, for reasons of space, to address audit trails on the various existing operating systems, the following will explain only the general concepts, valid for most operating systems.

The Internet makes it possible for audit trails to show in a particular manner requests for access to restricted objects and resources. Audit trails can be used on firewalls and screening routers to detect when an attack is under way. Depending on the information on a server, it is possible to use different levels of auditing such as computer auditing, characterised by reduced accuracy, directory auditing, characterised by average accuracy and object auditing, characterised by high accuracy. Audit trails can be used after an attack to quantify the damage that an attacker has produced against a system or data that have been altered or stolen. A correct corporate security policy must provide for periodic maintenance, analysis and copy of audit trails. Internal users always represent a source of potential danger even for the more secure systems and in this sense, audit trails are the only way to discover any unlawful activity being performed by internal users. Audit trails are an irreplaceable means of understanding the dynamics of an attack and the methods of operation to perform the same.

5.13 Java language and related security aspects

It has already been seen that to generate versatile and interactive Web applications, more powerful tools with respect to only HTML must be used. In this sense, there are other languages such as, for example, Java, which is used to produce Web applications called applets. Given that every time we download a program from the Web we run, the risk of infecting our computer with viruses (which will be illustrated in detail in the following), Java applets, to avoid this, cannot read or write any type of information from the computer's disk that uses them. This does not allow Java applets to introduce computer viruses onto the computer that is currently executing them, nor does it read from the disk to be sent to a remote attacker. Java is a programming language that is extremely powerful and feature-rich, capable of generating programs that read and write files and access system services. It is mostly used to write short programs, which as already stated are called applets, which are capable of performing relatively limited functions and that are performed in most cases in a Web browser. Java applications, in contrast to applets, are instead complete programs that are able to perform complete functions that can operate outside of a Web browser and are able, as has been said, to access the disk for read and write operations, being able to operate both on a server and on a client.

Java is provided with appropriate security mechanisms that prevent dangerous applets that are designed to attack a computer from damaging the same. Java applets can always generate DoS attacks that are possible when browsing a website that requires the use of applets.

Programmers use prudence in compiling most programs in order to make them faster to use. To do this, they use an appropriate program called compiler. The compiler translates into written program through instructions written in machine code, represented by binary machine instructions that can be understood directly by the microprocessor of the computer. Unfortunately, every type of computer uses a different machine code and a different operating system, for which reason programs compiled for a given operating system do not run properly on other operating systems. As such, programmers must compile the same program with different compilers to make it run on different computers.

Java was initially developed at the beginning of the 1990s by Sun Microsystems to create a software for smart devices, represented, for example, by household appliances. Its original name was Ohio. It was felt that this area would not be sufficiently profitable and the project was abandoned but the various members of the working group created a presentation using personal computers and this was a field that was not explored with this language. The demonstration consisted of applets that were downloaded from the Web on various platforms and that were run within the Web browser. At that point, Sun understood the potential of the new language and changed the name of Ohio to that of Java. Java is very similar to C++ but is characterised by certain differences.

A first difference is the use of pointers that represent a means for direct access to the memory addresses of the computer. When Sun decided to develop Java in order to be executable on any computer and given that every computer accesses memory differently, the final decision was to not implement pointers. The lack of pointers is an extremely important element regarding the security of Java.

A second difference is the internal management of *threads,* entrusted to a particular system in Java. In essence, a *thread* represents an interaction of the program with the operating system. Thread management allows the program to control when and how a Java program is run.

A third difference is that the Java compiler compiles programs in a structure called *bytecode*. Once a program has been completed, the compiler places the *bytecode* in a class. A suitable element, called *Java Virtual Machine*, which is contained in the Web browser on the local computer, translates the *bytecode* into machine code that can be executed by the computer. C++ programs are, however, compiled directly into machine code and can only be run on that computer.

A fourth difference is that Java programs are executed on computers only after the *Java Virtual Machine* has translated programs into machine code. The *Java Virtual Machine* is located above the

operating system and generally within the Web browser. Once the Web browser finds the <APPLET> tag in an HTML file, it removes the file of the class relating to the applet specified by the server. Subsequently, the *Java Virtual Machine* performs a series of checks on the applet to be certain that the latter does not compromise the security of Java and of the computer itself. Once the *Java Virtual Machine* has performed relevant checks, it generates the applet into machine code of the computer's processor that, in practice, controls the computer that is running the applet.

Even if Java is a programming environment characterised by a good level of security, it is important to consider various points that allow us to protect ourself from various security issues that may result from the use of Java. As *Java Virtual Machine* interprets applets locally, the applet usually uses considerable amounts of system resources. Applets programmed badly or even hostile ones are able to consume a high amount of processing and memory resources and may eventually even lock the computer that is currently executing them: in this case, the only way to get it to restart consists of turning off and restarting. The high consumption of system resources is usually used for a DoS attack.

Java applets use a security system also known by the name of sandbox that is very useful to protect our computer from the intrusion of hostile applets. The sandbox restricts access to the system by the applet, limiting its effectiveness in well-defined areas as an applet that operates in a sandbox is characterised by limited access to system resources, since it is not possible, for example, to access the hard disk, to open new channels of transmission, to return detailed information about the client running the applet, etc. Standard applets and Java standard library are sandbox applets. Sandbox applets are known to be secure unless the applet finds a way to exit the sandbox.

There is a variant of the Java model, called *trusted*, whose applet has access to all the system resources and is able to work outside of the sandbox. These applets are created by a trusted organisation within a corporate intranet or are signed by the author prior to transmission over the Internet. Since this type of applet is characterised by unrestricted access to the system, it is generally not possible to guarantee security.

Java is characterised by a series of very powerful functions. In the following, the functionalities that represent the critical issues with regard to security are illustrated, referring the reader to specialist texts for more in-depth study on such programming language.

Applets that are downloaded from the Internet or from a remote address are loaded into the browser on the local computer and are not able to read or write file types that are present on the client's file system. They also cannot open other connections and can only communicate with the host from which they come. Applets that are downloaded from the Internet cannot start other programs on the client, may not load libraries and cannot define native method calls as the latter access the computer's operating system, providing full access to the computer, and lead to a loss of security features of the sandbox in which the applet is running.

It has already been said that the standard applet cannot read or write on files that are present on the local system. In particular, the *Java Virtual Machine*, once the program has been downloaded from a remote site and runs on the Web browser of the end user's computer, does not allow applets to execute the following steps: reading of a local file, writing of a local file, renaming of a local file, checking for the existence of a file on the local computer, creation of a directory in the file system of the local computer, listing of the contents of the local files of the system, checking of the type of a local file, checking the date and time of a file in the file system of the local computer, checking of the size of a file.

The main goal of network administrators should be represented by the fact that the executable files should not be able to open a transmission channel of the source computer or of another unknown computer. The opening of a transmission channel is particularly dangerous since the applet can be run without the slightest interaction with the user of the computer, and even without the user being aware of it. If an applet is not subject to security checks, once downloaded from the network, it would produce no graphic effect on the screen in order to avoid warning of its presence and may collect information that relates to the computer itself, the network, the password file and other types of

confidential information, and transmit such information to the remote host from which it originates or even to another host, using the available communication channel.

In order to avoid the hostile applet being able to send confidential information, the security procedures of Java do not allow applets to open network connections with other computers on the Internet or on the local network with the exception of the host computer that has provided the class files. The host computer is a machine that contains the HTML page that starts the Java program. Some Java applets that have access to the archive of a server can typically communicate with the source server to receive information directly on the archive itself. Java security procedures allow applets that run on the local computer to connect with the one server from which they have been downloaded. If an applet tries to open a connection other than the one from which it comes, it would generate a security error and would be blocked, causing a possible warning message in respect of the user of the program itself.

The only network connection that an applet can open is that which allows connection with the source host. When a Java program tries to open a connection with the source host, the same program must specify the host in the same way that the browser called the source host when the applet was extracted. The applet neither connect to the destination host using a 4 byte IP address nor can it use a shortened form of the host name. In some situations, the firewall may not allow any type of transmission in output.

In relation to programs, in many cases reference is also made to persistent objects. Persistent objects are objects that the computer can save permanently or that in any case it saves in a certain place. In this case, a persistent applet is an applet that the operating system also saves permanently on the hard disk or on the network. A persistent object is an object that provides a reminder of everything that has happened before, without the need to open, save or close the relevant file. Every time access is made to a persistent object, it continues its implementation from the point where the same was stopped. The main difference between objects and persistent programs is the transparency of the storage. A persistent object handles all the file operations independently and without requiring any user intervention. As persistent objects need to access the disk and files, they do not offer maximum security from the Java point of view. In practice, the persistent applet can be used several times while applets that are not persistent are inevitably erased when they are downloaded by a computer's memory. In any case, applets retain the status of persistence on the server where they are stored. Applets can also create persistent files on the server and read the same files from the server.

The sandbox security model does not allow applets downloaded via the internet from the Web browser to start other programs on the client: in practice, a process cannot start one of its replicas of a process on the computer on which it is running. This model also avoids the launching of programs external to the sandbox, in other words programs that could provide the applet with data that the same applet may use to send it to the original server. Without the sandbox, an applet would be able to read the system directory and direct the output towards itself to then send the content to the original server. Applets, being unable to replicate or in any case not being able to control files outside of the browser may not activate the closure procedures of the system. Applets may not, in addition, access other programs running in memory.

It has already been stated that applets, unlike programs written in C/C++, may not resort to the use of pointers. In this sense, programs can access Java objects by requesting a reference (*handle*) for the object from the *Java Virtual Machine*, a substitute of the objects pointer representing the reference. Java programs are not able to manipulate references using pointer arithmetic. In any case, Java applets may not change the references.

With programs written in C/C++, it is possible to perform various operations on pointers in order to implement strings and arrays and their operations. In Java, there are high-level commands that allow the operation on strings and arrays without having to use pointers. The *Java Virtual Machine* accurately controls the limits for execution of strings and arrays to prevent wrong, intentional or unintentional operations resulting in hazardous operations for the security of the computer that is running the applet. If an error occurs, the program is blocked, providing an appropriate error message.

Once we create a string or array object is created, the program can no longer modify the length of the same: in practice, it is not possible to perform operations of dynamic memory allocation for security reasons.

Java strings may vary only within the program but may not change their sizes. By forcing programmers to use strings that are not variables, it is possible to avoid mistakes that can be used by attackers to make hostile applets. The problem of fixed strings is that often, to avoid remaining short of space, they become excessively oversized, occupying a large amount of memory. Deactivation of the computer's operating system, due to the loading of strings characterised by exaggerated size, represents a possible type of attack. However, there are other types of attacks that can be conducted despite the security precautions that exist in Java.

The compiler, in any case, checks before recalling an object method, that the object itself is correct for the method. A method represents a specific function of a certain type of object.

The latest version of Java is equipped with new application programming interfaces (APIs) of language security. APIs are a set of procedures available to the programmer, usually grouped together to form a set of specific instruments for a given task. They represent a method of obtaining an abstraction, usually, between the hardware and the programmer or between low- and high-level software. API helps programmers to avoid writing all the features from scratch. API themselves are an abstraction: the software that provides a certain API is called implementation of the API.

These APIs provide new capabilities for Java applets, including the support of encrypted transmissions and Java Archive (JAR) archive files with digital signature. Websites use JAR type files to transmit applets and the related associated files. Digital signatures and digital certificates ensure a high level of security from the point of origin and allow identification of the subject that has signed them. Unfortunately, digital signature alone does not represent a guarantee with regard to the security of the applet: it merely confirms that someone has registered the applet.

The Java security model is quite complete but unfortunately does not guarantee absolute security. The most common attack that is conducted is represented by prevention of the use of the system itself. In this case, the Java applet requires a browser to run the applet continuously resulting, in general, in a system block. However, there are also hostile programs that exploit verifier bugs to close other applet processes running in the browser. To protect our computer, it is thus necessary to have a certain familiarity with hostile Java applets. There are applets that attempt to disable the keyboard and mouse, making it very difficult to interact with the computer to block the applet itself. There are many other applets that create windows of the size of one million by one million pixels, placing them one on top of the other in an attempt to exhaust the resources of the computer and to block it. There are certain applets that attempt to add large amounts of strings to the memory buffer, again in an attempt to exhaust the resources of the computer and to block it. The two activities are often carried out together in order to increase the probability of blocking of the system.

5.14 Web browser security

The browser is a program that is greatly used for navigation on the Internet. All the major software production houses produce a browser program. They differ in functionality, performance and graphical interface but are all targeted at Internet browsing. They are very much used and also subject to attack by external hackers, aiming to compromise the security of the computer on which it is browsing. The purpose of this section is to explain all the possible threats to which we may be exposed when browsing the Internet and the countermeasures that can be taken.

The main purpose of networks designers, when the latter provide connections to the Internet or more simply in intranet connections, is the security of incoming traffic to the various computers. These

issues have already been described, in detail, in the previous sections, which addressed the impact that an insecure installation may have on the network and relevant countermeasures.

One of the problems that is often overlooked is represented by browser security and a user who is not prepared and equipped with a browser that is not secure can compromise the security of the entire network to which the same is connected.

It has already been said that Web developers generate Web pages by inserting within a text document particular commands called tags. Tags that can be used by developers are the default HTML. Developers, in a Web document, use HTML tags to define the font size, the position of the images and the various links to other documents that are on the Web. Users, in order to be able to read and interact with a Web document, must use a program called Web browser. In Internet, there are millions and millions of websites and their number is growing constantly.

Users, when surfing on the Internet, feel relatively secure but do not know that, without them being aware of it, sites visited can gather information about the users themselves and the system being used in such a way as to analyse their browsing habits. In addition, users themselves can be attacked in various ways.

When we visit a Web page, the remote site is able to gather the following information:

1. browser used, the type and version;
2. operating system;
3. CPU type;
4. type of connection;
5. settings of the security system;
6. IP address;
7. screen resolution and colour support;
8. profile information stored in the browser.

There are many types of Web browsers but the most popular, historically, are Internet Explorer and Netscape. Internet Explorer is very popular because it is provided with the Windows operating system in all versions. The other browser, very popular until a short time ago, is Netscape Navigator, which is a direct descendant of Web Mosaic. There are currently several browsers available in addition to the two browsers mentioned above, all created by different manufacturers.

Browsers, like all programs in general, even if developed with meticulousness and precision, can have bugs in the code that can be used by attackers, once known and studied, to attack the browser and the relative computer on which the same is running.

It is well known that all the times a new version of a software package is created, the version number is increased. This applies to all programs, and thus also for browsers. The new versions are made available immediately, in most cases free to all holders of older versions. As the improvements that concern browsers in general relate to security, it is recommended that the latest and updated versions are always used. The use of final versions and not the trial version, in most cases called Beta versions, is also strongly recommended. In fact, the interim versions, for which the manufacturer does not guarantee its assistance, may be characterised by certain problems including those relating to security and operation of the program itself.

A critical element that relates to the issue of security is linked to the version number as a browser characterised by a recent version number has undoubtedly addressed and solved the problems of security of a browser characterised by a minor version number. It may also happen that a new version is characterised by new problems typical of that version. It is therefore very important to visit the manufacturer's Web page from which the program was downloaded to check if there are any updates (*patches*) to be able to download and install on our computer.

5.14.1 Simple attacks on Web browsers

Web browser attacks can be very complex or very simple, depending on the vulnerability discovered by attackers in the specific browser. For example, in HTML files, an <a> tag is used to create a hyperlink that users can select to visit another page. In practice, within the HTML file, every <a> tag specifies the URL of a certain site. The <a> tag can also be used to point to programs that are on the user's hard disk. If in some way an attacker is able to download a virus program onto hard disk (which will be discussed below in more detail) with the extension.exe or.com, using the <a> tags, it is possible to activate the same program and unleash an attack against the computer. To avoid this, the most recent browsers, when the same are prompted to activate a program with the extension.exe or.com, trigger a dialogue with the user asking what they have to do with this program (whether to save it or activate it). This window is activated even if the program already resides on the user's hard disk. An inexperienced user may not understand the full significance of this window and accept implementation of the program, activating an attack against the computer that could have catastrophic results.

5.14.2 ActiveX components and associated security issues

We have seen previously that, to extend the functionality of a website, different languages have been created including Java, through which developers can create any type of secure application, thanks to applets, which can be downloaded from the Web and run. Since applets have limited access to the system, Java is a discreet secure solution for Web applications.

With an increase in the complexity of Web applications, developers needed programming languages that would gradually be more powerful and versatile. Java, on the other hand, having to use the virtual machine to translate the programs into *bytecode*, may prove to be slower than different solutions. To avoid this, Microsoft developed ActiveX technology which, unlike Java applets that are independent from the computer being used, is only supported in the Windows/Intel environment, thus avoiding having to perform the translation operation because the code is already written in machine language.

In a way similar to Java, ActiveX has had considerable success on the Internet. ActiveX programs, as opposed to those written in Java, reside directly on the computer's hard disk, and the Web browser, when encountering one of these controls in the pages, directly downloads the program onto the hard disk. This mode might lead to problems of security but is characterised by a number of advantages including:

1. reusability, because other Web pages can access the ActiveX control previously downloaded without requiring the browser to download them again, greatly speeding up surfing time. Unlike Java applets that run exclusively on the page that contains the applet, ActiveX controls can also be used after the user has left the page, allowing other programs to use them;
2. extensibility, since developers can generate ActiveX controls that accept different parameters in such a manner that it is possible to control the appearance and actions of these;
3. speed, in that since the Web browser saves the ActiveX control on the hard disk, it is sufficient to extract this control only once unlike Java applets that must be extracted every time, even when accessing the same page several times at different times;
4. power, as the activities of the ActiveX controls are not limited by a sandbox and can carry out more complex operations such as saving the result of the operations on the hard disk, the transfer of data, other applications and so on. Sandbox, on the contrary, prevents Java applets from writing on the hard disk, or even from them accessing the information contained on the computer itself.

From what has been seen so far, it can be inferred that ActiveX is characterised by merits and defects from the point of view of the security of our computer. As the browser saves the ActiveX programs on the hard disk, any security issues that may arise are even more serious than those relating to Java.

The main problem of executable programs and ActiveX objects that are downloaded from the Internet is represented by the fact that an attacker can create dangerous executable code. The problem of dangerous code is a significant one for all codes that are downloaded from the Internet and not only for ActiveX controls. Attackers can, without any problems, create macros, applets or Java applications that operate outside of the sandbox, browser plugins that are able to perform destructive or otherwise dangerous activity.

It is therefore very important to check the security of the software to prevent being attacked on the network by hostile programs. These programs can result in damage to the system through the introduction of a computer virus or can ensure unauthorised persons access to confidential information such as, for example, a credit card or a bank account number with access passwords. The software can cause problems for end user even unintentionally, by accepting a single download request and starting a hostile program during normal navigation on the Internet.

Before the intensive use of Internet, software was sold in a sealed package with a significant manufacturer's guarantee and this mode ensured that such software could be guaranteed and virus-free. With the rapid spread of the Internet, software can be downloaded from the network and is, therefore, devoid of any packet that allows verification of the identity of the manufacturer, and, when it is downloaded and installed on our computer, we do not have certainty surrounding the purpose of the software itself. In addition, since most producers do not use additional protection mechanisms, such software can be modified during transmission for purposes hostile to the user.

In order to meet the security needs of manufacturer and end users, ActiveX provides two complementary mechanisms represented by the levels of security and authentication. The level of security of an ActiveX component allows the user to know the cases where it is permitted to run the component with a certain level of security. Authentication of the component ensures the use of a species of electronic packet that allows identification of the software and prevents anyone being able to change it. The security model of ActiveX is similar to the security model of Java applets and is shown in Figure 5.70.

As can be seen in Figure 5.70, the ActiveX controls are characterised by a level of access to a user's computer that is even higher than allowed for Java applets. The level of security of an ActiveX program indicates the circumstances in which a user can execute the program with relative security. An ActiveX

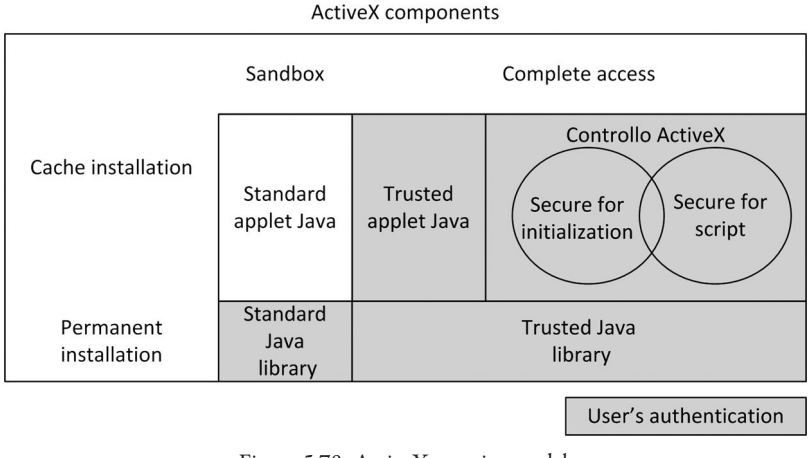


Figure 5.70 ActiveX security model.

control represents a particular type of program that can be used by other Windows programs to carry out particular activities. In this sense, ActiveX defines two types of components depending on the access level of the component in respect of system resources.

The first type of component is sandboxes that are characterised by limited access to resources of the user's system. A component sandbox cannot, for example, access the hard disk of the user. This type of component includes applets and standard Java libraries. Sandbox components are secure provided they remain within the sandbox. ActiveX allows Java applets to access Component Object Model (COM) objects that are external to the sandbox. ActiveX denominates the applets and libraries that work externally to the sandbox as Java applets or trusted libraries. However, it has already been said that it is not possible to guarantee the security of applications that work external to the sandbox and for this reason it is of vital importance to know the identity of the person that created the applet or the library.

Components that operate outside the sandbox such as ActiveX controls or trusted Java applets are also called full access components and in this case ActiveX uses an authentication process. Within this context, Microsoft developed *Authenticode* technology. Because of the authentication, it is possible to know the identity of the person that created the control, also providing a means to contact the author in the case where such control performs incorrect operations on our computer.

As can be seen in Figure 5.70, ActiveX uses authentication on standard Java libraries, including those found within the sandbox. It extracts libraries as permanent installations and this means that the libraries themselves require a further level of security. The authorising of a program to check for other software components of course presents more security problems. Trusted software also, when activated by other hostile software, can generate extremely dangerous results. Trusted software, if controlled by hostile software, may also even be able to delete all the files from the hard disk. To meet security needs, ActiveX supports two security attributes represented by initialisation security and script security.

A control is secure for initialisation if the same control does not behave incorrectly, apart from the initial values that the user assigns to this control. When the browser loads an HTML page that contains an ActiveX control that is not secure for initialisation, it is able, depending on the security settings set by the user and the version number, to show the user a security dialogue window.

A control is secure for the purpose of the script if the control itself behaves correctly regardless of the manner in which a script may act on this control. When the browser encounters an HTML page that contains a script and an ActiveX control that is not considered secure for the use of script, it will display an appropriate dialogue box to ask the user whether to continue with the action or not. This window can be generated by any scripts.

The components sandbox cannot work outside of the sandbox and for this reason cannot be linked to dangerous behaviour. ActiveX components, on the contrary, even if they claim to be secure for initialisation, or secure for scripts, can also operate incorrectly, regardless of any statements concerning their security, since these statements are dependent on the creator of such control and are based uniquely on the relevant certificate rather than on a test performed on the field. In practice, to be certain that a control is secure for initialisation or the script, the source code must be carefully analysed.

5.14.3 Web cookies

The concept of cookies and their use in browsing the Web have already been illustrated in chapter 1. They are, in practice, the text files that are stored by the browser on the computer's hard disk. While surfing the Internet, the various sites that are visited generate cookie files that are stored on the hard disk of the computer in order to preserve the useful information during a user's visit and on subsequent visits, it is possible to use the data of preceding navigations to customise according to the tastes of the user and what the same has viewed in previous visits for subsequent visits.

The first browsers stored cookies in a single file called "cookie.txt", in which all the information relating to previous navigations were found. Current browsers, on the contrary, generate individual

cookie files for each site visited. These files are stored in the temporary folder of Internet files and can be deleted at any time using the corresponding commands that vary from browser to browser.

Many servers use cookies to collect demographic information and site access preference. Other servers store in cookies the elements that the user has selected but that has not bought, in order to propose them again on subsequent visits. The server may also use cookies to send the user emails or to gather information from other servers on the sites where the user has made purchases. The server may also store in cookies information relating to the user that could in any case cause problems if they ended up in the wrong hands.

Cookies are, therefore, very useful because they help remote sites to provide the best service to its users but can also be used for hostile actions. For this reason, there are usually options on the browser that allow their use to be disabled but in any case, it is good practice to delete them from the temporary files folder that is used by our browser, using the commands that the same browser provides.

In order to reduce the ability of websites to acquire sensitive information, anonymous browsing can be used, passing through suitable sites that act as intermediaries, offering, in most cases, services upon payment. These sites hide the identity of users and transforming them into anonymous surfers.

5.15 Scripts and security issues

Often, when browsing the Internet, we will come across pages that contain interactive content that the same developers have created through the use of scripts. Many sites use script languages for the processing of forms that request from the user certain information while other sites use scripts to adapt their profile to the needs of the user. The languages used to develop scripts can be normal programming languages such as C or C++ or languages specially developed to operate on the Web such as practical extraction and report language (Perl), Personal Home Page (PHP) and application server provider (ASP).

Web programmers use script languages to generate pages that accept input data by the user and, usually, immediately respond to such data by means of appropriate processing performed on the remote server. Script languages can represent a potential danger both for the client and for the remote server. In fact, in terms of the client, a hostile script could be created that, once downloaded by the client, may attack the computer itself. However, in relation to the server, an attacker could use a script to attack the same. In the following section, we address the scripts and the potential dangers that they can represent for surfing the Internet.

5.15.1 CGI scripts

The CGI standard specifies a format of the data that the browser, the server and programs use to exchange information. A CGI script is a program written in any of the languages referred to above, which accepts input data introduced by a user and possibly generates, if provided, a response as a result of processing that occurs on a remote server. This response is usually returned through an HTML page that the remote server sends to the browser of the user. Is it possible to use CGI scripts on our website to generate dynamic pages that are able to interact with user responses.

Scripts are used intensively on online selling sites or in sites that provide information of any kind, in order to return the HTML page with the required information in response to a question. The main purpose of CGI scripts is, therefore, the possibility of dynamically generating HTML documents.

The creation of a Web server that dynamically generates Web documents is not a particularly complex process with the software tools that are currently available. It has already been stated that an HTML file is an ASCII text file that tells the browser how to graphically present the data contained in it. Usually, the designer generates an HTML file and loads it onto the Web server to be viewed by

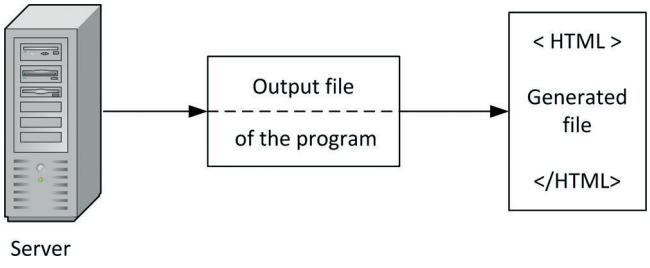


Figure 5.71 Example of writing of an ASCII file to a dynamic HTML page.

remote users that request it. In some cases, this is extremely useful but there are cases where explicit requests are made to the server that are difficult to predict and, as such, it is not possible to provide, in advance, the relevant Web pages. In this case, dynamic Web pages are used that are created at the time according to the user's request.

Through the use of any programming language, it is relatively easy to generate an ASCII text file, since the same program must write data on a file using the normal output routine. When the program writes data, it calls on its routine to write the data on a file. Figure 5.71 shows how to operate this program.

Since the Web server is a simple program, it can also generate an HTML file by adding a suitable routine and by changing the code of the program so that the server sends the ASCII text file to the browser of the user. This mode is shown in Figure 5.72.

As the server is capable of generating HTML text files, the reason why CGI scripts are being used may not be clear. In this sense, we can say that it is not necessary to use the above script to generate dynamic HTML pages as these can also be generated in another way. In any case, without the use of a script language, we would need to modify the server program from time to time and, with the passing of time, these changes could generate very large and complex Web programs. It is for this reason that CGI scripts are used. By using a CGI interface, the server delegates the task of dynamically generating Web documents to a specific application that responds to certain needs. The program can be written in C, C++, Perl, Javascript, VBScript and in various other languages. In practice, CGI files are very useful for designing a new website as it is possible to modify or replace CGI scripts with the new code without the need to rewrite or recompile the server program. It has already been shown that when a browser wants to communicate with a server, the same performs an http transaction in four steps, as shown in Figure 5.73.

CGI scripts can be found on the server and thus the server and the programs are able to communicate directly and this direct communication allows a CGI script to receive data dynamically and to provide such data to the server.

The *Java Virtual Machine* has already been addressed that is usually part of a Web browser, which processes locally the files of Java applets and that executes them on the user's computer. Javascript and VBScript programs are also executed in the user's browser. CGI scripts are mostly run on the server and for this reason, in the case of Java and JavaScript, it is said that these are programming languages for the

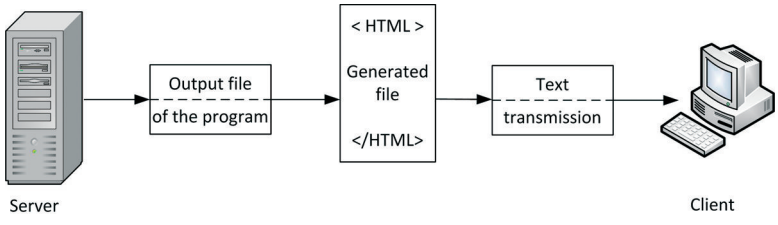


Figure 5.72 Example of writing a dynamic HTML page to send to a remote client.

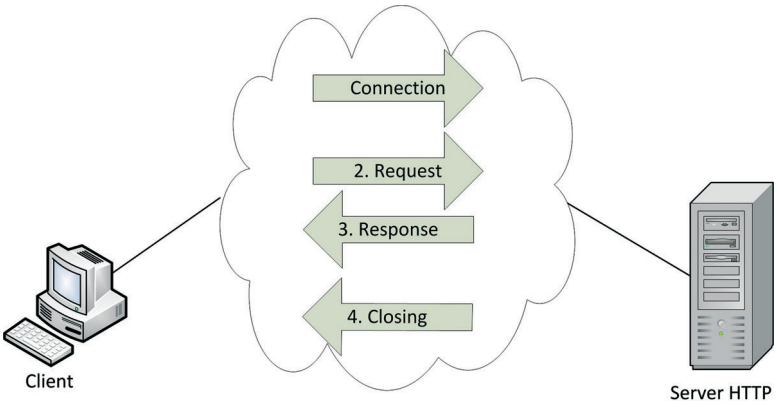


Figure 5.73 Transaction in four steps for an HTTP connection.

client side while in the case of CGI scripts running on the server, it is said that these are programming languages for the server side. In this sense, the generic user cannot perform a CGI script in their browser.

Servers that use CGI scripts must maintain these scripts on the computer where the server is found: in order for the user to be able to display the output generated by the script on their Web browser, the server must run the script itself and send the results to the user's browser.

In the four step transaction, the browser, after having established a connection with the Web server, requests the document. If the requested document exists, the server responds by sending the same document to the browser. Once the request has been granted, the server should close the connection while, in reality, it waits for a certain period of time to satisfy any further browser requests.

When a CGI script is written, the only modification that is carried out is that occurs on the server side, and in this sense, the browser is not even aware of what is happening on the other side of the communication. In this sense, the CGI script does not need any further operations in the process of http request and performs no interaction with the Web browser. When a browser contacts the server to request data that is generated by a CGI script, it will be the responsibility of the server to run the actual script. The script, once activated, processes the data needed to produce the required output. Once the server receives the response from the script, in HTML format, the same can, if necessary, add any header information and send everything to the Web browser. Having done this, the server awaits any further requests from the Web browser.

When the server activates the script, it must provide the script with the data required that the user has sent via the browser and manage its output, adding any information that may be needed to the browser to correctly interpret the data that were sent by the script. It has already been said that http is the protocol by which Web servers and clients communicate with each other. The http header information is fundamental to allow efficient communication.

The specifications of the CGI interface push the server to deal with the provision of data to a CGI script and manage the information that is provided by the CGI script. In order for CGI scripts to comply with the CGI specifications and to operate properly with the http server, it is important that they meet the requirements defined by the CGI specifications.

When a Web user selects a link that generates the call on the server of a CGI script, the server must provide the above-mentioned script with information that also depends on the script itself. The server must pass the information given by the user and any other general information. The above information is used by the script using the command line together with the environment variables that the script must use as input. Environment variables contain information about the browser that made the request, the server that is handling the request and any other information that the server may need to

provide to the script. The CGI environment variables make a distinction between uppercase and lowercase letters and are managed according to CGI specifications.

ASP pages are used very often in websites to create pages whose appearance varies from user to user. An ASP page uses both HTML code and script language, such as, for example, JavaScript or VBScript. To generate an ASP page, the programmer generates an ASCII file within which it enters both the HTML code and the language instructions. When the user requests the page using our browser, the Web server first and foremost runs the instructions on the page that will be replaced by the HTML instructions generated. Having done this, the server sends the result to the browser.

It has been seen that the CGI script uses environment variables to operate correctly and these variables are passed to the server that generates them according to the request made by the browser. Unfortunately, the environment variables can be used by an attacker to help him/her breach the server. In fact, until recently, these variables were used to attack the server generating an overflow error. An overflow error takes place in the buffer when an attacker is able to send more data than that the script variable is able to receive in input according to the specific variable. Depending on the software used by the server, a buffer overflow error causes blocking of the script that allows the attacker to gain access to the server. It is therefore very important when generating scripts that the capacity of the variables is sufficient to accept the data that the user can send to the script and that in any case the server is able to deal with any overflow without leading to a block.

The script can be attacked in other ways. For example, when a file receives an ASP page, the page that contains the script is an ASCII file. If an attacker manages to obtain access to a certain site, he/she can modify or replace the script file, for example, to acquire credit card numbers, if this is an e-commerce site, which will be sent via email. To avoid this kind of attack, a site should enable protection on the files and folders and use audit trails to control any system violations. It has already been seen that an attacker can enter the server in the middle of a session, putting himself between the user and the server, managing to intercept the messages sent by the user to the server and modifying the data transmitted. To avoid this problem, secure connections must be used that employ encryption of the information exchanged between the client and the server. An example of interception is shown in Figure 5.74.

5.15.2 The languages used for creating scripts

It has already been stated that when creating CGI scripts, different programming languages can be used. In essence, CGI scripts can be created using any programming language provided that such language is able to read and write in the format of input and output provided. Apart from the

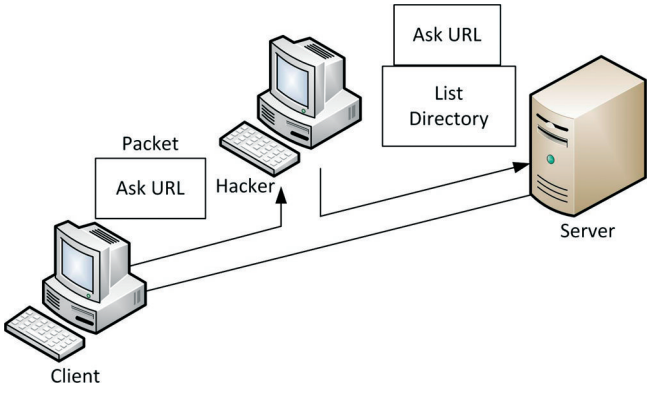


Figure 5.74 Example of how an attacker intercepts a session between a client and a server.

programming languages that are very popular such as C and C++, other languages such as Perl, Visual Basic, ASP, PHP, JavaScript and UNIX shell commands are also used.

A good number of scripts created in C and C++, Perl and other programming languages are used. Scripts do not, however, represent the only means of interaction with users. In addition to CGI scripts, remote servers also use server-side assists (SSA) that allow servers to reprocess the input of the users and generate HTML response documents, avoiding having to run another program as in the case of CGI. In addition, instead of performing a specific program, certain servers use the API interface containing functions that process input and generate an HTML response in output. Regardless of the method that is used in a site, the procedure remains the same. The user sends the data and the software that resides on the remote server processes the received data, generating an HTML response. Each of the three possibilities, represented by script, SSA and API, is characterised by advantages and disadvantages.

The SSA system uses the same server program. It is clear that a server is a program that is very complex, and the fact of having to modify the server source code every time we need to process the data is not an efficient or functional solution. In addition, an error in the code of the server can be a serious problem for the security of the system. The great advantage arising from use of SSA is high speed. In fact, in most operating systems, one of the operations that requires more time is the execution of a program: if the server could respond directly to the sending of data, there would no longer be the need to continually utilise an external program, reducing the time required for execution.

The advantage of the CGI script is the ease of use. They are very simple to create and also allow easy creation for non-specialists in the field, at least for scripts that are not particularly complex. The disadvantage is that the server must activate an external program and if the program itself contains errors or vulnerabilities, this program may represent a considerable risk for both the server and for the user. The advantage of the APIs is that the program is not inside, but is located in a trusted place and not in a program written by third parties.

5.15.3 Perl language

Perl is a programming language that for a long time was used to quickly generate scripts that are executed by the server. Perl is an interpreted transportable language that is suitable, in particular, for applications that require the processing of texts. Perl is capable of supporting the structured programming typical of high-level programming languages and provides functionality that comes from the evolution of language in UNIX environments.

Perl is a language that is available for free and has not been used for many years: in its place, PHP language was used that appears to be very similar to Perl as it is a mixture of the latter and ASP. Perl was developed in 1986 in order to create reports from UNIX environment text files. The author of the language has gradually added new features and has made it available to the public, causing its popularity to grow, becoming one of the preferred programming languages by users of the sector.

Perl is characterised by a structure similar to the C language, which it resembles significantly at first sight. Most operators and C structures are within this, with the exception of pointers, the structure and definition of the types.

Perl is characterised by very useful and versatile functionality such as:

1. associative arrays that programs can index using non-integer keys;
2. between integer type automatic conversion, floating-point numbers and strings;
3. automatic resizing of arrays;
4. conversion functions of binary data;
5. extended support for regular expressions with search and substitution capabilities and other operations of text analysis;
6. input/output functions on file;
7. output functions prepared in a format that can be selected at will;

8. group of C operators and string comparison operators;
9. manipulation functions of lists that can support stacks, queues and other types of data;
10. system functions;
11. instructions and control structures.

In CGI programming, as indeed in all areas of network programming, security is a factor of vital importance. In many cases, it is necessary to protect files and other resources of our system from users who are not very practical or particularly honest. This aspect is particularly important when Web servers that are connected to the Internet are concerned, an area in which there are a large number of potential attackers. A possible way to protect a system against attacks is represented by passing all data through a gateway in such a manner that the gateway itself only allows the data to enter that it considers secure. In many cases, Web servers run under UNIX and are created in C, which is a very powerful language but that is subject to possible security flaws if we do not pay particular attention to the commands that access the system and the computer, such as pointers.

One of the great advantages of Perl is the possibility of producing secure gateways due to the fact that string type variables can increase automatically, reaching the sizes that are required to contain all the characters that the script is assigning to a variable, thus avoiding system blocks that are dangerous for security. There is also another version of Perl that prevents passing of data to the server that comes from non-trusted sources. Perl is also very useful for database querying, being equipped with very powerful features.

Its Interface (CGI) allows sites to interact with client programs, usually by browser. In most cases, when a server and a client must exchange data, scripts deal with access to the database that contains the data required. Due to the CGI interface, it is possible, for the general user, to have access to a database using a normal Web browser. In this case, the CGI script receives as input the data of the user that it processes by connecting to the database, running the query, preparing the output data according to the HTML format requested by the user and sending it to the latter. It is important that all this takes place securely.

In an operating system, a shell (or terminal) is a program that allows users to communicate with the system and to start other programs. It represents one of the main components of an operating system, together with the kernel.

The shell is the work environment through which it is possible to send the computer commands, requiring the execution of programs. A specific class of such programs are shell scripts.

There are many types of shell, which are mainly divided into textual and graphic:

1. When reference is made only to “shell”, or even “terminal”, this is commonly understood to denote a textual shell with which the user interacts through a terminal or terminal emulator (the latter may also be a part of the shell itself rather than be a separate program) via a command-line interface.
2. In the case of shell graphs, it is common to refer to the so-called *desktop environment*, which provides users with a graphical environment from which to manage files and start programs.

Since Perl was created to replace shell commands, it ensures interaction with the system, including the option of activating external programs. It must, however, always be remembered that if a script is allowed to activate an external command, it exposes the whole system to risks from a security point of view.

5.15.4 CGI scripts and security issues

CGI scripts, if not properly programmed and used, can cause serious security problems. When data provided by the user are sent to the shell, script programmers must pay particular attention to the aspects of security and eliminate the meta-characters of the shell, preventing the latter from being able to interpret and execute specific commands. The following are some basic rules. First, it is very

important to eliminate from the input the following characters:'; < > * | ' & $! # () []: ' / ' that is provided by the user before passing the same to the shell. The end line half character is not on the list illustrated, whose elimination from the input row usually creates significant problems. Examining most CGI programs available on the Web, it is found that they are vulnerable to meta-characters. Practically, all the CGI programs that convert characters from a hexadecimal value to its ASCII character and then pass the converted input to a shell are usually vulnerable to attacks via meta-characters. It is also very important to check that all the temporary files stored in a *tmp*-type directory or in any other directory are deleted after execution of the script.

If an attacker manages to breach a CGI script, depending on the nature and location of the script, the attacker is able to perform a series of dangerous operations from the security point of view. The main operations may include:

1. sending password file emails that enables the attacker to decode the information received at a later date and remotely, without the risk of having to do this during real-time connection;
2. sending of email of the file system map under attack that enables the attacker to receive a great deal of useful information from a remote location;
3. sending of email of the system information that is taken by the computer directory under attack;
4. commencement of a login on the server on a high port and activation of a Telnet session;
5. beginning of an interruption of service attack by using the commands that involve high costs relating to the file system or system resources;
6. modification or cancellation of the records present on the server.

Attackers, when performing an attack via a CGI script, can attack a site where the Web server is running via the root user on a computer server. A Web server run by the root user is characterised by unrestricted access to the computer itself. In this sense, it must start the Web server as the root user in such a way that the server itself connects to port 80, which represents the http connection port. The privilege level of the Web server must then be changed into *nobody* or into a similar account equipped with generic privileges. Depending on the Web server being used, the configuration file of the server should allow specification of the user under which the server is run. Usually, the Web server uses the "*nobody*" user that represents a generic account without special privileges. In certain cases, we will need to take into consideration the possibility of running the server with a user code and identification of specific group for the Web server: this prevents a program that operates as *nobody* being able to interfere with the server files.

From what has been said, it is clear that system administrators are very wary of input data for CGI scripts that can cause serious security problems for the system itself if the attacker discovers any vulnerability. To make CGI scripts secure, it is very important to implement all possible measures of defence in relation to the input data.

In the computer field, it is said that a process creates a *fork* when the same replicates. A potential attacker can use the system commands to create a new shell by means of a *fork*. In the case of the UNIX operating system, the latter creates a shell, that is a space reserved in the operating system, for each user that connects to the server. When a user requests a shell fork from UNIX, UNIX generates another copy of the shell that run together with the first copy. This often enables the attacker to break into the system and for this reason it is very important to avoid creating scripts that allow users to create new shells. Unfortunately, from this point of view, there are several commands that allow execution of a shell fork.

In general, there are two solutions that enable secure CGI scripts. These solutions consist of only using data that may not give rise to problems at the time of installation and in the control of the data after receipt, to be certain that the same can be used with security.

With regard to Perl and its use for the creation of Web pages, it is very important to bear in mind the fundamental rules to ensure security of the system. It is important to remember that languages such as Perl are equipped with the "eval" command that allows creation of a string and to run this string via

the interpreter. Unfortunately, attackers can also attack scripts by sending to the scripts themselves a piece of data that starts with a semicolon and in this way attackers can make even simple scripts become dangerous. A prudent client deletes characters that have a precise meaning for the shell so that the script does not interpret these characters incorrectly. In fact, a dangerous script can use special characters to disorientate the script and gain a level of access to the high system. It is very important to turn off the *server-side includes*. As the server can support such functionality, the same must be disabled with regard to directories that contain scripts. Attackers that exploit scripts that generate all content in output that the same attacker can send to the script are able, with great ease, to use such functionality for their illicit purposes.

Besides C, C++, Perl and the problems of security, it has already been stated that other programming languages are used for creation of the script. One of the most used is JavaScript. It is seen that Java is a programming language that allows the generation of websites with animation containing graphics, audio, dialog boxes and other elements. Since Java is a true programming language, certain expertise is required to generate Java applets.

The creator of Java devised JavaScript to meet the needs of Web designers and programmers. In a manner similar to Perl, JavaScript is a scripting language that can be used for the creation of interactive Web pages. Microsoft, for its part, as a response to JavaScript, created VBScript that represents a programming language for script that is based on the structure of the Visual Basic and in fact the features of JavaScript and VBScript are very similar. Through the use of JavaScript and VBScript, it is possible to generate scripts that operate on the server, in a manner similar to Perl scripts or scripts that operate on the client's browser.

The great advantage of JavaScript and VBScript, compared to other programming languages such as Java or C/C++, is that the former do not need to be compiled and can be run directly on a Web browser. Since JavaScript operates on the client's browser and not on the server, the same can also perform operations of interaction in real time.

Using JavaScript and VBScript, programmers can insert instructions in the HTML files that will be executed by the browser. When a user visits a website containing a program in JavaScript or VBScript, he/she does not realise that the browser is usually executing the program instructions. To minimise the possibility of an attacker using these languages to create the virus or to read confidential information within the user's computer, the browser limits the operations that can be performed by a script. In this sense, a script may not write data onto the user's hard drive, with the exception of cookies, and for this reason cannot introduce any viruses on the hard drive within the computer. Moreover, both JavaScript and VBScript cannot read information contained in the files of the user.

As browsers have undergone, and undergo, a process of continuous evolution, they were designed to prevent hacker attacks. In this sense, any mistakes that allowed attackers to attack a system using JavaScript and VBScript were corrected, and in this sense, now, the use of such languages in our browser presents virtually negligible risk. Hackers are, however, always in search of new methods, techniques and vulnerabilities to attack systems and for this reason it is not said that they will not succeed in the future, also with regard to the above languages, considered secure at the time.

5.16 Computer viruses and security policies

Most people that surf the Internet or in any case that have used a computer have had to deal with a computer virus. It is well known that such hostile programs are capable of causing very serious problems to the computer being used, the most hostile ones even leading to the destruction of all the data contained on the hard disk, an event that can lead to considerable economic losses if it occurs in large operational, commercial and financial organisations. From this point of view, it is clear that the costs of restoring a system attacked by viruses are certainly higher than a correct policy of prevention.

The early 1990s were a period of relative calm from the point of view of virus attacks and most of the infections took place through the exchange of floppy discs (limited memory capacity diskettes that were used at the time). As a result, the massive diffusion of the Internet has increased the quantity and speed of the exchange of information as well as the connectivity of the various computers, increasing the risk of spreading of computer viruses that in the meantime were created to be more and more difficult to identify by antivirus programs and increasingly destructive and dangerous to our computer.

A study of the National Centre for Supercomputing Applications (NCSA), which was performed in 1997, found that 99.33% of the organisations studied had recently experienced problems with viruses. In 1996, the monthly frequency of virus infections was 10 computers in every 1,000. By 1997, this figure had already tripled. A subsequent public study highlighted the fact that in relation to computer virus infections, every 100 computers, at a minimum, doubled every year during the period of 5 years taken into consideration, reaching 160 computers in a 1,000 in the year 2000. This trend is still growing.

The following considerations have also become evident:

1. Annual losses caused by virus infections for companies were estimated to be in the hundreds of thousands of euros.
2. Of all the companies examined, more than 40% had experienced data loss caused by virus infection.
3. A total of 70% of the companies had problems with email due to virus-infected files upon receipt.
4. The frequency of infections reported was 99.67%.
5. Major virus infections, where at least 25 computers were involved at the same time, affected 51% of the companies.
6. Of the 51% mentioned above, 80% received the virus via email attachments.
7. A total of 64% of server outages caused by virus infections lasted more than an hour and the average time of inactivity was equal to 21 h.
8. A total of 70% of the computers had an automatic and continuous antivirus protection.
9. A total of 76% of the organisations considered that the problem of virus had worsened compared to the previous year.

At the time when the research was performed, it was found that boot virus infections, or viruses transmitted via floppies, were almost nil. This is normal because, as mentioned, the research was carried out when floppies were barely being used any more, and the use of email for exchanging files was just starting. Viruses are not only aimed at computers and August of 2000 saw the first case of infection of a personal digital assistant (PDA).

A virus that, in reality, may seem very simple, in practice, it is not easy to define rigorously. The tendency is generally to group together under the single name of viruses of various types represented by viruses, worms, Trojan horses and so forth. This name is quite reductive as it does not include all the types of such programs. In addition, there continues to be some confusion with regard to the precise definition of the elements that make up a virus. A virus is usually divided into three functional parts represented by replication, concealment and bomb. The combination of these parts makes the program as a whole a virus.

5.16.1 Replication

A virus must include at least one replication method, that is a way to copy or duplicate itself. The process of copying in a file is also called infection. As most people would never intentionally upload a virus into their computer, replication ensures that the virus is likely to infect other computers.

Replication takes place when the virus is loaded into the memory and has had access to the system as the virus cannot replicate if it is present only on the hard disk. For this reason, the infected file must be executed by the computer to be able to begin hostile actions for which it is programmed and in order

to be able to start replicating. The replication method falls into the category of infection of the file or into the category of boot replication.

With regard to file infection, it may take place if and only if the virus is executed and has access to the computer. For example, a Word document does not directly run any type of command in the memory. On the other hand, Word is capable of reading macro commands contained in documents and of running them in the computer's memory. In practice, it is infected with the Word document but in substance it is the Word application that represents the means through which replication occurs.

The most widely known type of infection is directly executable files, that is from files with the extension COM, EXE or BAT. In this case, the virus adds a small part of the code at the beginning of the file that ensures loading into the memory of the computer before actual application, when the file is executed. Once this has taken place, the virus enters the part of the code that remains within or at the end of the file.

Once the file has been infected, replication can fall into two distinct categories: resident and non-resident. In the case of the resident virus, once the same has been loaded into the memory, it waits for the execution of other programs, infecting them in turn. Some viruses have shown that this infection is also possible on systems with protected memory. In the case of the non-resident virus, the latter chooses one or several executable files on the disk and infects them without waiting for them to be processed in the computer's memory. This type of infection takes place every time the executable file is started.

The virus often uses the search order of the extensions of the operating system in order to facilitate the loading of its code without infecting, in practice, the existing file. This type of virus belongs to the category of companion viruses that operate ensuring that their executable file is activated before execution of the legitimate file.

It should be remembered that, in the case where only the file name is entered, the DOS and Windows operating systems usually try first the one with the COM extension, then the one with EXE extension and then the one with the BAT extension. In this sense, virus programs often try to assume the name of a widely used file, shrewdly taking the COM extension, in such a way to be able to be executed before the legitimate program.

With regard to boot replication, in this case the viruses infect the area of the disk that is read when the disk itself is read or is started. This region can be master boot record, the boot sector of the system or both. Viruses that use both replication of the file and boot are also called multipartites.

A virus that infects the above areas usually moves the instructions of the system into other areas of the disk and inserts its code into the boot record. When the system is initialised, the virus is loaded into the memory and indicates the new position of the system instructions. In this way, the system is booted normally, but the virus succeeds in being loaded into the memory.

A boot virus does not require the execution of programs from an infected disk to facilitate its replication: it merely needs access to the disk. In fact, in most cases, computers run a system check upon start-up to check functionality of the disk. The verification process in question is sufficient to activate a virus in the boot, if there is one.

Boot viruses need a disk to disk contact, or even pen drive to disk to replicate. These supports must be connected to the same machine. For example, if we access a shared directory on a system that has boot sector viruses, these viruses can be copied into the local computer. This is because the two computers do not share memory.

There are also programs, known as *dropper*s, that are able to increase the spread of viruses in the boot, also through the network. A *dropper* is a set-up utility for the virus that is coded to hide the virus that it contains, trying to avoid its detection by antivirus programs. In addition, the application tries to convince the user of its usefulness, seeking to be executed: once activated, it installs the viruses on the computer. Through the use of a *dropper*, an attacker is able to infect through a boot virus, passing through the network. Once the virus has been released, in order to ensure replication, there must be disk access.

The methods of infection illustrated have one characteristic in common: the virus must have a self-revelation system. This characteristic is necessary to avoid auto-corruption due to double infection of

the same. In fact, if self-corruption has taken place, the virus could lose its effectiveness and the user may notice its existence on their computer and in any case the replication process may stop, causing the virus to be extinguished.

One method that is widely used both by the creators of viruses and by antivirus programs to prevent the duplication or double infection of files consists of inserting a string note into the virus that is unique for the virus. The virus itself, before infecting a new file, checks if the string is in the file itself: if the string is present, this means that the file is already infected and the virus does not continue, otherwise the same proceeds with the infection. This string, once the virus has been discovered, is the same one that is looked for by antivirus programs in an attempt to control all the programs and all of the files on our computer. This attempt is therefore a double-edged sword for the creators of viruses.

One category of viruses is macro viruses. They are performed by an application, rather than by the operating system. They are mostly written in Visual Basic for Applications (VBA) and performed by the Microsoft Office package, composed of at least of the following programs: Word, Excel and Access. In this sense, macro viruses hide in documents and spreadsheets and are spread by suitably infecting similar documents even though, often, they may infect, damage or destroy other system files.

Macro viruses represent about 80% of all known viruses and their number is constantly growing. Their greatest danger is their ability to infect the files at any operating time of the application: opening, saving, editing or deletion of the document. As it is not particularly difficult to learn how to use VBA, no particular expertise is needed to generate such macro viruses.

5.16.2 Concealment

In order to replicate with greater ease, a virus must be able to hide its existence on a computer. If the virus was visible in the taskbar, its existence would be evident. There are various methods used by viruses to hide their existence and in the following only a few will be discussed.

Viruses are usually characterised by small memory occupation, for obvious reasons. Even a large virus usually has a memory occupation of less than 2 Kb. This allows the virus to hide appropriately on local storage media and to reside in the memory without taking up a great amount of space, reducing the likelihood of detection. To minimise memory occupation, most viruses encode directly using the language of the microprocessor, called assembly language or machine language. In addition, if the virus is small, the same can be attached to other files without substantially increasing the size. There are also viruses, called cavity virus, that look for repeated sequences within a file, looking at null sequences, installing themselves in these areas and hiding themselves within the file itself without changing their sizes.

The first computers operating under the DOS operating system, in order to protect the files from virus infection, set permissions for read-only executable files. It was believed that if the file was not editable, the same could not be infected with a virus. Creators of viruses have circumvented this problem by adding a code that allowed them to control the attributes of the file prior to infecting them. If the attributes were set to read-only, the virus changed those attributes, infected the files and reset the original attributes. This security method was therefore inadequate both for the operating systems at the time and for current ones.

This concept is not applicable to multi-user environments within which the level of permits varies from user to user. If administrator privileges are required to change the file permissions, the virus is not able to vary the attributes when the same is executed from the account of a normal user.

The lack of security of accounts is the main reason for spread of a virus in DOS, Windows (not all) and Macintosh environments. In this sense, there are a few viruses that have been created for the higher category UNIX and Windows environments as the possibility of deciding on permissions for the files reduces the possibility of the virus to replicate and infect other files. For this reason, virus programmers have turned their attention to other more vulnerable platforms within this context.

In addition to access permissions, viruses are also capable of varying the date and time of a file. This operation is performed to be certain that a user is not aware of the possible infection of a file looking at the variation of the temporal attributes. The first antivirus systems controlled changes in an attempt to identify viruses: since most viruses are able to restore the original date and time after infection, this technique has been abandoned as it is not effective.

Different techniques are used to conceal the virus including the technique called *stealth*. This technique allows a virus to hide changes that are made to a file or to a boot sector. When the virus is loaded into the memory, it controls the calls of the system to files and to the disk sectors: when a call is intercepted, the virus modifies the information that is returned to the process in such a manner that the call shows the original information that is not infected. This operation allows the virus to stay undetected.

Many boot viruses are equipped with this ability. If the infected disk is started by loading the virus into the memory, programs such as fixed disk (FDISK) sees a normal boot record. This takes place when the virus intercepts calls to the sector that are carried out by FDISK and returns the original information of the boot sector. If we start the system from a disk that is not infected, we do not have access to the unit. If we execute FDISK again, the program returns a corrupted boot sector on the same unit.

The concealment may also be conducted by varying the information that is returned from utilities such as Directory (DIR) and Memory (MEM). This allows the virus to conceal its presence on the local storage medium and in the physical memory. To use *the stealth* feature, in any case, the virus must be executed actively in the memory and this means that the concealment component can be detected easily by antivirus programs.

Some viruses are able to put into practice certain techniques to avoid identification. These viruses constantly monitor the system to check if there is an active antivirus software program, taking appropriate measures to avoid identification. This ability represents an active *stealth* characteristic. Some viruses control the activity of the system once they are active in the memory. If the same realise that a scan for viruses has been started, they are able to deceive the control software into believing that there is another virus in the computer. As the virus identified requires destructive cleaning that damages, though only slightly, the system, the virus takes advantage of this to install itself in the system file until a system recovery is performed, after which it resumes infection of the new configuration. Even in this situation, it is vital that the virus is active in the memory to monitor the activities of the computer. For this reason, it is very important to start the system from a boot disk, in most cases external and inserted at the time, of which we are certain of the absence of viruses, before starting any virus elimination activity. In DOS systems, it is essential to ensure effective restarting of the system as many viruses are capable of capturing the start sequence set by the user and generating a false start, deceiving the user and remaining active in the memory.

The creators of viruses also use encryption that allows them to hide calls to the system and the text strings that are located in the hostile program. As the virus code is encrypted, the creators make it more difficult for antivirus programs to find the virus. Revelation is not, however, complicated because most viruses use a very simplified version of cryptography and the same key for all the code. This means that the sequence of decoding is the same for all viruses while recovery of the virus code may be difficult. If we manage to breach the decoding key, we can use the same to locate all future versions of the virus. Even if the decoding key is not violated, it is always possible to find the typical encrypted string of the virus, which has already been addressed, to operate the antivirus program correctly. The answer, in terms of quality, of the antivirus program depends on the encrypted string that has been found. It should be remembered that the antivirus program is not able to know if the same is looking for encrypted or plaintext information and if the encrypted string is very simple with a non-hostile file code, the antivirus program may have difficulty in distinguishing the infected files from files that are not infected.

Another way to hide viruses involves polymorphic mutation. A polymorphic virus is able to change its signature from infected to non-infected file while remaining operational. Most antivirus programs try to detect the presence of hostile code by looking for the signature code. As a polymorphic virus is able to vary its appearance during the various stages of infection, it is more difficult to identify it. A polymorphic virus can be generated by inserting a series of cryptography schemes that use different encoding procedures. In this case, an antivirus program is not able to identify all possible variations of the virus since it does not know all the procedures of decoding. This becomes virtually impossible if the virus uses a key or a random sequence during the implementation of cryptography. Many viruses include a benign code, also called dormant, which is moved within the virus before encryption without affecting the operating capacity of the virus itself. In this way, the encrypted string generated by the process changes depending on the instance of the virus as this varies the code sequence. The most effective way to generate a polymorphic virus consists of the insertion of links in a known object module, also called mutation engine. Since this engine is modular, it can be added in a simple manner to any existing virus code. This engine change includes a random number generator that allows alteration of the ciphertext that is obtained. Because it uses a random number generator, the ciphertext that is obtained becomes absolutely random and will change with the various replicas of the virus. In this way, the virus becomes extremely difficult to locate even among the same versions of the virus.

5.16.3 Bomb

Once the virus has replicated with success and has not been identified, it is ready to act and carry forward the actions for which it was programmed. The majority of viruses are programmed to wait for a specific event that can be very simple, such as the arrival of a date and a specific time or infection of a number of files or even the identification of a certain activity. Once the event has occurred, the purpose of the virus becomes clear because the same manifests itself. This object may not be particularly pernicious, consisting perhaps of the simple playing of music on the computer speakers but can also be extremely dangerous, for example the cancellation of all the files on the hard disk of the computer.

In practice, a virus behaves like a real bomb that explodes at a scheduled time. Many bombs are able to perform a dangerous activity in DOS and Windows environments because they do not provide a clear separation between the operating system and programs that are running. A virus may have direct access to lower level features and this option is possible because the operating system considers the programs reliable. For example, the DOS and Windows applications refer directly to the memory and the interruptions table. This operating mode improves the performance of a program and allows it to circumvent the operating system but at the same time it ensures a range of other features that a virus needs in order to use the *stealth* techniques. There are, in any case, limits to the activities that can be performed by a bomb as the same can make the computer unusable from the software point of view but it is not capable of causing physical damage: in the worst case, the entire hard disk must be reformatted, the operating system must be reinstalled along with all the application programs with considerable loss of time and money if this operation has to be carried out within an organisation on a certain number of computers.

There is a class of viruses that is not particularly dangerous but that, nevertheless, generate panic. It is the so-called hoax virus and is based on social behaviour. They have the same characteristics of a normal virus but for their spread they rely on persons rather than the virus itself. In most cases, this is an email message that informs us of the existence of a dangerous virus that is transmitted through email and that is able to delete all the files on the computer. People who receive it then make a point of sending this message to all their acquaintances. This fictitious virus is equipped with the main characteristics of viruses, that is:

1. replication, because such viruses rely on good intentions and on the ingenuity of replication because in most cases we believe in what we read without examining in detail and the virus is forwarded to everyone without any prior filtering;

2. concealment, as to disguise the danger, the virus uses a language that makes the message credible to the average user;
3. bomb, since the same is the waste of bandwidth because it involves the widespread dissemination of the message, in addition to the needless spread of fear. Needless fear because the message contains a disaster warning in the case where the same is ignored. The bomb is therefore represented by the manner in which these messages affect the resources of the computer and users.

In practice, no antivirus is able to locate this virus because the same, while having the three main characteristics of the virus, is, in practice, a hoax.

5.16.4 Worm virus

Worm viruses are an application that can replicate by means of a permanent network connection or a switched line (increasingly rare these days). These worms, unlike the traditional virus that is spread on the hard disk or on the computer's file system, represent the programs that are supported autonomously. Usually, a worm takes its functional copy in the active memory of the computer, not writing anything on the disk. In practice, there are two different types of worm.

The first type works on a single computer, in a manner similar to a common application, using the network connection as a channel of communication to duplicate itself in other system and to transmit information. Depending on the purpose, the worm virus may or may not leave a copy of itself running on the initial system once it has been replicated in a new computer.

The second type of worm virus uses the network connection in a similar manner to a nervous system in order to be able to dispose of various segments of its code running on different systems. In cases, where there is a central node that supervises and controls all the segments scattered on various remote hosts, the worm is also called polyp or octopus.

Worm viruses have always had a negative connotation. One of its variants that were widespread in the past was the vampire worm. That particular worm by day remained inactive and at night became enabled, using the resources of the computers vacated for complex tasks and intensive processing. In the morning, this worm became dormant again. It was eventually discovered as a process not carried out correctly had blocked all the computers of an organisation. Worm viruses find their natural means of transport on the Internet, given its dissemination and widespread use.

The dreaded Internet worms or worm was introduced onto the network on 3 November 1988. In less than 6 h, this virus, composed of only 99 rows, has in fact blocked 6,000 Sun and VAX systems connected to the Internet. It was able to start a small process in the background on each computer that it encountered. This virus would not have been noticed if it had not been characterised by a lack of programming since the virus, before it infected a computer, did not check if the same had already been infected, causing a multiple infection of computers that led to paralysis of the same. Administrators were having to address a threat by which they appeared to have already been defeated from the outset as, once a computer had been cleaned, it was immediately re-infested by others and the whole process began over again. To solve everything, administrators were forced to disconnect the computer from the Internet, but this manoeuvre was counter-productive, since the same could no longer receive useful information on how to combat the same virus.

Once the initial confusion had passed and all the systems had been restored, this experience was used to understand and to identify a new threat that was looming on the horizon of the network and that is still active. This incident led to the creation of Computer Emergency Response Team (CERT), which is an organisation, that is responsible for the documentation and assistance in problems of computer security.

In addition to the Internet worm, there were several other network viruses created including Worms Against Nuclear Killers (WANK). This virus only infected DEC systems and used the DEC net protocol. It was therefore not able to use the IP network protocol. It:

1. sent email by reporting the breached systems, logon names and passwords;
2. varied passwords of existing accounts;
3. opened up further avenues of access to the system;
4. sought users on random nodes and called them directly using the function of phone dialling;
5. infected local COM files to be able to restart in case the same had been deleted from the system;
6. varied the advertisement banner to indicate that the system was infected;
7. varied the logon script to give the impression that all the files of a user had been deleted;
8. hid the files of the user after logon to convince the user that these files were had been deleted.

This worm virus has caused significant problems for system administrators who have had to deal with it.

5.16.5 Trojan horses

Trojan horses are applications that hide unpleasant surprises within a harmless or pleasant packet. The surprise is a process that performs a task hidden to the user, and certainly not appreciated while the official and visible application performs truly useful functions. The hidden application represents the true Trojan horse.

Trojan horses are to be considered different from viruses because they do not replicate or attach to other files. A Trojan horse is an independent application that includes the bomb in its original code and does not become dangerous via another application. There are, for example, a number of Trojan horses created for UNIX and targeted to replace the applications of existing networks. An attacker could replace the process of the telnet server with one created by him/her in such a way that the same performs its functions normally, not arousing suspicion in the user, but in the meantime storing the login names and passwords, sending them to the address it was programmed to. This Trojan horse appears to be very dangerous, as this allows breach at a later date of all the systems that have been contacted by the infected computer, thanks to the knowledge of login names and passwords. However, there are many other versions of Trojans that are not listed for reasons of space.

5.16.6 Virus prevention

The most secure method to check if a program is dangerous consists of a thorough analysis of the source code. Since virtually all programs are supplied already completed, a translation of the original code must first be performed in order to be able to study them and this operation is not economically viable for all organisations.

This technique guarantees 100% security. There are however several technologies that can be used to avoid an infection from viruses, each of which is characterised by advantages and disadvantages and thus the best technology consists of adopting several methods together.

In this sense, the verification of a criterion of access control is not a valid protection technology but may also prevent the spread of hostile programs. Access control should not be confused with the attributes of the file that can be altered with great ease by the virus. Access to the file must be managed by a multi-user operating system that allows the administrator to set the permission levels for the files in a manner that varies from user to user. Access control does not remove or identify the existence of a virus, but represents a method to allow computers to resist infections. For example, most viruses are confident that an infected computer has full access to the files: if a system administrator has modified these permits in such a manner that users are able to access executable files that they need, a possible virus will not be able to infect them. This method does not work with all executable files as there are some files that change autonomously during execution. In this sense, users must have written access to these executable files and the information of time and date may vary at regular intervals. In general, it is

not possible to know which executable files require write access and a number of attempts must be made.

Another technique used is checksum checking or checksum control, also called cyclic redundancy check (CRC) that represents a mathematical check of the data in a file. This testing allows representation of the contents of the file in numerical form. If a single byte of data within the file changes then so does the checksum value even if the file size remains the same. In this sense, a reference of a system that is not infected is first created. The CRC is usually carried out at regular intervals by searching for any changes to the file. This method has disadvantages as a CRC is not able to detect infection, only to report modifications of the files. In this sense, executable files that are written autonomously threaten the checksum test at regular intervals and even if the change was carried out by a virus, a CRC is not able to clean up the file. It must also be remembered that many viruses have been created in a specific form to fool a CRC in such a manner as to make the information on the file remain unchanged. A CRC is not an effective control against viruses but may represent a valid check against the substitutions made by Trojan horses as some of these programs are designed to replace an existing authentication service such as a telnet, a software client or an FTP server without modifying existing files but forcibly replacing them. This replacement is declared and reported by a checksum test while an antivirus would not be able to notice anything as the files do not contain any virus. For this reason, the CRC is much more effective in the identification of Trojan horses.

Another technique used to prevent hostile programs from taking control of a system is monitoring processes that observe the different activities that take place within a computer, trying to intercept any suspicious activity. For example, the BIOS of the most recent desktop computers is equipped with antivirus capabilities that if enabled, allows the computer to detect any attempt to write in the boot record. If a boot virus tried to install itself in this area of the disk, the BIOS would be able to stop it, requesting prior permission from the user of the computer.

However, there are a number of problems. The first problem lies in the fact that the virus and the normal programs are characterised by similar attributes and it becomes complex to distinguish them. For example, the FDISK utility activates the BIOS warning message, even though it is not a virus, since this activity is viewed with suspicion by the BIOS. In this case, this is what is called a false positive because the BIOS believes it has discovered a virus when in fact it finds a harmless program. This situation generates a second problem due to the need for intervention and competence on the part of the user. For example, a user who receives a message due to a false positive event must have sufficient knowledge to understand that it is a normal operation and there is no presence of a virus.

However, it is possible that there is actually a virus in the disk while FDISK activates, making the user believe that this is a false positive when in reality it is a real problem. Even if the warning message is issued when FDISK is activated and not upon closure, the user must always have certain expertise to discriminate and identify the problem correctly.

The matter of the distinction between a normal activity and a virus is even more important when it comes to controlling processes. For example, the use of the file management utility to erase files often produces false positives and the same thing happens when checking file changes, memory swap and other similar activities. These activities could be performed by a virus, but also by normal applications.

The only useful control of the processes is the alert by the BIOS even though the latter can also provide messages of false positive as in the case of implementation of the FDISK command, which is in any case a command that is executed when installing a new operating system, reducing the frequency of false positives to a minimum.

The most used technique to identify viruses is the use of dedicated and specialised antivirus software. These types of software use signature files to find the virus within an infected file. A signature file is an archive that contains information on all known viruses and their characteristics. These characteristics include the code of each virus, the type of file that is being infected with these viruses and all the useful information for identification. When using a virus archive file, it is possible to

download an update file periodically without the need to update the program as a whole, in such a manner as to be sure that the antivirus software is able to cope with the virus just introduced into circulation.

When an antivirus checks a file, it checks to see if there are any matches between the code of the file and the entries of the virus signatures archive: if a match is found, the user is notified with an appropriate message and the software can perform different actions, depending on the settings of the user, that go from trying to correct the infected file to its movement into a special folder of quarantined files, or even its deletion if that file is not of vital importance and the software itself is not successful in the repair operation.

But there is a limit of antivirus software, represented by its ability to locate only known viruses: there is, therefore, a temporal window of risk between the time when a new virus is issued and the time when the manufacturer of the antivirus software updates the signature archive and issues the new version that is loaded onto our computer. This problem is even more evident in the case of a polymorphic virus that is able to vary its signature with every new infection: in this case, the antivirus software should have loaded into the signature archive all the possible mutations of the virus in order to be able to locate it. In the case where the archive does not contain just the one mutation, the virus could easily infect a computer when the mutation occurs. For this reason, when selecting an antivirus software, one must be chosen that is able to find most of the viruses, including polymorph viruses and all its variants.

Compressed files and encrypted files can lead to problems with regard to antivirus software, since both vary the manner in which the information is stored and an antivirus may not be able to detect the presence of a virus hidden in them. This problem is even more evident for encrypted files given that the antivirus is not able to decrypt the files in search of a possible virus within.

There are essentially two types of existing antivirus: on request antivirus and memory-resident antivirus.

The antivirus on request must be started via a manual or automatic process and when activated they seek viruses in a specific unit or in the entire system. This search involves the RAM memory and mass storage devices such as hard disk, pen drive or optical media.

Memory-resident antiviruses are programs that run in the background. They are usually activated upon system start-up and always remain active. When a file is accessed, the antivirus of this type intercepts the call to the file and checks that there are no viruses within the file before allowing loading of the same into the memory, where it could have catastrophic results.

Each of the two antiviruses is characterised by merits and defects. The antivirus on request works, of course, on request. If it is not started before access to any file, the computer may be infected with a virus before being identified. Memory-resident antiviruses, conversely, are capable of detecting a virus before the latter infects the system but, as the process is always active, the computer inevitably decreases its performance: this represents a price to pay to ensure a good level of security. In the latter case, each file checked will decrease the speed of accessing the file system, also decreasing, as a whole, the speed of the computer. The latest antiviruses are characterised by high efficiency and reduce slightly the operating speed of the computers on which they operate, while still remaining active. However, it is possible to obtain better performance by controlling only the more likely virus signatures or by examining only those files that might be infected with greater probability such as COM-type files. Good security provides for the simultaneous use of antivirus software on request and memory-resident antiviruses.

It has already been said that antivirus manufacturers continually release virus signature updates to ensure maximum protection by the antivirus program. The updating of signature files can constitute a considerable amount of work for system administrators, especially when large networks are involved. In this sense, when using DOS, Windows and Macintosh on our desktop, updating should be performed individually for each computer.

Most producers have tried to solve this problem using the concept of virus domain to group multiple servers and desktop computers. In this case, the network administrator can add signature files, see reports and check the scanning parameters from a single location, significantly reducing the workload required to protect a larger network environment.

A valid business solution should also provide an enhanced feature of alerting in such a manner that the network administrator is informed of all viruses that have been identified on any system on the network.

In the environment of the prevention of infections by viruses, heuristic analysers are very popular and carry out statistical analysis to determine the probability that a file contains a program code that might be a virus. A heuristic analyser, as opposed to a normal antivirus, does not compare its archive of signatures with the files to be checked but uses a classification system that allows determination of the likelihood that the code of the analysed file contains a virus. If the code of the controlled file exceeds a predetermined threshold, then the heuristic analyser will warn the user in this sense, with appropriate messaging. Most antivirus software is still equipped with a built-in feature of heuristic scanning.

The great disadvantage of heuristic analysers is that they do not require updating. In fact, since the files are catalogued according to a points system, there is no need for any signature to perform comparison, allowing the analyser to intercept new viruses with good probability. This type of analyser is very useful if it is not possible to upgrade the archives of virus signatures with periodicity.

The worst disadvantage of heuristic analysers is their tendency to give false positives. In fact, if a virus is characterised by a code that is very similar to that of the normal programs, the analyser may have some difficulty in distinguishing them. For this reason, the use of a heuristic analyser, given the high number of false positives that it can generate, may represent a loss of time for network administrators.

The antivirus, in terms of application, represents a new type of virus protection. This antivirus, instead of protecting a system from viruses, integrally protects a certain service of an organisation. For example, email represents a formidable vehicle for the spread of viruses, given its heavy use at all levels. In this sense, an antivirus in terms of email application is dedicated to scanning all file attachments and search for possible viruses. Most of these products, in addition to the SMTP traffic, are also able to check the FTP and HTTP traffic.

Products that integrate directly with the firewall are commercially available. In this manner, there is a single point of management of both security and protection from viruses.

5.16.7 Virus protection

Operating environments are usually composed of a number of servers and a number of clients, represented by desktop computers that can use different operating systems. With regard to desktops, the same, even if using different operating systems, have a coherent hardware platform and are thus vulnerable to different types of identical viruses. For this reason, the recommendations and the procedures relating to desktops should be standardised also where there are different operating systems.

A very effective recommendation is, especially in terms of cost savings, activation of the protection of boot sectors through the BIOS of systems. It is a technique that is very fast and effective to ensure that the boot sectors of all the systems that are connected to the network are appropriately protected. To make the solution more effective especially to make messages clearer to the user, it is possible to integrate this system with dedicated messaging. If users do not try to update their operating system, reports of false positives should be reduced to a minimum.

Desktop systems should usually use an antivirus on request that is configured to perform a complete check of viruses of all the local units and of the internal memory at regular intervals. Such monitoring should be performed periodically, for example every night, leaving computers appropriately running. In the case where this is not possible, these control operations could be performed during

periods of inactivity, such as during lunch breaks or on a weekly basis as part of a server logon script. It is very important that the antivirus on request controls all local files to ensure that any virus has not been inserted through a dropper or has not been hidden with an unknown file extension. A suitable antivirus should also include a heuristic scanning function and should display the results on a central location in such a way that the data are accessible to the system administrator.

Every desktop computer should activate a resident antivirus in memory upon start-up of the system to search for viruses before they can be stored on the local system file or executed in memory. To ensure maximum performance of the computer, it may be necessary to select the files checked by the memory-resident analyser.

As a regular search on request should be performed on every system, it is possible to establish the level of file verification with the memory-resident antivirus in memory with a certain degree of precision. Paying attention to those files that are more susceptible to infection, it is possible to decrease the risk of impact of such search on the performance of the system. In this case, the security settings may decrease slightly but this is the price to be paid to increase performance. The memory-resident antivirus should be able to check at least: read access to files, worms, executable files such as those with COM or EXE extension, documents with macros such as those in Word or Excel.

It is very important to check read files, and not necessarily in write because also checking this second phase is an extremely burdensome operation that slows down the computer excessively. In addition, if a virus is not a virus at the time a file is read, it is highly unlikely that the virus will manifest itself in write. It is also very important to look for worms, as they do not save any information on disk in the event of infection and could not be identified by an antivirus on request. It is also advisable to configure our resident antivirus in memory in such a way as to check files characterised by a greater likelihood of infection, represented by executable files and those able to save macro commands.

File attributes, or checksum checking, are not taken into special consideration in virus protection as these methods are not particularly effective.

A further possibility is the capacity that certain operating systems have to use files access permissions. In this case, the system administrator can reduce the chance of infection by controlling access to files but workstations do not have the standard configuration used by other desktop computers. This possibility is still not valid for macro type viruses that are hidden in document files and users must have write access to their archive folders of documents to save their files. In this case, the option introduces more difficulties than advantages in the work environment in which it is used.

In networks of a certain size, as well as with desktop computers, there are also usually servers that operate with an operating system that is different from desktop computers. In such systems, virus protection is more complex and difficult, since the same can be used as a means of propagation of the virus. Not only can certain operating systems of server machines represent a propagation means but can themselves become infected.

In any case, even for such systems, performing a complete search on request for all files every night is recommended, given that these computers are constantly running. Most antivirus products for servers are equipped with a planning utility for this activity. If nightly backups are run, the antivirus should be set in such a way as to check the system file before performing a backup: in this way, we are sure that none of the files that will be stored have been infected by a virus.

The antivirus program present in the server memory is able to check the memory itself of the server and the files that are stored on the local system file. The resident antivirus memory works in a somewhat different manner if it runs in certain server environments as in the above environments the server is not capable of running standard executable files. As the system is used for the storage of files, there is no need to check the memory of the same while the monitoring of incoming traffic is more advisable. A memory-resident antivirus for servers should check: the local memory in search of worms, viruses and Trojans; executable files in input from the network; macro documents in input from the network. In a manner similar to desktops, the implementation of minimum control of the file is recommended in order to avoid significantly degrading the performance of servers. In the event that a

virus in any case manages to enter the server, it is hoped that the antivirus on request is sufficiently powerful to locate and correct or delete it during its nocturnal activity.

In some cases, the use of products of different businesses may be recommended to protect desktop computers and to protect server computers. All of this is important to ensure maximum security in the entire system and in the network on which the system is located.

It has already been stated that the setting of access permissions to files with regard to users ensures that executable files are not subject to infections. This configuration ensures benefits on the basis of the type of storage applications on the network. If all the applications are stored on the local computer, there will be no executable files on the server protected by the setting of read-only access at the user level. If applications are activated by one or more servers, it is always possible to minimise the possibility of virus attacks by selecting the minimum level of required permits.

In the following are provided some specific suggestions for systems based on the UNIX operating system. The major problem of UNIX is the possibility of a Trojan horse entering the system aimed at acquiring authentication information. A potential attacker would replace the existing telnet server with his/her own in such a way as to have sent to a remote location all the logon information of all the users who authenticate on the system. To avoid taking such a risk, it is very important to perform health checks of files consisting of a CRC verification check to identify any changes even if the current file has the same size, same date and the same time. Telnet and FTP servers should also be checked and all the other processes that accept input connections. This type of control should be performed as part of an automated process and the results should be sent to another computer for necessary analysis. This mode is preferable to avoid the results being manipulated by the same subject that has attacked the computer under control.

Another possible risk of UNIX systems is the possibility that an attacker can introduce into the system a virus worm that would appear as a new process that is running in the system. In this case, in a manner similar to the integrity check, the process should automate the results check on a different computer and, as all the processes that are running on the system can be checked, it is possible to act immediately with appropriate countermeasures if a new suspect process appears.

With regard to file access permissions, there is a default setting on UNIX systems where only the root user can overwrite software that is being run as a server on the system and for this reason, a potential attacker should first take possession of the root account or be able to access at the root level before being able to replace any type of software on the server. This mode of access should be maintained in order to minimise any possibility of tampering with the system and for this reason normal users should not have access to sensitive files for write. Because of the inherent characteristics of UNIX, there are relatively few viruses that are able to strike while the greatest risk is represented by worm viruses and Trojan horses.

5.17 Analysis of attacks

This section illustrates the tools that can be used by a potential attacker to enter the systems of others, with hostile intent. The purpose of this section is to indicate to network administrators or in any case to simple users those behaviours that are usually adopted by attackers to find the weak points of a system, in order to try to enter the system itself, whether it be a network with more than one computer connected or a simple desktop computer.

It is assumed that the attacker is a subject external to the network and that the same does not have much information available on the system itself. In this sense, an internal attacker would avoid many of the steps as he/she would usually have available a greater amount of information and would not need to circumvent certain defence systems that an external attacker would in any case need to breach. It has already been said that most attacks are carried out by internal users and that the defences put into

practice on the outer perimeter of the network may be in vain in the case of internal attacks, for which specific defences must be provided.

If the attacker does not know anything about the system that he/she intends to attack, the same must carry out preliminary investigative work.

The first thing that the attacker can do is to perform a whois query with InterNIC that owns the database of all the domain names registered and accessible to the public. This database can in fact be consulted via the whois function. If the attacker performs a search on the name of the organisation that he/she is interested in, he/she is able to learn the relevant domain. The data that are provided are the domain name of the organisation, the location of the organisation, the administrative contact of the administration, the telephone number and fax number of the directors, the valid subnet address within the organisation.

Knowledge of the domain name of the organisation is of vital importance because it allows the attacker to obtain relevant information. Any host or user associated with the organisation of interest will in turn be associated with the domain name. This knowledge allows the attacker to have a key word to be used for the next query.

When this is done, the attacker knows the physical address of the organisation that he/she wants to hit. If the same intended to harm the network or steal vital information, he/she could attempt to get hired with a temporary work contract or arrange to be accepted as a consultant. This action would allow him/her to have a certain level of access to the network to continue in his/her work of network intrusion. At this stage, he/she may try to secretly install a service port, or back door, to also access the network from outside as there is no easier way to breach the perimeter of a network that to be invited from the inside.

Knowledge of the physical address also indicates to the attacker where to go to rummage around in the rubbish in search of excerpts of documents that may be of interest for his/her own intrusive purposes. In fact, in the rubbish, it is possible to find documents or parts of documents of any type, including valid account names, passwords or technical and financial information. In more prudent organisations, this rarely occurs but can always happen. In addition, with the introduction of selective collection that separates paper from other materials, the search for paper documents can also be a clean operation from the point of view of hygiene.

The administrative contact allows identification of a person that is responsible for maintenance of the network. In many cases, a technical contact is also indicated in addition to the administrative contact. Knowledge of the names of people also allows an attack to be attempted based on the social behaviour of sending targeted emails, with the intent of receiving more useful information to penetrate within the network such as a valid logon and password that allow the attacker to have minimum access to network resources. This minimum access may represent a good starting point to attempt more dangerous attacks with respect to the network and its organisation.

Phone numbers can also be very useful. Most of the organisations of a certain level use what is called direct inward dial (DID) that allows direct contact with the desired extension by adding to the number of the organisation the digits that allow the same extension to be reached. Because of this system, the attacker can discover all the phone numbers of the organisation. The same attacker could try to call phone numbers preceding or subsequent to those of the official contact in an attempt to reach the greatest number of members of the organisation and attempt to launch a social attack in the search for the greatest amount of useful information for the final attack. The same could use an automatic calls dialler to verify consecutive phone numbers. This automatic dialler is a simple piece of software that is controllable by a computer that calls the numbers involved. Due to this dialler, the attacker can identify the numbers to which a computer responds in which to attempt to sneak through due to valid information received through the social type attack being carried out in the meantime.

Any useful information that is provided by the whois command is an IP address. As this host is part of the objective domain, it is reasonable to assume that the subnet to which it belongs also forms part of the same domain. The attacker is not able to know if the host is inside or outside the firewall, but in

any case the same knows a valid subnet that can be used at the time of the attack or to retrieve useful information.

Because of the whois command, the attacker has a way to receive the original information. The nslookup command can then be analysed that allows DNS servers to be queried to receive useful information about hosts and IP addresses. If the attacker intends to attack the network, he/she must learn the hosts that can be used as targets and in this sense, the nslookup command is of extreme importance.

Very often organisations do not pay particular attention to the security of devices for remote access, enabling users, in some cases, to have their own modem connected to the computer that they normally use: this modem may represent a breach in the security system of the network that can be exploited by an attacker, during scanning of the company numbers, to enter the network.

When an attacker uses nslookup, he/she is informed of the current DNS and if the attacker requires more information, the same should change the default DNS server into one of the systems indicated by the whois output data.

Once this change has been made, the nslookup utility indicates one of the servers of the organisation of interest. Once this is done, all queries will be directed to this system. The first thing that the attacker can attempt to do is to implement a *zone transfer* of the data relating to the names of DNS managed hosts. This transfer allows him/her to gather all the information about hosts and IP addresses using a single command. Due to the zone transfer, the attacker can receive a valid list of all hosts registered with the DNS. It is obvious that the latest DNS systems do not respond positively to this request, unless the transfer was authenticated at the beginning. However, it must be highlighted that many network administrators do not trigger such a simple security procedure. Once this is done, the aggressor may leave the nslookup utility as he/she has succeeded in collecting a number of initial pieces of information in a relatively short time and with extreme ease.

If, on the other hand, the attacker has not succeeded in zone transfer, he/she will be forced to systematically try some common names such as email, ftp and www in order to be able to discover other subnets internal to the network of the organisation of interest. In any case, it must be emphasised that with this method the attacker will not necessarily be able to identify any valid name because he/she has still had to proceed by trial and error.

If the zone transfer is successful, the attacker has a series of addresses of valid subnets upon which he/she need to focus. Other information may also be received such as the address of the email system to which an attack may be directed at a later date, if he/she has not succeeded in attacking the main network. Other useful information is the Web server address and the address of any FTP server that could coincide with the Web server, and which could therefore be used for an attack aimed at damaging the Web pages of the organisation.

Very often, if an email relay is used that does not filter all the information on the internal network, it is possible to receive in the email a series of useful information to understand the type and organisation of the internal network.

When all the information of a general nature has been collected on the objective, the attacker can try to find out if there are other systems or services in the network of the organisation that he/she wants to attack. To do this, he/she will need to overcome the barrier of the firewall, or access to one of the internal computers connected to the network or even attempt to enter via a remote access. This activity allows the attacker to have a clear picture of the network, to prepare a map of the internal organisation of the same and to have a list of available services.

A command that is greatly used is traceroute that serves to trace the network path from one host to another. It is very useful for documenting the network segments between two hosts. In Windows, this command is known as tracert because of the limit of eight typical characters of that environment. If this command is executed within the perimeter of the network, information on all the IP subnets and internal routers that connect them is obtained. If the attacker chooses more significant hosts, the same is able to have a complete diagram of the network.

Another operation that can be performed is the search for hosts and services to find out which systems are active in the network and which ports are open on each system. It represents the next step that allows for identification of the systems that might be vulnerable to attack. In this sense, the attacker must find every system on the network, every service running on each system and the weak points of each service. These operations may be performed one at a time or through the use of programs that allow them to be performed all together.

In this sense, an operation that can be performed is the scanner ping that sends a simple ICMP request to each IP address in sequence on a subnet and waits for a response. If the command receives a response, it deduces that there is an active host at the address queried. Once a response has been received, the command stores in its log all the hosts that have responded and tries to translate the IP address into a host name. To perform a simple ping scanner, all that is needed is a script or a batch. However, there are also some simple types of software that perform these functions.

Another operation that can be performed is port scanners that allow checking, in sequence, of a number of ports on a target system to check whether there are listening services. A port scanner identifies simply which known services are listening and awaiting a connection request.

Another operation that can be performed is TCP half scanning that was developed to circumvent the problem of inclusion in the log. This operation does not attempt to establish a full TCP connection but transmits only the initial SYN=1 packet: if the target system responds with SYN=1 and ACK=1 then the half scanner learns that the port is listening and immediately transmits a RTD=1 to close the connection. As a full connection has not been established, most of the systems do not record this activity. Even the TCP half scan, which bases its operation on the initial SYN=1 packet, can be blocked by a packet filter or by a firewall in a similar manner to the complete scanner.

Another type of scanner is the FIN scanner that does not transmit a SYN packet=1, trying to establish a connection, but a packet with ACK=1 and FIN=1, so as to convince the target system that we want to terminate a connection even if said connection has never been established. Its great danger is that neither the static filtering of packets nor firewalls are able to block it, allowing the attacker to identify systems behind a firewall. On the other hand, the target system, if the port in question does not have a listening service, responds with ACK=1 and RTD=1. On the contrary, if there is a listening service, the target system will ignore the request and this is due to the fact that there is no connection that the system should terminate. Depending on the ports that have responded, the FIN scanner can deduce which ports are active for a given system. FIN scanner search is not able to run on every system. For example, a Microsoft TCP stack would respond with an ACK=1 and RTD=1, even if the port was not active. This means that the Microsoft TCP stack does not conform to RFC 973, and it is therefore not possible to use a FIN to identify the ports of a Windows system as, given the type of response, it is as though no port in the system were characterised by active services. The response characterised by ACK=1 and RTD=1 indicates, however, to the attacker that there is a system with a Windows operating system and this can also represent useful information for a future attack.

Another operation that can be performed is traffic monitoring in order to acquire more information. This can be done in a direct manner by installing a network analyser, provided that we have access to the network itself or, alternatively, through identification of internal systems. It has already been shown above what can be learned using a network analyser.

For example, if the attacker somehow manages to get internal users to connect to his/her website, he/she manages to obtain important information including: the operating system used, the processor used and the browser used. Once this information has been obtained, the attacker learns more details about the machine of the network that he/she intends to attack. One of the greatest advantages offered by proxy machines is that these machines eliminate the information relating to the operating system and to the browser, making acquisition of the data by the attacker more difficult and complex.

If the aggressor performs most of the operations shown so far, he/she is able to have a clear idea of all the systems that are connected to the network and the services that are running on each of them. At

this point, he/she can start the search for vulnerabilities that may be exploited to attack the network. Such search can take place by successive attempts, checking from time to time the result obtained.

A particularly aggressive attacker is immediately directed towards the search for vulnerability without proceeding randomly. In this way, he/she will not risk activating a possible network alarm by random processes. Control of the vulnerabilities can be done manually or automatically via software.

With regard to the manual control of the vulnerability, it takes place by means of an instrument, such as telnet, in order to connect to a service and check who is on the other side. Most services can identify when a remote host requires a connection. This operation is performed to locate and correct problems but may also provide a great deal of valuable information to an internal attacker. Table 5.5 lists a series of commands that can be used when connecting to a service port using telnet.

It is obvious that manual checking requires a certain amount of work and time because verification of target services must be performed manually. In addition, this type of checking requires the attacker to have some experience in this respect since knowing what kind of service is running on the target system may lead to nothing if the attacker is unable to use this information.

With regard to automatic checking of vulnerabilities, this can be performed by means of suitable software that performs all the tasks of survey and search that an attacker would have to perform manually, with large expenditure of time and energy. This software is also called automatic scanner of vulnerabilities. These scanners can be directed towards a single system or entire IP subnets. The first thing that the scanner does is to identify potential targets. Subsequently, it analyses the ports controlling all the active ports to search for vulnerabilities that will be reported to the user of the scanner. Depending upon the type of scanner, the same can also be provided by utilities capable of exploiting the loopholes that are identified. Vulnerability scanners simply automate the manual process of identification of potential vulnerabilities. A very practical attacker is however able to ferret out hidden flaws in the network with the highest probability compared to an automatic system such as a scanner that is limited to performing security checks at the same skill level of the person that programmed it. It is virtually impossible for a remote vulnerability scanner to find all the flaws of a network or system without working against the system itself. For example, it is impossible to know if a system is resistant to attacks of any kind without launching the attacks themselves to see if the same is capable of coping with them. This means that the security of the system itself is not guaranteed only by the fact that a vulnerability scanner has found no fault. A scanner is able to identify flaws without having to exploit them for this if, and only if, the program is run on the system to be checked or if the same has full access to the file system. A program that runs locally is characterised by the ability to be able to compare the application and driver dates with a list of known corrections, in such a way as to be certain that a certain update of a driver was performed at a date subsequent to a specific type of attack and for this reason, this driver is not sensitive to that type of attack. It is worth remembering that

Table 5.5 Commands that can be used during a connection to a service port using telnet

Service	Port	Commands	Comments
FTP	21	User, pass, stat, quit	Only one session command can be given and it is not possible to transfer files.
SMTP	25	Helo, mail from:, rcpt to:, data, quit	An email message can be forged using these commands.
http	80	Get	We receive an error page but in this way we know that the service is active.
POP3	110	User, pass, stat, lost, retr, quit	Mail may be read by connecting to the POP3 port.
IMAP4	143	Login, capability, examine, expunge, logout	All commands must be preceded by a unique line identifier.

vulnerability scanners are instruments and not solutions to the problems and must be considered and used as such. They are very useful for providing initial information but are not able to tell which of all the systems connected to a network are secure and which are not secure. In this sense, there is no tool able to replace the activities of a network administrator and system expert, which is always up to date on all security issues and has an excellent knowledge of the systems that it manages and controls.

5.17.1 Execution of the attack

Once the attacker has identified the vulnerabilities, he/she is ready to start the attack whose type depends on the objectives that the attacker has. He/she may be interested in a certain resource or all the network systems or be aiming to breach a system or intending to suspend all the services. The type of attack will develop on the basis of his/her objectives.

The following will show a common set of system weaknesses and related attacks that can be launched against them, taking into consideration that it is a continuous leapfrogging between new vulnerabilities, new types of attacks and new measures of defence.

A big problem is hidden accounts that can entirely circumvent a system's security policy. For example, this can occur with network devices that contain a hidden account administrator-level password with non-editable password and it can occur with certain device manufacturers, opening a huge flaw in the security of the network and of the systems that use it. The solution to leave a back door in the device is not an oversight by the manufacturer, but is a clear choice to allow technical staff to be able to enter a device remotely in which the administrator has forgotten the access password. This solution, despite all the good intentions on the part of the manufacturer, poses significant problems from the point of view of security. To avoid this problem (forgetting of the administration password), certain manufacturers require removal of the device from the network and physical access to the same for password recovery: this solution ensures a greater level of security. Hidden accounts constitute a significant problem to which the utmost attention must be given. In this sense, unless the manufacturer of a certain device attaches a declaration with regard to this matter, it is not possible to know if the same is equipped with an administrative hidden account. This means that it is not possible to rely exclusively on password authentication, but that it is necessary to put in place additional measures such as full disabling of remote management or limitation for management with well-determined IP addresses.

Another type of attack is *man in the middle* already discussed in Chapter 2. In the case of networks, this attack consists of having an attacker positioned between the client and the server, and equipped with a packet analyser. There are many variations for this type of attack and practically all of them use the fact that the vast majority of network communications does not pay particular attention with regard to authentication: if the two communicating parties do not verify in continuation the identity of the subject that is located on the other side, the communication itself could be diverted to another hostile subject without the user that is located on the other side knowing about it, exposing serious security problems. When entering a communication in progress, this is referred to as session diversion or *hijacking*. In this sense, an attacker waits to establish a communication session between two subjects, filtering into that communication, giving the other person the impression that they are dealing with the subject with which this communication was begun. In this sense, not so very long ago, there were tools that allowed users to hijack the communication session of the supervisor and to pass their identification logon as equivalent to that of the supervisor.

When a programmer creates an application, he/she must provide for memory spaces, or buffers, suitable to receive the input data from users or from other applications. For example, a login application must ensure a certain amount of space in memory to allow the user to enter a name and a logon password. To allocate enough memory, the programmer must predict the length of the strings of input. It is of course possible to impose limitations in input that are not to be exceeded and to

introduce mechanisms for the management of input data longer than expected in order to avoid overloads of memory or buffer overflow. There are many types of attacks that can be conducted by exploiting memory overflow and the most known have been:

1. sending of packets of oversized ICMP requests;
2. sending to an IIS 3.0 server of an URL request of 4,048 bytes;
3. sending of email messages with attachments whose file name is 256 characters long to email clients;
4. sending of an Server Message Block (SMB) logon request to an NT server with the size of the data identified incorrectly;
5. sending to a Pine user of an email message with the sender's address longer than 256 characters;
6. connection to a POP3 Wingate port with subsequent input of a user name with 256 characters.

From the types of attacks referred to above, it is easy to deduce that the problems of buffer overflow considerably affect the behaviour of the server and the relative security. The only way to know if an application is not secure from the point of view of the buffer overflow is by the examination and inspection of the source code. It is always possible to launch an attack as the fact of not being able to produce a buffer overflow does not mean that a system is immune.

Another type of attack is the SYN attack that exploits the use of a small buffer during the TCP *three-packet handshake* to prevent a server from accepting TCP connections in input. When the server receives the first SYN=1 packet, it stores the connection request in the queue for processing. As the sessions are usually established very quickly, this queue is characterised by a small buffer and is capable of storing a small number of connection requests. A SYN attack attempts to overload this small queue via a large number of connection requests. When the recipient system responds, the attacker system does not respond resulting in the request remaining in the processing queue until expiry of a predetermined time interval. If this queue is filled with a high number of false connection requests and there is no response to the same when they are made, the attacker system overloads the input queue, preventing the correct reception of legitimate connection requests and blocking at point of receipt the system that is under attack. For this reason, a SYN attack is considered a DoS attack. As the use of two memory spaces is a standard feature of TCP, there is no other way to solve the problem. The options available include the increasing of the size of the queue being processed and reduction of the time of persistence of those connection requests that do not reach their final goal and elimination of them as soon as possible from the processing queue. The increase in the capacity of the queue is an increase in allocated memory in order to be able to store a greater number of requests. In any case, it must be considered that a considerable amount of space may be required to avoid very fast networks being exposed to SYN attacks while for systems characterised by slower networks this need may not exist. If the time within which the connection needs to take place is excessively reduced, there is a risk of excluding very busy systems or systems characterised by a slow connection.

The search for a correct balance of a system in order to prevent SYN attacks is not a simple operation. It is necessary to increase the memory capacity of the queue to be able to manage a number of connections at the same time while avoiding dimensions that are so high that they constitute a waste of memory. In addition, it is necessary to precisely adjust a connection timeout in such a way as to be able to eliminate irrelevant aspects but at the same time to ensure successful connection to authorised systems. Unfortunately, the majority of operating systems do not allow such a high adjustment and the manufacturer's default values must be accepted hoping that they are suitable to meet the needs of those that use them.

Another type of attack is teardrop that has already been explained and that will be illustrated again in the following. This attack uses the fragmentation offset field and the length field within IP packets. The offset field is usually employed by the router: if a router receives a packet that is too large for the next segment, it must break it up before transmitting it. The fragmentation offset field is used in conjunction with the length field in such a manner that the receiving system can reassemble the datagram in the right order. When a device receives an offset equal to 0, it contends that this is the first

packet of fragmented information or that fragmentation has not been used. If fragmentation has been used, the receiving device will use the offset to understand the exact placement of the data within each packet when the same reassemblies the datagram. In practice, the IP fragmentation offset indicates to the receiving device at what distance from the beginning of the datagram the received data must be placed: if everything works correctly, this system allows reassembly of the datagram of the correct order. In this sense, the length field is used to make a final check to ensure that there are no overlaps and that the data has not been damaged in transmission. In the event of error in reassembly, the receiving device sends a request for new data transmission. Most IP stacks are capable of managing overlaps or blocks of data too large for the corresponding segment.

A teardrop attack begins by sending a normal data packet of normal size and with fragmentation offset equal to 0. Checking the initial data packet, this type of attack does not appear to be distinguishable from an ordinary data transfer. Subsequent packets, on the other hand, are characterised by offset fields and altered length. The receiving device will not be able to reassemble the packets due to the false information of reassembly and will crash. When the second data packet is received, the fragmentation offset is checked to understand at what point of the datagram the information should be placed. In this type of attack, the offset of the second packet requests that its information is placed in the first fragment. By checking the field relative to the block of useful data, the receiving device realises that these data do not have dimensions such as to be able to extend beyond the end of the first fragment: in practice, the second fragment does not overlap with the first but is completely contained within the first. As this represents an unforeseen error condition, there is no routine able to manage this condition and a memory overflow will be generated that will cause a crash in the device. In many devices only one altered packet is needed to cause the block, while many others need more packets.

Another type of attack is the so-called *Smurf* that uses a combination of *spoofing* of IP and ICMP replicas to saturate a host causing a DoS attack. The attack is divided in the following manner: the attacker sends a Ping spoofed packet (echo request) to the broadcast address of a network with a large number of hosts and an Internet connection characterised by a high bandwidth. This network is called rebound network. The false Ping packet presents the address of the system that the attacker intends to attack as the source address. When a router receives a packet sent to a broadcast IP address, it recognises that it is a network broadcast and maps the address to the FF:FF:FF:FF:FF:FF Ethernet broadcast. When the router receives this packet from the Internet, it will transmit to all hosts on the local segment. As a result, all hosts that are connected on this segment respond with an echo reply to the false IP address. If the Ethernet segment local is large, there may be hundreds of hosts that respond to each echo request they receive. Since most systems try to manage ICMP traffic as quickly as possible, the system being attacked is rapidly saturated by the echo reply, becoming unable to handle the traffic and generating a DoS effect. This mode of attack is aimed not only at the system that is being attacked but also at the Internet connection of the entire organisation to which the target system belongs. In fact, if the rebound site is characterised by a high-speed connection while the organisation of the target site is characterised by a slow connection, all the inbound and outbound communications will be blocked. There are several methods for preventing a Smurf-type attack. They are block at source, block on the rebound site and block on the target site.

With regard to block at origin, it should be remembered that this type of attack is based on the ability of the attacker to transmit an echo request with a counterfeit source address. This attack can be blocked at source by using a router access list that ensures that all traffic that originates from our network should be characterised by a well-defined source address. This operating mode prevents the counterfeit packet being able to reach the rebound site.

With regard to block on the rebound site, there are two possibilities. The first possibility is to simply block all incoming echo requests, preventing such packets from reaching the network. If this is not possible, then the routers must be prevented from mapping the traffic directed to the network

broadcast addresses to the local area network (LAN) broadcast address. If this mapping is avoided, the systems will not be able to receive echo requests.

With regards to blocking on the target site, very little can be done to avoid the effects of such an attack on a wide area network (WAN) connection. This traffic of attack can be stopped on the perimeter of the network but in this case, it is too late to prevent the attack taking possession of the entire bandwidth of the connection. It is in any case possible to minimise the effects of this attack by at least stopping it on the perimeter. If we are using dynamic filtering of packets or any firewall that maintains the status, it is possible to prevent such packets from entering the network. As the state table is able to know that the attack did not originate from the local network, as the same would not have an entry in the table that shows the original echo request, such an attack would be managed as any other spoofing attack and would be immediately blocked.

Another type of attack is brute-force attack that simply represents an attempt to try out all the possible values when attempting to authenticate with a system or to breach the hidden key used to generate a ciphertext. For example, an attacker could try to access a server at administrator level using a set of words in a dictionary as possible passwords. In this sense, no particular skill is required as the attacker is limited to trying all the possible words in the search for the password. The system most often used to carry out a brute-force attack is the use of a program called *password cracker*. The system administrator is unable to control the use of these programs that are available for each platform. It is of course possible to prevent an attacker from protecting files that include information about passwords and to prevent network sniffing through the use of routers and switches.

In addition to attacks via the network, it must be remembered that there may always be physical attacks via direct access to one or more computers connected to the network. This mode requires a physical presence within the organisation to be attacked. In this respect, the systems that are kept in isolated areas or in areas with restricted access are more vulnerable because the attacker has available a greater confidentiality to operate on them. It has already been said several times that most attacks are from within an organisation and this allows an attacker a greater level of access to network resources. Once there is physical access to a system, an expert attacker will not have great difficulty in obtaining an administrator level of access, having the opportunity, at this point, to operate any activity on the system itself. Once an attacker has physical access to a computer, the same can perform a series of operations such as:

1. opening the computer and disconnecting the battery to reset the password in a Complimentary Metal Oxide Semiconductor (CMOS) memory;
2. execution of the system boot from an external hard drive to access the local system file;
3. elimination of the password of the local administrator to gain full access to the local operating system;
4. rebooting of the system with the disconnected network card in order to be able to log on locally as administrator without triggering any alarm;
5. varying the level of logging in such a manner that suspicious activity is not stored;
6. installation of sniffer software to check the network communications;
7. use of a breached password to attack other systems on the network and many other activities.

In practice, a well-prepared attacker who has physical access to a system can completely bypass the security of an operational environment in a short space of time and most of the time by waiting for switch on/switch off times of the system. If we manage the security of a large environment, it is essentially not possible to fully protect client computers, because if it is possible to have access to the same, it is possible to breach them and to breach the entire system on which the network sits with great ease if we possess the necessary technical expertise.

5.18 Prevention of attacks

Since modern software is generally very complex, there may always be flaws that make, and will make for a very long time, network security vulnerable. It is often guaranteed publicly that current software is code free that can be used for hostile actions but this fact does not ensure that future software will be so too. On the other hand, memory overflow has been an issue for programmers since the 1970s and still nowadays this problem persists. To maintain a secure environment, it is necessary to know the weaknesses of the software when the same are discovered and to intervene promptly, without waiting for upgrades of a product or *a service pack* relating to security. To avoid this problem, some of the big software manufacturers release so-called *hotfix* relating to continuous security in order to avoid leaving open gaps when the same are discovered. In this sense, it is very important to install the same to ensure immediate protection while waiting for the manufacturer to release a patch.

The official channels of manufacturers are the main point of reference to find more updated security patches. Very often the most up-to-date information and details on the security issues of certain products can be found more on third-party sites that, unlike manufacturers, do not have any constraints or direct interests of marketing.

There are many third-party resources to which we can refer to stay up to date on the latest security vulnerabilities. These resources are usually made available by willing subjects and by organisations that specialise in network security. These resources are usually free and no economic commitment is required to access related information. Many of them still use harmless advertising to support the management costs of the service. Resources of third-party security also include databases of vulnerabilities, websites, *mailing lists* and *newsgroups*. Each of these resources is characterised by advantages and disadvantages that are as follows:

1. Databases of vulnerabilities allow vulnerability searches but do not provide any information on other matters.
2. Websites can be characterised by direct links to information on *patches*, sometimes with detailed descriptions, but it can be difficult to find a well-determined vulnerability.
3. *Mailing lists* immediately report the new vulnerabilities as soon as they are discovered but some of them, when we sign up and authorise the sending of email, can in some cases send several tens of mails a day, clogging our inbox.
4. *Newsgroups* are able to offer more in-depth discussions on particular vulnerabilities but we need to read all the messages on them to be able to find what we need.

The security of a network environment can be a very challenging activity particularly when we need to check more than one server. However, the management of daily interventions is also a very difficult task and providing security management can become a considerable task. If the network administrator is overwhelmed by daily chores, he/she may neglect certain basic tasks such as security. The biggest problem is not knowing what to look for, as most network administrators may change settings or load necessary patches but would gladly make reference to a document or to a guide to follow to ensure the highest level of security. The most likely choice, in such cases, is to refer to a security consultant but this can be a costly solution and is not available to all organisations.

If we need to locate any security flaws, a good starting point is vulnerability scanners. In many cases, all that is needed is guidance on how to create a more secure and more protected network environment and in this case we can use programs for the control of security. This type of program is not directed at the search for flaws or known vulnerabilities but allows us to check that all network systems follow the security policy of an organisation.

5.19 Disaster prevention and recovery

By protection of disasters is meant the set of precautionary operations that are implemented to ensure that any damage to the resources of an organisation cannot impact on the performance of daily activities. Prevention of disasters is similar to insurance: we invest money for a future need that it is hoped will never happen.

Restoration after a disaster is closely linked to prevention. Despite all the precautions that can be taken in order to prevent the occurrence of catastrophic situations, it is very important to develop a recovery plan and to know what behaviour to adopt in the case where these situations occur. This activity represents the boundary between recovery and the final catastrophic situation.

Risk analysis and the identification of critical resources are of vital importance for the prevention of disasters. It is also very important to attribute an economic value to the impossibility of using resources in such a manner as to understand how long a possible downtime can be tolerated. In the following section, we will address the options available to maintain resources that are accessible and how to deal with any threat for the security of our network.

5.19.1 Division of disasters

Possible solutions for disasters can belong to two different categories:

1. maintenance or restoration of service;
2. protection or recovery of lost, damaged or deleted information.

Each of these categories has its own specific function in the protection of assets and it can be safely said that there is no complete solution to a disaster if there are no such solutions available, partially or in fully.

For example, suppose we have a server with two hard discs with reciprocal active mirroring. This functionality ensures that both discs always contain the same information. When using *mirroring*, the possible malfunction of a disk drive does not cause blocking of the server as the remaining disk unit can continue to guarantee space to store and read data. Mirroring is considered a service function for recovery from a disaster as it ensures the availability of files.

If a user wishes to read a file deleted a long time before and if *mirroring* is the only recovery procedure for the recovery provided on the server, then the situation is problematic in that the files are both written and erased from both the discs and, therefore, there is no system to recover the lost information: for this reason, *mirroring* is not an effective solution for restoration.

In this sense, we can definitely say that a solution for complete recovery after a disaster must make it possible to find a way both to remedy the faults of the services and to obtain the information that has been lost. Both aspects are critical for a contingency plan aimed at addressing any disaster situation.

5.19.2 Network disasters

Even if network disasters are able to block the communications of an entire organisation, they are, in most cases, not taken into account. Most organisations are more interested in the fact that a given server is always available and for this reason less importance is given to the network that represents the key element for access to the server: if a network is not functioning properly, the server becomes unusable.

The following will address the main networking technologies and their vulnerabilities that may cause a partial or total loss of network functionality. A thorough understanding of strengths and weaknesses of the characteristics of the different network topologies is an indispensable aid to the solution of any problems, in particular, when in the presence of faults that are not immediately visible.

5.19.2.1 Media

Quality procedures for disaster recovery began with network media. Even if physical network cables remain the main support for most LAN networks, increasingly widespread appears to be the use of wireless networks that must be taken into account as an integral part of a potential recovery from a disaster. When using cable, the wiring chosen allows us to choose the strength of the network in case of failure. Regardless of the chosen media, the media itself will have to support all the network communications, and for this reason a fault that occurs at this level can have catastrophic effects.

Thinnet and Thicknet cabling refer to the original Ethernet specifications of the 1970s. Both specifications allow connection to multiple systems on the same logic stretch of cable. This allows the existence of a central point of failure because if any part of such a cable becomes defective, all systems that are connected to it are not able to communicate. The percentage of use of such wiring in new networks appears to be in continuous reduction and is, in any case, at a very low level. One of the major improvements that can be made to the availability of a network consists of the replacement of these wiring systems with very modern solutions.

Category 5 wiring (CAT5) represents a relatively modern standard for most network installations. It is, in any case, in the process of being replaced due to the use of optical fibres, given the continued increase in demand for bandwidth. It is however still used in hybrid connections such as in Gigabit Ethernet that, even if based on optical fibres, for relatively short lengths of the order of 50 to 75 m, allows the use of CAT5 cabling.

Unfortunately, category 3 wiring (CAT3) is still in use that allows operations up to 10 MB. CAT5 does not guarantee 100 MB or faster actions. It merely ensures the ability to support these speeds. The components of CAT5 wiring must be assembled and tested so that everything operates in the best way possible. It is possible that CAT3 or CAT5 wiring installed incorrectly can allows connection and communication between 100 MB devices. In this case, the problems do not occur until the network is subjected to a very heavy work load. Problems occur, in this case, with slower performance, frequent retransmission of packets as a result of errors or disconnection of users from services. The best method to avoid the typical problems related to copper twisted pair wiring is to check and certify the cables before using them. If this is not possible, or we must use cables with low quality, it is preferable to segment areas with major problems with one or more switches. A switch has the ability to intercept packet errors and to isolate transmissions in multiple collision domains. This does not allow the correction of errors but limits the effect that the problems of cable can produce upon the remaining part of the network.

Another type of wiring that is greatly used is optical fibres that use electromagnetic waves at optical frequencies to transmit information over the network and for this reason are immune to electromagnetic interference (EMI). EMI can induce the transmission disturbances that become particularly evident if the wiring is subject to a high load. In this sense, optical fibres are immune from electromagnetic disturbances, thus increasing the availability of the services that use this type of connection.

Each logical topology is characterised by appropriate specifications that indicate the maximum cable length that can be used. For example, 10 and 100 MB stipulate that the maximum length of the twisted pair connection does not exceed 100 m. These specifications ensure that a system that is at one end of the cable is capable of correctly receiving information that is sent from a system that is located at the other end. If the specifications are not complied with, it is possible to check the intermittent failures due to poor signal level, slowing communications along the entire stretch of affected network due to the increase in collisions. Since these issues are intermittent, they are extremely difficult to resolve. Using a network analyser, it is immediately possible to find out if we have exceeded the cable length limits for the logical topology chosen.

Very popular at the moment is the use of wireless networks. Because of new technologies and new standards, it is possible to have high-speed wireless LANs (WLANs). A WLAN is a transmission

system that is independent of the location, allowing access to the network using radio waves instead of via cables. WLANS are used as final connection between a set of client computers and a wired network. A WLAN is however vulnerable to interference and to installation and configuration. It should be remembered that wireless technologies must fall within the disaster recovery plan of any organisation and can be used as emergency networks in the event that the fixed infrastructure was out of use. In addition, as WLANS are gradually added to an existing wired network, the same cables can be used as reserve cables in the case where the WLAN was out of use.

5.19.2.2 Topology

The topology that is used is a decisive influence on the resistance of the network in the event of a failure as certain topologies behave in a manner that is more efficient compared to others in relation to restoration. It is not always possible to change topology. The purpose of this section is to explain the most common problems of the various topologies.

A ubiquitous technology is Ethernet. As cable is used, this connection mode may prove to be extremely resistant to failure due to wiring problems on every single section of the network. This situation allows us to isolate the problems since only one system is compromised. If such a system is a server, the connection may affect multiple users. The main problem with this technology is that a single system can take possession of the entire network, blocking it in the event of a malfunction. This possibility is very remote with new network cards but was not such a remote possibility with the old network cards that were subject to the so-called jabbering in the event of malfunction. In this situation, the card caused a continuous transmission of network traffic, blocking all the other systems that were waiting, for transmission, for the end of communications by the defective network card. As the network card continued to transmit as long as the same was supplied, the network was completely blocked. This problem is not an issue in new network cards and the use of switches, thanks to sectioning of the network, has virtually reduced this problem to zero. In fact, if a switch is in the presence of a defective network card, since its packets are not in accordance with the standard, the switch itself, through error control, realises it and does not forward them on to the rest of the network, isolating the problem.

Another fairly widespread technology is Token Ring that has been designed to tolerate failures but that is not without defects. It behaves in a manner that is very efficient when all systems are working as expected. In this sense, a network card connected to Token Ring performs a self-test if the other network cards report a possible malfunction. It is clear that a faulty card may not be able to carry out a proper self-test and the card would revert to transmitting on the network generating significant problems for the rest of the network. A possible error that Token Ring can make lies in the identification of a network card is in its reprogramming with an incorrect network speed. As this type provides that each card passes a token to the next card, a single card programmed at the wrong speed is likely to block the entire network. An Ethernet network does not have the same problem as if a single card operates at an incorrect speed, it would itself not be able to communicate while the rest of the network is able to operate properly. There could even be the case whereby two network cards are characterised by the same media access control (MAC). In Ethernet, a duplicate MAC only affects systems characterised by the same code while in Token Ring this possibility can cause a block of the entire ring. This is due to the fact that each system must register the MAC address of its neighbour upstream and its neighbour downstream: the presence of a duplication can completely confuse the ring systems during the process of identifying these other systems present on the ring.

Another very used topology is the Fibre Distributed Data Interface (FDDI) that is a ring topology into which was added a second ring to avoid the problems that were encountered with Token Ring. This second ring is in a state of rest until an error condition occurs. When this condition occurs, the FDDI systems are able to work together to isolate the problem area. FDDI is considered as a

technology in extinction because no effort was made to take its speed over 100 MB. In any case, it is a very reliable technology due to its ability to tolerate failures.

In a FDDI, each station is connected to both rings to ensure protection against the failure of the wiring and hardware. In the event of a malfunction in a ring, the system is able to realise this, and to activate the second ring, alerting suitably all the systems to use the second ring until restoration of the first. Another topology that is very much used is the WLAN 802.11b that defines two primary subjects as follows:

1. The station that is represented by a computer equipped with a network card without cables.
2. AP that operates as a bridge between the permanent network and computers that are connected by means of radio waves. It consists of a radio, a wired interface (Ethernet) and a bridging software. It is the basis of connection of mobile computers.

This topology works in two modes:

1. Infrastructure, also called basic service set (BSS), that represents a wireless network with at least one AP connected to a wired network and one to a group of one or more wireless stations. Two or more BSS in a single subnet represent an extended service set (ESS). This mode is the most widely used solution in large organisations.
2. Ad hoc, also called independent BSS (IBSS) or peer-to-peer. In this case, there is only one group of wireless stations that communicate with one another without the AP bridging services.

In a similar manner to Ethernet (802.3), 802.11b forces the sender to listen to the media before transmitting. In Ethernet, the complete access protocol is called carrier sense multiple access with collision detection (CSMA/CD). In a wireless network, it is not possible to find a collision, as a station will not be able to transmit and listen at the same time and so cannot identify a possible collision. To avoid this, 802.11b uses a technique called CSMA with collision avoidance (CSMA/CA) that operates in the following manner: the sender listens and if the same does not hear any activity, it still awaits for a certain random period of time and then transmits. If the packet is received intact and unchanged, the receiver sends confirmation to the sender that closes the transmission. If the sender does not receive any confirmation, it deduces that there is a collision and the packet is sent again. Unfortunately, CSMA/CA generates a further work load and this makes a network based on 802.11b slower than an Ethernet network of equivalent speed.

Another potential problem is the hidden node, which has already been widely discussed previously. Designing a WLAN with multiple APs avoids the issue of a single point of failure. Even if the re-association with a new AP takes place when a station is physically moved from its original AP, it can also be the case that it happens where radio characteristics occur or where there was high network traffic, ensuring a balance of the network.

Private circuit WAN topologies such as dedicated lines or T1 connections ensure a good level of confidentiality but introduce a single point of failure. A dedicated line or a T1 circuit is equivalent to a single cable stretched between two geographically separated sites. If any part of this circuit is interrupted, there is no built-in redundancy to ensure that the information can be exchanged between the two sites. If we are using a private circuit, an analogue redundancy option can be taken into consideration. This operating mode is very important, given the current trend of moving servers from offices and concentrating them in a centralised location. This also allows us to have a single management point and a single point of failure. If a secondary server-less site loses its connection to a private circuit, it will remain free of network resources.

Another technology that is very much used is frame relay that makes WAN-type connectivity through a shared public network possible. This network is packet switched that means that if any one stretch of the frame relay fails, the traffic could be diverted onto other operational connections. This would inevitably cause traffic congestion but a certain level of connectivity could be maintained. However, it is impossible for the entire frame relay network to fall. Frame relay provides improved fault

tolerance with respect to private circuits but is not immune from failure. If we have a stretch of WAN that operates on a frame relay that must be operational again, it is necessary to provide some form of circuit redundancy.

Another technology is the digital subscriber line (DSL) that uses existing copper telephone lines and requires short distances (less than 5 km) to reach a decentralised switching site. DSL is oriented to the circuit but instead of using physical circuitry for the entire length of the connection simply requires a complete circuit. This fact reduces the number of points of failure in any condition. DSL can ensure a high speed that can reach 32 Mbps of download and up to 1 Mbps upload. This speed is unfortunately not fixed and this fact can be a problem for organisations that require a constant flow for video conferencing and other multimedia use.

5.19.2.3 Single points of failure

The best way to minimise network disasters lies in the identification of single points of failure and in the introduction of appropriate redundancy or in the development of a contingency plan. The inadvertent creation of a single point of failure is the most widespread error that is committed when designing a network. Take into consideration, for example, the elements that make up a typical connection to the Internet are single firewall, single router, single Channel Service Unit/Data Service Unit (CSU/DSU), single dedicated line or T1 connection. This configuration is composed of three electronic devices and of a network circuit that is not under the control of the administrator. These elements, if malfunctioning, are able to break the connection to the Internet. With regard to electronic devices, they do not provide easy and immediate replacement unless such devices have been purchased in duplicate and one of them is ready in stock for replacement. The WAN circuit is controlled by the supplier of local connectivity and the relevant response time of the service provider depends on the type of contract signed with the same.

Such factors may represent significant problems for organisations that use the Internet for their daily operations. Usually, the Internet connection becomes an essential service for many business functions after a certain time from its installation and, at least at the beginning, the possible loss of connectivity represents a non-essential element that becomes critical as the main services of the organisation begin to organise themselves around it. For this reason, it is very important to review risk analysis in order to identify the single points of failure of the network. It is also very important to evaluate the effect that the loss of service may have on the organisation in consideration and if the critical points of the network have been identified, it is necessary to intervene with appropriate redundancies.

At the beginning of the 1990s the most popular devices were represented by hubs that allowed a large number of ports with a single point of management. The hub could be characterised by dozens and dozens of ports and its users that, if the device were to fail, completely lost their connectivity. For this reason, the organisation resorted to more stacked hubs, each connected to a relatively small number of users, in such a manner that malfunctioning of one of them would cause an impact on service restricted only to connected users. Switches were then introduced that are characterised by higher performance and greater reliability because they have a single point of management and greatly reduced management costs.

Stackable solutions allow for greater flexibility in recovery after a failure: in the case of a single unit failure, all that needs to be done is to remove the same from the stack and to introduce there a new one without loss of connection to other users connected to the other units.

We have seen previously that dynamic routing can be used to exploit multiple paths between the various network sections. Many routing protocols perform evaluations of this type when they must decide on the best route on which to forward the traffic. Static routes are the best choice when there is only one path or in areas where there is the risk that an attacker is able to damage the routing table, such as in an Internet connection. In any case, in the internal network, the use of a dynamic routing

protocol is recommended. Where there is a single point of connection in each of the routed segments, an option might be the use of a second router, to ensure redundancy, or the addition of other network cards on servers. When using the hop count and the associated cost, it is possible to configure the network in such a way that it does not pass through the server if there is an emergency. This ensures that the server does not need to support a further load unless the primary router malfunctions.

WAN connections always represent a single point of failure. Most organisations do not provide any type of redundancy, in this sense, due to the high costs required for the latter. This prevents any type of reliability control on this sector of the network, the latter depending on the service provider. A possible solution is to configure the border routers in such a way as to fall back on an emergency circuit in the event that the primary line was malfunctioning. This reserve element may be an analogue line or by a pair of modems even if the available bandwidth is drastically reduced if a T1 line is being used but where it is still possible to provide a connectivity of lower level, a solution preferable to not providing any connectivity. The configuration of a router, in order to ensure a reserve circuit, is not a particularly complex process.

5.19.2.4 Saving the configuration

The network disasters considered up to this point are related to the availability of services. It should be remembered that no recovery from a disaster can be considered complete if we are unable even to restore lost information, with reference, of course, not to the data that are in transit on the network. Protocols play an optimal role in ensuring that the above information will not be lost and the real problem is the possible loss of the configuration files that are used to program the various network devices such as routers, switches and hubs. When a network device fails, it is possible that its programmed configuration is also lost. It could even happen that someone unintentionally may vary the configuration making the device no longer useful for its purposes. In such a situation, a copy of the configuration files must be available in such a way to be able to rapidly restore the original configuration.

The best way to be able to save the configuration information is *terminal logging*, since most terminal emulation programs are equipped with a system that records all the information that passes through the screen of the terminal itself. If the network device has a command to show all the configuration information, *terminal logging* can be used to store such information in such a manner as to be able to resume at a later time, if necessary.

There are certain devices, such as routers and switches, that allow us to paste this information in the session of the terminal in such a manner as to be able to configure the device. If the device was to malfunction, it is possible to quickly add a new device and to reprogram it in a fast and simple way. The disadvantage of terminal *logging is* that it only works for the configuration, not allowing us to save the operating system. In addition, if the network device is not able to use a single command to display all the configuration information, this operation must be done line by line, becoming very tedious and time consuming and subject to typing errors.

A very useful tool is Trivial File Transfer Protocol (TFTP) that is similar to FTP with the difference being that it uses UDP as transfer protocol with no authentication. When a client wants to download a file from a TFTP server or save a file on that server, client must know only the file name and the address of the TFTP server. There are no command parameters that allow the performance of authentication or allow passage to a new directory. As there is no authentication, it is not a very secure object to pass through a firewall. Most network devices use TFTP to save or retrieve configuration information, and a single TFTP server can store configuration information for each network device: if a network device is malfunctioning, it just needs to be replaced to assign an IP address and to use TFTP to download its configuration file. If the configuration information of network devices is saved, recover can be very rapid.

5.19.3 Server disasters

In addition to network failures and precautions that can be adopted to make a network more reliable, there are also failures on servers. There are a several solutions that can be taken to ensure that servers become more resistant to disasters and the only limit is the sums available and that are to be invested in this area and the type of operating system used. The prevention of disasters on a server is considered a very costly solution as it is applied to a single system.

5.19.3.1 Continuity groups

Computers, as they are electronic devices, require a source of electricity that is available and uniform. This matter becomes particularly important in relation to the server as a certain number of users use this system. A good power source not only avoids blackout and voltage dips but must also prevent the occurrence of oscillations and voltage spikes that would be extremely dangerous for the electronics that compose the server. Even a fluctuation of 10% of the current is able to generate an error situation in a computer, and if this fluctuation is constant, a greatly reduced percentage than 10 would suffice. Voltage dips and blackouts can be identified with ease because they generate system reboots while oscillations and peaks may generate more underhand errors such as those in operational applications. Current is often considered in a manner similar to network cabling, that is it is not considered until after having spent a great deal of time replacing drivers and loading patches. To avoid all this, continuity groups are used that represent a buffer system that mediates between electric power and computers, suitably equipped with batteries that are loaded from the mains supply and whose capacity depends directly on the electrical power that they must deliver in the absence of mains voltage and on the time at which this power must be delivered. The physical footprint of continuity groups depends directly on the electric power to be delivered and the time for which that power must be delivered. A good continuity group provides an excellent solution in terms of the security and reliability of any computer system and must be considered to be of fundamental importance for servers. A group of smart continuity is equipped with software that is able to turn off the server in the event that the current is not going to be restored within a certain period of time, ensuring that the server will not crash when the battery within the continuity group is flat.

5.19.3.2 RAID

Redundant Array of Inexpensive Disk (RAID) represents a system that ensures the fault tolerance in the event of a crash of the hard discs but is also able to improve the performance of the whole system. RAID divides the copies of data to be saved on multiple hard discs, preventing the whole system being blocked due to the fault of a single unit. It is able to improve performance, since the discs can work together to save large files at the same time.

The mechanism through which the data are divided across multiple discs is called *striping*. Depending on the level of RAID that we are using, the system is able to store similar types of information known as error correction code (ECC). Some RAID systems are *hot-swappable,* a term that indicates that the units can be replaced while the computer is working, by minimising the time of non-activity.

RAID can be implemented both as a hardware solution and as a software solution. In the case of using a hardware solution, the RAID controller handles all the features, making the array appear to the computer as a single logical disk. RAID software is a programming code that is part of the operating system or is available as additional software. RAID software is usually slower than RAID hardware, since it uses more CPU resources. Whatever solution we use, RAID classifications are divided into various levels ranging from 0 to 5.

RAID 0 is used to obtain better performance and is unable to provide any fault tolerance. RAID 0, instead of saving files on a single disc, divides the data on multiple hard discs, improving performance, as the storage load is split across multiple units, but with the possibility of a failure increasing, since the crash of a single disk causes deactivation of the entire array. As fault tolerance is not present, this system is not widely used.

RAID 1 keeps a complete copy of all the files for each disk: for this reason, it is also called *disk mirroring*. If a single disk fails, each of the remaining discs retains a copy of the entire file system. This operating mode prevents a system crash having to depend on any one of the discs. This means that the disk storage is limited to the size of a single disk. A RAID 1 disk array provides lower performance than in the case of a single disc, as the same data must be sent to different discs, limiting the speed of the system to that of the slower disk. To avoid this, a technique called *disk duplexing* has been developed that works in a manner similar to *disk mirroring* with the difference that several controller cards are used, reducing times as each controller must communicate with a single disk drive. This technique also increases fault tolerance since the system is able to withstand not only the failure of a disk, but also the failure of a controller.

RAID 2 is similar to RAID 5 with the difference that the data stored on disk are one byte at a time. In addition, the correction of errors is used to prevent a failure of one unit causing disabling of the array. Data transfer via block mode used by other RAID specifications is more efficient than byte modes used by RAID 2. That is why RAID 2 is found to have performance that is not ideal and this eventuality becomes evident when dealing with multiple small files. Due to its poor performance, RAID 2 is not commonly used.

RAID 3 and RAID 4 are characterised by the same specifications with the difference that RAID 3 uses three discs while RAID 4 uses four discs. These RAID specifications use a single disk for the correction of errors and use striping to divide the data on the remaining discs. In practice, with RAID 4, the first three discs contain the striped data while the fourth disk is dedicated to the correction of errors. This mode of operation allows the disk array to always remain efficient even in the absence of operation of a disk drive. ECC is used to perform a mathematical summation of data that is stored on all hard discs and its value is generated on block by block basis: in this way, it is fairly easy to deduct any missing value. It is obvious that greater processing is required and disk access appears to be slower but, in this case, the array is able to regenerate the missing data and to return the file information. This mechanism allows RAID 3–5 to restore data following a disk failure. RAID 3–4 enables commencement of a considerable improvement in performance with respect to the case of using only one disk and provides the option of ensuring a certain tolerance for failures, which is less burdensome from the point of view of storage. As data are stored on all the discs except one, the total storage capacity of a RAID 3–4 array is equal to the overall storage capacity of discs minus the capacity of one of them.

RAID 5 is similar to RAID 3–4 with the difference that all discs are used both to store data and to store the Error Correction Code (ECC). This operating mode increases the speed of RAID 3–4 that may be slowed down by the throttling the ECC parity drive. The storage capacity is further improved given that five units are used simultaneously. In a manner similar to RAID 3–4, the total storage capacity is equal to the capacity of all the discs minus one. For the reasons stated above, RAID 5 is the most widely used solution after *disk mirroring*.

5.19.3.3 Redundant servers

Redundant servers use the principles of RAID and apply them fully to themselves. Reference is often made to server fault tolerance. Redundant servers are able to provide one or more complete systems that are able to enter into service if the primary server fails irrespective of the fact that the malfunction is dependent on a disk, a memory error or a motherboard failure: as soon as the primary server no

longer performs its job properly, the redundant system replaces it. Redundant servers usually have two communication channels in common, as shown in Figure 5.75.

A connection is the connection to the network while the second connection is the connection between two servers. Depending on the implementation, the latter connection can be generated through the use of communication or proprietary cards or high-speed Ethernet cards. The secondary server is upgraded via this high-speed connection. Depending on the type of implementation, updates may be related merely to the information on the disk or also related to the memory addresses. In the latter case, the secondary server is able to take over from the primary server in a straightforward manner (hot backup).

High-speed connections are not provided for in all solutions for the redundant servers, through the use of the normal network. The advantage for the use of the network is that the secondary server can be placed wherever we want, even at a remote site. As the secondary server can be placed in a secure and possibly distant place, this configuration is more durable and reliable against problems that may affect the entire building such as voluntary attacks (sabotage, theft and tampering, use of explosives, etc.) or incidental events (fire, lightning, flood, etc.).

If we do not have a connection between the two systems, the memory information is not shared and the secondary server is not able to subenter in a straightforward manner (cold backup). In this sense, the request of a client previously sent to the primary server must expire and be cleared before being able to be satisfied by the secondary server following a new request. This entails a delay of a few minutes before the secondary server can be made operational. Another disadvantage is the greater use of the network that does not use a dedicated connection between the two systems. Redundancy of the server can also be made at the operating system level or as an add-on product. There are several products available and each of them supports redundant servers in a different manner.

5.19.3.4 Clustering

Clustering operates in a manner similar to redundant servers with the difference that all systems participate in service requests. The cluster works as a smart unit to balance the traffic load. From the client point of view, the cluster seems a very fast server. If a server fails, the work continues undisturbed even if with a reduction in performance. In this sense, clustering is found to be more efficient with respect to the redundancy as the secondary systems ensure reduced time of processing and do not wait for the failure of the primary server to operate, allowing maximum level of use of the existing systems.

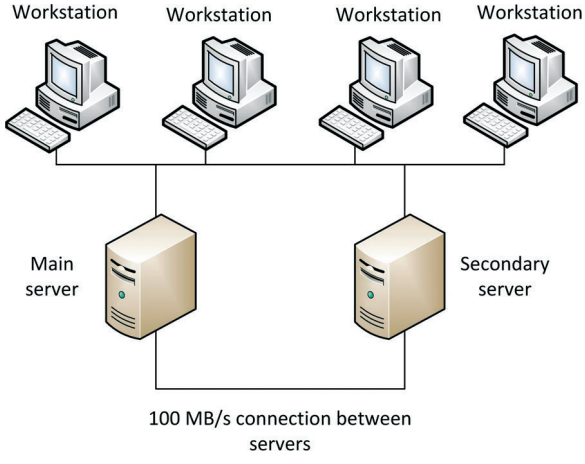


Figure 5.75 Diagram of a redundant server.

Clustering is, therefore, a very functional solution to ensure a high fault tolerance and optimal performance. It is available for most operating systems.

5.19.3.5 Data backup

The best way to protect our data from disaster, damage and loss is through duplication or backup. Different technologies are used: starting from the traditional tapes up to the most recent backup via the Internet that allows us to store data in a different place and to reduce in-house costs for maintenance and backup procedures.

Tape backup is a traditional historical way of protecting and restoring lost, damaged or deleted information. All the concepts discussed so far refer to the maintenance or restoration of the server as a service but none of these concepts is able to retrieve a particular file that has been accidentally deleted or damaged some time before. In this sense backup, and in particular that on tape, is able to safeguard information that is stored on the server. However, there are other backup types such as arrays of hard discs, cds and dvds.

Tapes are recommended if the data must be stored for less than a year. If tapes are to be used for a longer period, the same must be stored in optimal climatic and environmental conditions or it is necessary to resort to another type of media such as optical media.

Backup is usually supported by suitable software and most of them support three methods to choose the files that should be archived. They are full backup, incremental backup and differential backup.

Full backup consists of the archiving of all the files on the server. It is the best method when it is time to activate a procedure to recover from a disaster as it contains a complete copy of the file system that can be found on a single tape or on more than one tape. The problem of full backup is the amount of time necessary to execute it, since it requires more time than any other type of backup. Full backup is usually performed at night because the server is subjected to a smaller workload and the files to be stored tend to remain unchanged: if the amount of data to be duplicated is high, this work could not be done in one night and it is not therefore possible to execute a daily backup.

Incremental backup is restricted to copying the files that have been updated or changed recently. In this way, the process of duplication is speeded up as us store only the files that were modified after the last backup. A full backup is typically performed once a week and an incremental backup every day, generally at night. If the server must be reset, the first step is full recovery and subsequently with all the incremental backups that were created after the full backup. The main disadvantage of incremental backups is that deletions are not recorded and there is, therefore, the possibility that by the end of the process more data can be restored than can be contained on the hard discs. This problem is typical of incremental backups and for this reason system administrators prefer to run differential backups.

Differential backup are different from incremental backups because they record all of the files that have been modified since the last backup. It is not limited to storing the files since the last backup: in this way, we reduce the chance of encountering problems of space at the moment of recovery, despite the possibility of having more data on the backup media with respect to the capacity of the unit, even if this possibility is quite rare. A further advantage of differential backup is that only two backup media need be restored after the crash of the server, speeding up the process but also reducing the risk of failures.

A further type of backup is backup via the Internet that is part of a broader context of remote management. Using specific products that are normally available on the market, it is possible to copy encrypted data of an organisation automatically and with periodicity and to store it on another appropriately protected site. Backup via the Internet provides a series of advantages, including:

1. low administrative overhead, as the local implementation and maintenance are limited to a local software package as this type of backup works seamlessly, without requiring the intervention of the

dedicated staff of the organisation. In addition, it is not necessary to check the quality of the backup media and its deterioration, nor, indeed, its storage in a suitable air-conditioned and protected room;
2. reduction of the risks because the data are always stored in another place and thus averts the danger whereby a local disaster can damage the computing resources and local backups. Since that backup data are not stored on media, the probability of losing control over proprietary, sensitive and confidential data is practically reduced to zero because no attacker can steal the backup media.

However, there are also certain disadvantages that are as follows:

1. speed, because even with a broadband line, considerable time may be required where several gigabyte of data need to be restored. Even if the tendency towards the availability of higher bandwidth is continually increasing, the rate of data growth of an organisation is higher, making the process increasingly expensive from the commitment of time point of view;
2. recoverability, because the time required to restore the backup data via the Internet appears to be greater than local backup not only due to the reduced availability of bandwidth but also due to the fact that the backup service must identify the data and start the backup process, adding a further load to a process already limited in itself.

Even though there are disadvantages, many organisations prefer backup via the Internet, safe in the knowledge that the data are stored in another safe place.

5.19.3.6 Application server provider

The ASP solves the problem of server failures and data loss in a very unique and original way. All the services concerning data operate in another place while the client application for the end user is located within the organisation and all the data and services are provided via the Internet. The ASP needs only guarantee availability and redundancy not only of data but also of the applications. It is a solution that is fast gaining popularity, especially in small organisations that do not have adequate financial resources to afford the staff dedicated to the management of network resources and information systems. In any case, this solution is increasingly appreciated also by large organisations that may have broadband communication lines.

ASP is still also characterised by certain disadvantages. In fact, if our Internet connection fails (a common event, especially outside the major urban centres), the organisation is no longer able to access data and services, and freezes completely. In addition, should a dispute arise between the organisation and the ASP service provider, the latter might halt the service, stopping irreversibly operation of the organisation.

5.19.3.7 Server recovery

Even if the backup of data is suitable for protecting files, it is not an optimal solution to restore servers. If a server fails in a definitive manner, then we will need to create it again on a new platform. The procedure to be followed is divided into the following steps:

1. installation of the operating system on the server;
2. driver installation;
3. installation of service pack;
4. installation of hotfixes and security patches;
5. installation of backup software;
6. installation of any patches for the backup software;
7. restoration of the last full backup;
8. recovery of any incremental or differential backup.

This process is very long and expensive. A possible alternative is the use of software specifically designed for the restoration of the server. In general, these packets generate a series of boot discs and an image of the server. Boot discs allow us to restart the system without resorting to an operating system. Thereafter, the recovery software server accesses the image generated previously and restores all data on the server. After this, the server is restarted and the system is operational again. There are several products available for this feature that are usually provided, together with backup products. The only disadvantage of server recovery solutions is that they save the entire system as image: this operating mode speeds up backup and recovery but does not allow access to individual files. For this reason, even if we are using this solution, it is necessary to perform backup regularly in order to be able to recover, if necessary, the files that can be corrupted or accidentally deleted.

5.19.4 Disaster simulation

When developing a solution for disaster recovery, it is very important to check and document the solution choices as this is the only way to ensure that the recovery plan functions correctly. If we document the process, it is certain that, in the event of an emergency, the proper procedure is followed.

In this sense, it is possible to perform a non-destructive test that allows us to test plans for prevention and recovery without affecting the normal flow of operations. This is the preferred method because it is not desirable to generate a disaster in order to put a potential solution to the test.

There are many ways to implement a non-destructive test. The simplest method is to use alternate hardware to simulate a disaster. For example, we could use a server identical to that in use in an attempt to restore the back-up on this alternative system.

Not all organisations are able to afford redundant hardware available to check the recovery plans and in any case, these tests should be performed at night or during non-working days. Simulation of a disaster performed beforehand allows us to verify our ability for complete recovery after an actual disaster.

Documentation of the procedures to be followed is also very important, which should be prepared during the simulation in such a manner that the typical stress that is generated during actual disasters is not experienced.

5.20 Network security policy

To make a network secure, it is often necessary to change the way users work. It may be necessary to limit the functionality of computers and networks that are being used by users and it may be necessary to implement policies and procedures that many users could consider an invasion of their privacy. It is very likely that policies and security procedures will be characterised by a certain impact on users and may limit their activities. It is very important that we create in our organisation a structure that represents support to solve security problems. This section illustrates the basic concepts to create a security policy in our organisation and references to successfully create a security plan.

In order for a security policy is to be successful, it is very important that the same is supported by corporate management. In addition, this policy must be adaptable to all staff. The fundamental points of the policy must be included in an employee handbook for easy reference. In addition, the same organisation must always have a ready and rapid response to any breach of security policy. Before continuing in the illustration of the fundamental concepts to implement a correct security policy, it should be remembered that:

1. every organisation that uses computers and networks must have a security policy;
2. even if the security policy of each installation contains specific procedures and policies, the

organisation and the basic concepts of the security policy must remain the same;
3. since the security policy impacts in different ways on companies, key figures must be involved in each sector and department to discuss and evaluate together the implications that the security policy may have on them;
4. before the development of a security policy, it is necessary to carry out accurate analysis of the risks of the organisation. In many cases, it is learnt that external attackers represent a limited risk while most of the risk is represented by internal staff that, voluntarily or involuntarily, can damage or distribute the data of the organisation;
5. a correct security policy should specify what to protect rather than how to protect;
6. a security policy must be planned with accuracy before implementing it;
7. a security policy must also contain the different procedures through which to respond to violations of the policy;
8. a security policy is valid if and only if it is implemented and respected;
9. before accusing any employee of breaching the security policy, we need to ensure that all employees of the organisation know each procedure established by the security policy.

The implementation of a correct security policy should provide for the development of a management plan for the security of computers and the network. To ensure the security of computers and the network, not only external attackers and viruses should be considered because most threats can come from internal users. The development of a security policy is divided according to the following steps:

1. Determination of the resources that must be protected. Evaluation of the specific characteristics of each resource as a typology of the users that use the resource, users who can modify the resource, etc.
2. Determination of the subjects from which we must protect our installation.
3. Determination of the likelihood that there may be threats to the system. If our business is local and our market share is small, it is more probable that the risks can come from an ill-intentioned seller rather than from an external attacker.
4. Implementation of measures for the protection of an effective but not overly expensive installation. Password security, the use of encrypted records and the use of firewalls are effective and inexpensive examples.
5. Continuous review of procedures, continuous execution of audit to ensure that employees use the procedures and policies of security and improvement of the components of the network security every time a vulnerability is found.

When an organisation's security policy is being developed, it should be remembered that the objective is to define the procedures to avoid incidents that affect the security and the actions that must be implemented by the organisation in the event that such incidents occur.

The creation of a security policy may be a long and complex operation. The biggest mistake that is often committed is the high number of intended actions in a short space of time. Computers and networks are key instruments for the effectiveness of an organisation and for its employees. Ensuring a correct security policy requires that the same be divided into smaller and manageable operations.

Before consulting the various sectors of the organisation, it is necessary to develop a corporate computer security policy and to select the key elements that must be implemented first. In general, the first step should be to address the elements that can be dealt with in a short document. If we restrict the topics to a short document, it is possible to concentrate on the key elements, allowing all individuals that are participating in creating a security policy to understand the contents of the matters to be dealt with. In the initial phase, it is advisable to mainly focus on what to protect rather than on how to protect.

Table 5.6 shows a list that is a useful aid to distinguish *what* must be protected from *how* it must be protected.

Table 5.6 Distinction of what must be protected from how it should be protected.

What	How
Website	User name and password
Corporate intranet	Cryptography
Email	Training
Data on personnel	Times of use and access
Sales information	Backup
Research and development (intellectual property)	Limitation of remote access
Communications with laptops	Virus detection
Corporate database	Security auditing
Software piracy	Firewalls
E-commerce	Digital signatures
Conferences and video conferences	Modem call-back for remote logins
Data for accounting and finance	Smart cards and other authorisation techniques

When exploring the various resources, it is necessary to define with precision who can use the system and its related services. The security policy should define explicitly who is authorised to use such resources, how and at what times. Many attackers, in fact, perform their attacks after working hours. If we restrict access to the systems to these times, we can make many systems more secure. Having acquired from users more information on the manner in which they use the system currently and how they are planning to use it in the future, each element that is used by them and the manner in which this element is used (local or remote, using mobile devices, etc.) must be recorded. Ultimately, a list will be produced that will define the assets of the organisation. Then, for each element, we should identify the users (internal and external to the organisation) that must be able to access the resource and in what manner. If we find a resource to which too many groups of users have access, we need to identify and define the manner by which each group uses the resource. In many cases, each group may use the resources in the same manner and this simplifies protection of the resource itself. In many cases, it will become evident that a group needs to display the resource while another group needs to modify the resource. It is often also discovered that each group uses different parts of a resource, which implies that the resource should be divided into small-size components.

It has already said that it is not possible to develop, in a relatively short period of time, a security policy for computers and networks. The creation of a policy that is efficient and that appears to be accepted by employees and applicable in an efficient manner by the organisation requires time. In particular, it is necessary to acquire more information from the following groups of users:

1. Division of personnel to ensure that employees are deemed to be responsible for the security policy. In this sense, employees must be familiar with every policy and every procedure, and to do this, a training phase must be provided. In addition, in order to reduce every problem, each employee must countersign by way of acceptance of the security policy and procedures of the organisation's computers and network resources.
2. The involvement of each division of the organisation because the policy of computer and network security will be characterised by a considerable impact on all employees of the organisation. The policy could, in some cases, clash with the normal work procedures and this factor must be taken seriously. If an employee should breach a security procedure, the organisation must respond efficiently and immediately and this requires the support of the company management. It must also be remembered that the security policy of the computer and of the network of an organisation is

not effective if the same generates too many constraints that cause deterioration of the performance of employees. To avoid the use of policies characterised by reduced practicality, it is necessary to fully utilise the experience of the manager of the organisation to lower the security policy with greater ease in the context of the organisation in question.

3. Division of telecommunications, since at present, within companies, it is always more difficult to distinguish between the subjects responsible for computers and the individuals responsible for network and telephony. Since most of the connections via the Internet are guaranteed by a telephone company, there are often situations of conflict between telecom managers and information technology managers. In any case, the subjects responsible for telecommunications must actively participate in definition of the aspects of the security policy that affects the network and the computers.
4. Programming division, as in an organisation, programmers who generate applications play a critical role in determining the way and the place from which users can have access to data. In particular, programmers are well aware that where the most important data are located to which users must have access and in what manner it is possible to have access to such data. In addition, programmers are well aware of the data risks depending on the manner in which applications are divided.
5. Legal department, in that, in order to enforce the security policy of computers and of the network, it is very important that the staff of the legal department approve every security policy and procedure to be certain that the same does not lead to discrimination and does not breach privacy rights and the legitimate expectations of its employees. In addition, irrespective of the manner in which the security policy is written, it is very important to have certain flexibility in interpretation, particularly when a member of the organisation breaches a policy or a security procedure.
6. Sales and marketing division, as the staff of this division generates, unknowingly, the greatest risk to corporate data. In fact, very often the staff of this division travel with laptops that contain sensitive data such as sales forecasts, turnover and products sold. Furthermore, the staff of this division connect to the core business systems from remote locations, increasing the level of risk. In most cases, network administrators must do a remarkable job to properly program the firewall and proxy servers in such a manner as to control and limit external access to the data that staff subsequently leave on their laptops or on a storage medium that is exchanged between a computer and our personal computer that we use at home.
7. Financial division, since the implementation of a security policy also requires financial resources, faced with the costs to be implemented, both from the point of view of software as well as hardware. In addition, the organisation may have recourse to dedicated insurance against specific risks and the choice depends on financial heads.

It is very important to remember that the main direction in which to aim at the beginning of the activity should be especially towards those elements that must be protected and not towards the manner in which they are to be protected. When the various elements are examined, it is necessary to assign to each of them a well-defined level of importance. Depending on the level assigned, it is possible to identify priorities for action and for implementation of security policies and procedures.

Once it has been determined who should be enabled to access system resources, it is important to define and disseminate guidelines, in an acceptable manner, to use resources. The fact that a certain group of users access the resource in a certain way does not ensure that this access mode is dictated by security, but only by habit. At the end of the process, different indications could emerge for the various user profiles. The security policy should define with extreme clarity what is acceptable and unacceptable behaviour and must also include the option of limiting access. An acceptable use policy should also establish very clearly that the various users are themselves responsible for their own actions on computers, on the system and network of the organisation. In any case, there is always the responsibility of individual users regardless of the security mechanisms used. The security policy, must,

in addition, establish very clearly that the organisation does not allow the sharing of accounts or the option to override the security systems.

A very important element to consider in the process of developing a security policy is that the resources of our organisation and the commitment to ensuring security are able to bring to the organisation itself economic benefits. Even though this may appear obvious, very often organisations respond to threats that involve other subjects when, in reality, there is actually quite limited risk. For example, we often hear from the media that hackers were able to break into the network of an important organisation leading many smaller companies to begin investing in protection systems against external intrusions yet forgetting that most threats comes from internal users.

After system resources have been identified, it is necessary to take the risks involved into consideration. Also in this situation, it is necessary to synthesise this analysis in a brief note, stating the key elements. Below are listed, as a reference, a series of elements that may entail risks.

1. Hardware:
    (a) theft;
    (b) fires;
    (c) floods;
    (d) electrical damage;
    (e) fraudulent use;
    (f) downtime due to failures.
2. Websites:
    (a) hacker attacks;
    (b) denial-of-service attacks;
    (c) downtime caused by problems with the Internet or with the telephone company;
    (d) downtime due to computer failures.
3. Email:
    (a) spamming;
    (b) viruses;
    (c) personal use by employees;
    (d) distribution of intellectual property.
4. Database/intranet:
    (a) unauthorised access;
    (b) theft of corporate information;
    (c) fraudulent use;
    (d) denial-of-service attacks;
    (e) disk errors;
    (f) data loss.
5. Users computers:
    (a) viruses;
    (b) unauthorised personal use;
    (c) distribution of intellectual property through removable media;
    (d) theft of corporate information;
    (e) software piracy.
6. E-commerce sites:
    (a) hacker attacks;
    (b) theft of credit card numbers;
    (c) use of stolen credit cards;
    (d) theft of customer information.
7. Remote access/portable devices:
    (a) theft of devices containing business data;

   (b) unauthorised remote access;
   (c) unauthorised extraction of corporate data;
8. Business applications:
   (a) unauthorised access by users;
   (b) dangerous programs (Trojan horses, viruses, backdoors, etc.);
   (c) programming errors (bugs).
9. Corporate offices:
   (a) fires;
   (b) floods;
   (c) disasters;
   (d) unforeseeable problems (blackouts, closing of roads, security threats) that prevent workers from accessing computers;
   (e) loss of access to the Internet.

Of interest is the fact that the list of threats only identifies threats and, therefore, is not a document designed to prevent each of them because prevention inevitably involves compromises between use, costs, time and other resources of the organisation.

Once the corporate assets have been determined, it is necessary to identify priorities in relation to those assets necessary for operation of the organisation and those whose loss would cause only minor consequences. The next step is to try to assess the potential of every threat and therefore the relative probability with which each may occur.

Risk analysis concerns what must be protected, the threat from which protection is required and the manner in which to protect the assets. For this reason, it is necessary to consider all the risks that need to be assessed according to their level of severity. Risk assessment also requires economic decisions relating to the assets that are to be protected. It is clear that correct implementation of security requires that the amount spent for this purpose should not exceed the value of the asset to be protected. It should also be remembered that the basic objectives of security are availability, confidentiality and integrity. The final goal, for each threat, is to understand the way in which the threat may compromise the three objectives of the assets and the manner in which to provide protection against the aforementioned threat for each objective.

Once the assets to be protected have been identified, the threats to these assets and thus the potential of their loss must be identified. It is also important to identify threats from which it is necessary to protect assets. A threat that is very widespread is unauthorised access to information resources that can be carried out in many ways. A typical way is the use of the account of another user. In essence, this involves the unauthorised access to a computer without relevant permission. Unauthorised access varies its impact on security depending on the type of computer, its level of access to the system, etc.

A further threat is the dissemination of confidential information. For this reason, it is necessary to quantify the value and confidentiality of the information stored in the electronic archives. If passwords are moving in an uncontrolled manner, it is possible to perform any type of unauthorised access. The diffusion of a simple quote can be a considerable bonus to a potential competitor while a technical document may show the result of a very long period of research and relevant economic investment. For this reason, it is very important to understand which services are essential for every service necessary in order to quantify the effect on the organisation of a failure or loss of this service itself.

It is very important to quantify the level with which we want to ensure the security of the data contained in the systems first to ensure user's access to services. This assessment allows us to determine the level of confidentiality that users can store on their computers. If it is not possible to secure a system, it is imperative that users are prevented from storing sensitive information. For this reason, it is very important that users are aware of the use of confidential information and that they know which services are most suitable for the storage of confidential information. This part of the security policy

should provide for the storage of data via various modes (discs, tapes, servers, etc.). It is also very important to coordinate the security policy with the policy on system administrators with respect to normal users.

Even if the security policy aims to reduce the risk of threats against assets (computers, network, data, etc.), it may not reduce the risk itself to zero, a residual risk thus always remaining. For this reason, insurance companies stipulate that are able to compensate for damage caused by theft, fire, etc. or due to employees and customers. Once we have identified the threats, it can therefore be very useful to take out insurance against the interruption of normal operations of the organisation in the event of critical situations. Since insurance against damage may prove to be quite expensive, many organisations tend to avoid this. Depending on the type of business and on the potential risks for the activities of the organisation, insurance against disruption of the activity represents a safety buffer that is interposed between the security policy and the procedures that have been used.

In order to determine the risks, we need to identify the vulnerabilities. The areas that usually experience more problems are:

1. unauthorised users that use the APs. If there are several APs, the risk of unauthorised access to computers and to the network increases. Connections pointing to networks that are outside the organisation may represent a vulnerability through which external users can enter the internal network. A network connection, in general, ensures a certain number of network services and each service can, potentially, be breached. Dial-up lines, depending on their configuration, may provide access to a login port on a single system. If this line is connected to a server, the same will allow access to the entire internal network, which is a serious threat to the security of the same;
2. systems that are not configured correctly, which represent quite a significant risk in terms of security. In fact, operating systems and related software have become very complex, and understanding of their full functioning can be very difficult even for experts in the sector. It must be remembered that systems and networks managers are not always specialists and are individuals that are selected from within the organisation. Moreover, it is often the case that hardware and software manufacturers choose initial configurations that are not very secure for all the environments, in order to make the installation process easier;
3. software defects that are often used by attackers to perform unauthorised access. To avoid this, it is necessary to implement all the updates that are made available by manufacturers;
4. internal users of the organisation, which represents one of the most prevalent threats. Very often employees have direct access to computers and to the networks, and the ability to access, just physically, increases the risk of breach of systems. In this sense, internal users can operate on work stations to gain privileged access. Furthermore, any local network access allows access to sensitive data passing through it, and the accessing of other internal resources.

Often, as with most of the management documents of an organisation, corporate policy can develop into a complex and full-bodied set of rules. For this reason, it is necessary to proceed by degrees and sectors, establishing them one by one: the policies and rules of the individual sectors will then extended and integrated to form a single document. These procedures can involve email, the Web, username and password, etc.

With regard to email, every employee of an organisation should have their own email account. Personnel will provide an appropriate request to the IT department. If the employee leaves the organisation, personnel must always communicate the situation to the IT department in order to close the account. The address chosen should be as simple and as straightforward as possible and of the type firstname.surname@organizzazione.estensione. The organisation must give each employee suitable detection software for viruses in email messages, and the employee should never disable the use of such software. The organisation must not use or authorise spamming as an advertising, promotion or marketing technique. The organisation must realise that employees cannot, in most cases, avoid receiving email messages from their friends and family. But it should be remembered that each email

message consumes work time and business resources and for this reason employees should treat email in the same manner as they do personal phone calls. For this reason, the exaggerated use of company email for personal purposes may not be considered acceptable. The organisation can reserve the right to check the content of files and email messages, for reasons of internal security, consistent with the privacy rights of their employees. Email messages are one of the principal means of infection by computer viruses and for this reason, employees must not open attachments received from external entities that are not known and not trusted. The organisation will provide employees with encryption keys to encrypt the content of email. Employees should use cryptography to encrypt all sensitive information that is sent through email. The organisation cannot authorise the use of email for the sending of unsuitable content.

With regard to the Web, navigation on the same can produce a considerable loss of time. For this reason, employees should not surf the Web for personal purposes during working hours. The organisation may, if appropriate, provide one or more locations to put at the disposal of its employees for navigation during break times. It is obvious that employees may not use this facility for unsuitable and inappropriate activities. The organisation may not allow the presence, on its own terminals used by employees, of unsuitable material whatever this may be. Websites allow users to download programs but this possibility is also a means of propagation of computer viruses: for this reason, employees should not be allowed to download programs from the Web. If an employee requires a particular program, the same should send a special request to the IT department that will evaluate the actual utility and relative security in terms of the organisation's policy. The organisation can reserve the right to check websites that are visited by their employees through computers and business services, for the purposes of a correct corporate security policy and compatibly with the right to privacy of employees.

Regarding username and password, all employees requiring access to company computers and the network must be equipped with and use their username and password. Personnel will provide the IT department with a special request for each account. If an employee leaves the organisation, the procedure mentioned earlier must be used. Employees should not share their account with another employee. Employees must never share computers with each other without special authorisation from the delegated person. Employees should never provide their username and password to anyone for any reason. Employees must change their password frequently and at least every 30 days. Employees must never leave their computer accessible in their absence. Account passwords must be composed of at least eight characters composed of uppercase letters, lowercase letters, numbers and punctuation marks, with the prudence of using the maximum randomness possible, in order to make it difficult for an attacker to identify the password using brute-force or other attack techniques. The organisation can use smart cards, biometrics or other identification techniques to increase the level of security and to decrease the risks.

In addition to what has already been seen, there are numerous other policies and procedures that are very useful for a proper security policy such as:

1. data backup;
2. use of portable computers;
3. remote access;
4. file sharing;
5. system administration and many other aspects.

Very often, in the past, software companies struggled greatly to avoid their products being duplicated and sold below cost, thereby violating copyright laws. It often happened that large organisations bought a copy of a program and then performed multiple and illegal installations on many computers. This was not permitted as most programs are meant to be installed on a single computer. Nowadays such behaviour is generally sanctioned heavily throughout the world. A security policy should state explicitly that users should never install software of any kind on their computer and in particular software downloaded from the Internet that may contain viruses and Trojan horses that

would expose the entire system and the related network to huge risks. As stated, there may be users that require a specific program to be able to work more efficiently and in this sense, the organisation should provide policies and procedures appropriately intended to simplify the acquisition and installation of software, making this operation at the same time safe. If relevant directives are included in the employee handbook, it is important to ensure that all employees are aware of the procedures relating to the new software that is authorised by the organisation.

The security policy should indicate who is authorised to provide access to the system and network services. In addition, the policy should identify the types of access that may be provided. If within the organisation there was no subject that could decide on which users could access the system, it would not be possible to obtain control of users that use the same system. If, on the contrary, it is possible to identify with precision a person authorised to provide access, it is also possible to know, in the future, who provided certain access in case of problems or disputes. There are several schemes that can be used to control the distribution of access to services. When a person who controls the access to services must be identified, the following points must be considered:

1. To understand if the distribution of access must be from a single point or several points as either of two options can easily be chosen. In this case, both the security and ease of use must be considered. In any case, a centralised and unique system only tends to be more secure because it is easier to control with respect to a distributed system.
2. To decide on the methods that must be used to generate and close accounts. From a security point of view, the mechanisms that can be used to generate an account must be taken into consideration. In less restrictive situations, subjects that have authorisation to provide access may also enter directly into the system and generate an account manually. It is obvious that these mechanisms require an enormous amount of confidence in the person that has been authorized, that in turn has a great number of privileges on the system. The solution that is located at the other extreme is the use of an integrated system that authorised persons can use to generate accounts or that the users themselves can use to generate limited access accounts. It is very important to bear in mind that, even if an automatic procedure for account generation is identified, the probability of abuse cannot be reduced to zero.
3. To develop well-determined procedures for generating accounts that must be well documented in order to avoid confusion and to reduce errors. The vulnerability of the security of the procedure for the authorisation of accounts does not only depend on the possibility of committing breaches but also on the possibility of making mistakes: if we have clear and defined procedures, we can avoid many mistakes. It is, moreover, appropriate to be certain that those persons who must follow these procedures understand their true importance.

One of the most delicate aspects of network security is the responsibility of giving users access to the system. It is very important to select a password that would prove to be very difficult for an attacker to find, avoiding the use of a password derived from or related to username, real name or generated from algorithms that can be easily breached. Users should be discouraged from using the same password for an arbitrary period: continuous changing of the password is a security guarantee that reduces the risk of unauthorised access. It is also very important to encourage users to change their password upon the first connection. It should also be remembered that there may be users who never use accounts, making their initial password vulnerable. In many systems, the policy to disable accounts that have never been used is adopted to encourage the rightful owner to make a new request when they need to do so.

A very important decision from the security point of view is choosing the subject that has the privileges and the administration password of the system. System administrators must be empowered to carry out accesses but it can also happen that other users require special privileges and a correct security policy must also provide for such an eventuality. The reduction of privilege is a means to avoid the possibility of attacks coming from within. A big problem is the search for the right balance between

restriction access and privileges that users need to be able to carry out their work: in this sense, it is recommended that each is given only the privileges necessary to carry out their activities. Subjects who have specific privileges must also have relative authority that must be very clear in the security policy: if the subjects that have privileges are not also entrusted with the related responsibilities, there is the risk of not having precise control of the system with related difficulties in the case where there is a breach of security of the system and network.

It should be remembered that it is absolutely necessary to comply with the privacy of the user and to balance it with the need, by the administrator, to collect the information necessary to perform a diagnosis of any problems. There is a distinction between an investigation concerning possible breaches of the system and the needs of the system administrator to collect information necessary for diagnosis of the problems. The security policy should indicate with precision the level through which the system administrator can examine the files of users to gather information or to detect problems and the rights that are to be assigned to users. The creation of a plan that concerns the obligations of the system administrator to ensure the privacy of the information depending on specific needs can also be considered. In this respect, the following points must be taken into account:

1. The administrator may need to check or read user files for security reasons.
2. Responsibilities related to the fact that the administrator accesses the file of users.
3. The possible right of network administrators to examine the traffic of network and hosts.

The access rights of users and administrators must be determined separately just as access to confidential information and access to non-confidential information that is available on the network should be distinguished between.

It is practically normal that, once a certain security policy has been determined, it is the case that a user breaches these procedures. The security breach can take place due to negligence, error, misinformation, misunderstanding, etc. It is also possible that a user or a group of users perform with continuity an action that breaches the security policy.

When a breach of the security policy becomes apparent, the policy itself should provide the process that must be implemented to have an appropriate and immediate response. It is very important to carry out an investigation to determine how and why the breach took place. Then, certain corrective actions must be put in place. The type of response actions should depend on the topology of the breach that was performed.

Security policies can be violated by a wide range of users, both internal and external. Administrators can group users into internal and external according to the administrative, legal and political classifications. This grouping determines the type of action that must be implemented to provide a response to a possible breach that can range from a simple written reprimand up to a legal complaint. The actions must be based on the type of breach but a series of actions must also be provided that are based on the security policy.

It should be remembered that, with regard to compliance with the security policy, the best defence is correct and adequate education. If there are external users that use an internal computer in a manner that is not appropriate for compliance with the security policies, it is the task of the administrator to check that those users are aware of the security policies that are provided. This can be very useful in the case where legal actions are involved. If users are using their computers illegally, the problem is substantially similar: efforts should be made to understand which user has breached the policy and how and why this breach occurred. In a situation where a local user breaches the security policy of a remote site, the local site must accurately determine the actions that must be taken in respect of the local user. The security policy should provide a series of procedures and actions to protect the network against activities that can be performed from a remote site. Breaches of the network by a remote site can also involve legal aspects that must be considered when the security policy is being created.

The local security policy must include procedures that relate to the interaction with external organisations that may be other sites, agencies, legal organisations external response organisations and

press agencies. The procedure must determine the subject that is enabled to contact these organisations and the manner by which each situation must be handled. The security policy must also include:

1. the subject that is authorised to liaise with the press;
2. the subject that is authorised to liaise with legal and investigative agencies;
3. subjects that may provide various types of information.

The security policy must provide, in addition to security instructions, the procedures for the management of incidents and critical events. Any breach to this policy must be considered in the same way as an incident. Procedures must therefore be created that relate to all types of breaches of security policy and every incident that derives from such breaches. When an incident is being considered, the type of response to be used surrounding the incident itself must also be identified.

Once the assets to be protected and the risks to which these assets are exposed have been identified, we need to determine how to implement the controls that protect the same. It is necessary to identify controls and protection mechanisms able to suitably address threats that have been identified during risk assessment and to try to implement these controls in the least expensive way possible. Checks that have been identified are the first line of defence in the protection of assets and it is therefore very important to be certain that these checks are appropriate to the circumstances. If the increased risk for systems is external users, it does not make much sense to use sophisticated access systems, such as biometric ones, to authenticate internal users. If, on the other hand, the main threat is the unauthorised use of computing resources by internal users, then the activation of very restrictive control procedures is recommended.

In order to define a security policy, common sense must also be used. Very sophisticated security mechanisms can induce a certain impression and can also offer a certain efficiency, but it does not make sense to set up an excessive monitoring system when the simplest aspects are being overlooked that, despite their simplicity, are not necessarily any less dangerous, as far as security is concerned. For example, a user in possession of an inadequate password nonetheless represents a risk for the entire organisation, regardless of the efficiency and complexity of the existing security controls.

A very effective method to protect assets is the use of multiple strategies in such a way that if a strategy fails or if an attacker manages to circumvent it, security is guaranteed by another strategy. If several simple strategies are used, there may be the possibility of protecting assets in a more effective manner than with a single complex strategy. For example, login procedures can be combined with call-back modems. In this sense, multiple approaches can be used that provide different levels of protection of the assets of the organisation. It is very important, however, not to provide an excessive number of security mechanisms to avoid overloading normal work tasks with a high number of checks, always remembering the assets that are meant to be protected.

In terms of the security of a system and its network, if only one computer is not secure, the whole system is not secure. If the attacker has physical access to a computer, the same can stop its operation, enter into privileged mode, replace or change the discs, insert Trojan horses and in any case perform undesired activities that are difficult to prevent. For this reason, it is necessary to identify the critical links, the most important servers and other computers that perform the important tasks and place everything in a secure room. It has already been seen that systems such as Kerberos require that the central machine is located in a secure and controlled room. If we are unable to guarantee the physical safety of our computer, then at least the same should be made secure. In any case, certain access limitations should be considered both to non-secure computers and to secure computers. In this sense, close monitoring of the identity of the persons that need access to computers is required. It should also be remembered that custody and maintenance personnel possesses a copy of the keys to the rooms in which the computers to maintain safeguarded are kept and this contributes to increasing the overall risk.

Once the security policy has been drawn up and applied, it is necessary to start an appropriate process in such a manner as to involve all relevant subjects. In this sense, it is not enough to simply send

out a copy of the prepared security policy but instead it should be disseminated carefully in all possible ways and then wait for a certain period to receive feedback given that every subject involved knows well how this policy will impact their normal work, and in this sense, it is necessary to find the right compromise between security and productivity. To disseminate the security policy in the best way, it is appropriate to organise specific meetings involving all levels of the organisation, in which to illustrate the same in details and where all the aspects can be discussed. It is very important to remember that the security policy is effective if the same is applied to all members of the organization that must make an extra effort both at the initial stage and at the stage where the policy enters into force.

Once the policy has been adequately released, it is very important to keep attention focussed on it, perhaps by resorting to internal means of communication such as the organization intranet or by sending periodic emails and bulletins.

It is also very important, in order for employees to assume all responsibility, that the same possibly sign a document that sanctions that they have viewed the security policy, that they have understood it and that they will apply it in their normal work. This document is very useful in the case where an employee commits a serious breach of the security policy, allowing the organisation to resort to all the appropriate channels.

This page intentionally left blank

CHAPTER 6

SECURITY OF WIRELESS NETWORKS

6.1 Introduction

This chapter addresses the security of wireless networks and its fundamental concepts. Certain types of networks and protocols have already been discussed, in general terms, in Chapter 1. These concepts will be taken up again and examined in this chapter together with other concepts illustrated in Chapter 5.

6.2 Introduction to wireless networks

For the most part, wireless networks use radio frequency or microwave electromagnetic waves (henceforth, for the sake of brevity, the term "radio frequency" will be used), whose fundamental principles have already been discussed and that in any case will be further analysed.

Essentially, radio frequencies represent a physical quantity in the form of an alternating current that passes through a cable, produced by an electrical device, which is radiated into space by an antenna. Such an antenna radiates a certain power, generating radio waves that propagate in space in all directions if the antenna is omnidirectional, gradually decreasing their amplitude as they move away from the same antenna until they reach a size so small that they can no longer be detected by any receiver.

Radio transmissions were discovered by Heinrich Hertz around 1880. He used, as a starting point, the research on the electromagnetic theory of light of James Clerk Maxwell. Hertz discovered that where an electrical signal of a certain size is used, it is possible to send a signal through a non-conductive medium, such as air or empty space. This is the basis of radio signals and wireless communications.

When radio waves travel in the air, they can be picked up by a receiving antenna that is able to convert them back into an electrical signal characterised by the same trend over time of the original signal.

The way in which an electromagnetic signal propagates is similar to the way in which waves in water propagate after an object has been dropped into it, both being wavy-type phenomena and therefore regulated by the same rules of physics. In the case of water, waves gradually diminish their

amplitude moving away from the source until they become minute and invisible to the human eye once a certain distance from the falling point of the object has been exceeded.

Electromagnetic waves are produced by the movement of electrically charged particles, and for this reason, reference is also made to electromagnetic radiation because it radiates from electrically charged particles. All wireless devices use some form of these waves.

It has already been seen that these waves are part of an electromagnetic spectrum that names them appropriately depending on their frequency. Although this spectrum is infinite, the region of radio waves is limited to the range 100 kHz to 300 GHz.

Considering a wave that propagates, the number of times, in the unit of time, that such a wave oscillates, passing from a crest to a trough, can be observed; the resulting number is equal to its frequency, which is expressed in hertz (Hz). In the case of wireless networks, the current operating frequencies are of the order of billions of hertz or gigahertz.

Electromagnetic waves are emitted with a certain power, which is measured in watts (W). The greater the output power, the greater the distance to which these waves can be propagated while maintaining an amplitude such that they can be detected by a receiving device. Of course, there are various multiples of 1,000 W or kW and sub-multiples such as the thousandth of a watt or milliwatt.

In the case of electromagnetic field, given the relative amplitude of the values in play, decibels (db) are often used that represent the logarithmic ratio between a quantity to be measured and a reference quantity. As such, what is used is the decibel referred to the milliwatt (dBm), the direct gain of an antenna referred to an isotropic imaginary antenna (dBi) and the direct gain of an antenna compared to a dipole half wavelength antenna (DBD).

When we talk about bandwidth, this concept is often associated with the performance of networks. However, it is necessary to differentiate between frequency and bandwidth: frequency represents a specific location in the electromagnetic spectrum, while bandwidth represents an interval between two specific frequencies. A single channel of 40 MHz may have the same bandwidth both at a frequency of 2 GHz and at a frequency of 5 GHz, for example.

6.2.1 The propagation of electromagnetic waves

Electromagnetic waves, when propagated in the air, can be affected by many factors that modify their properties, altering the properties of the signal reception itself when predicted theoretically. A typical factor that affects propagation is interference. Interference is a real challenge for the improvement of the quality of reception of the signals, and this has also led many governments to restrict the use of certain frequencies in order to reduce interference itself. In fact, the restriction of the use of frequencies reduces the number of devices that use these frequencies with consequent reduction of mutual interference.

Interference may be caused by many factors, among which are the objects that stand between the transmitting antenna and the receiving antenna. In fact, when an electromagnetic wave hits an object, a part of it reflects and a part of it refracts, passing through. This behaviour is strongly influenced by the material the object is composed of, its shape and size.

With regard to reflection, this occurs when an electromagnetic wave strikes a relatively large surface compared to its wavelength. It can take place on surfaces such as floors, walls and buildings. Once reflection has occurred, radio waves are radiated, usually along directions that are totally different from the original propagation. An example is shown in Figure 6.1.

As can be seen in Figure 6.1, the electromagnetic wave is characterised by its direction of propagation that intersects the object. Once the wave strikes the object, it is reflected along other directions. The action of reflection causes a reduction in its intensity with respect to the incident wave. In most cases, the electromagnetic wave passes through the object, rather than being reflected, and the

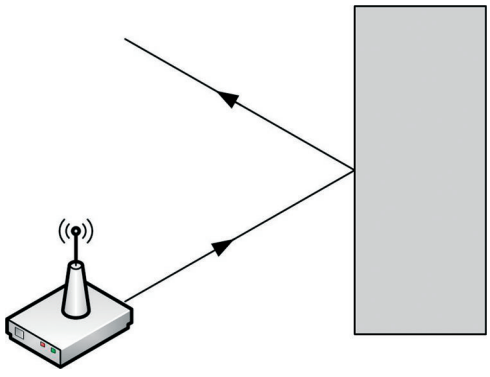


Figure 6.1 Example of reflection of an electromagnetic wave on an object.

intensity of this behaviour depends on the many factors listed above. In any case, all types of interference have a considerable influence on the quality of the signal.

With regard to refraction, this occurs in close conjunction with reflection, as part of the electromagnetic wave, and under appropriate conditions cannot pass through the object. Walls, floors and buildings, which should normally reflect radio frequency, let it pass through without difficulty in many cases, behaving in an unexpected manner. The signal penetrates such objects and decreases its amplitude, however not allowing it to achieve the same distance that it would have been able to achieve in the absence of the same objects. Refraction always occurs at the time of reflection, and the signal is divided between one part, which is reflected, and another part that is refracted. The result is always that the signal, as a whole, is found to be attenuated and deteriorated. An example is shown in Figure 6.2.

The propagation of an electromagnetic wave is also influenced by other factors, such as diffraction, scattering and absorption.

Diffraction is very similar to refraction because it takes place when an electromagnetic wave meets an object on its path of propagation. Diffraction depends heavily on the relationship between the wavelength of the electromagnetic wave and the dimensions of the object. On the basis of this parameter, and under appropriate conditions, the signal can propagate around the object, generating areas of shadow, characterised by absence of a signal behind it. This is because the signal is not able to penetrate the object and can only move around it. An example is shown in Figure 6.3.

While refraction describes the behaviour of the electromagnetic wave that propagates within an object, diffraction describes the behaviour of electromagnetic waves that propagate outside and around the object itself.

As can be seen in Figure 6.3, the shadow does not exist if the signal propagates in the object because of refraction, as the same would be able to penetrate the object and to exit, even if with reduced

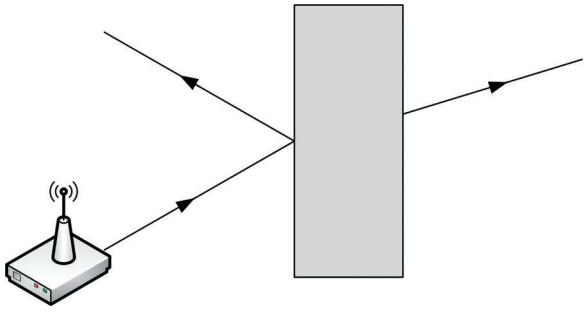


Figure 6.2 Example of refraction of an electromagnetic wave on an object.

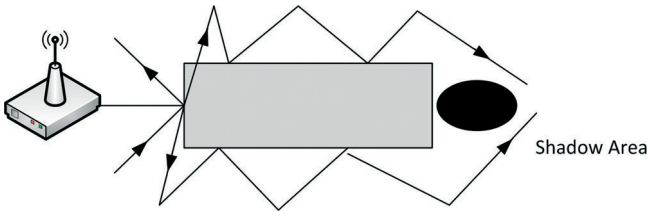


Figure 6.3 Example of diffraction of an electromagnetic wave in relation to an object.

intensity. Often, it can happen that the signal will not be able to penetrate within the object but the effect of diffraction and multiple reflection around it is combined in such a manner that the shadow behind the object does not develop.

With regard to scattering, this takes place when an electromagnetic wave encounters an uneven surface or a set of objects located very close to each other. This situation causes a division of the incident signal in a flood of secondary signals that are reflected in all directions causing a significant reduction in the intensity of the original signal. An example is shown in Figure 6.4.

It is clear that this type of interference may cause significant problems in the reception of the original signal, and this is due to the fact that the receiver itself receives both the original signal and a multitude of secondary signals that have undergone the scattering phenomenon, making the decoding operation by the same receiver difficult.

With regard to absorption, this happens when an electromagnetic wave propagates within objects mostly containing water such as trees and paper. This type of interference considerably disturbs point-to-point or point-to-multipoint connections. In this sense, trees, because of the considerable amount of water they contain, absorb a high quantity of electromagnetic waves, and this effect is more marked in the case of evergreen trees that contain a greater amount of water.

6.2.2 The signal-to-noise ratio

In the case of wireless networks, we have seen how many types of interference can exist; some of them can be avoided or reduced, while many of them are, unfortunately, always present. Interference, which is always present, depends on the movement of the electrons in matter and the manner in which the same emit radiant energy. This means that regardless of the precautions that can be adopted, there is always a residual amount of interference that cannot be deleted. This residual interference represents a base that is always present in any propagation environment. To transmit a wireless signal, you must

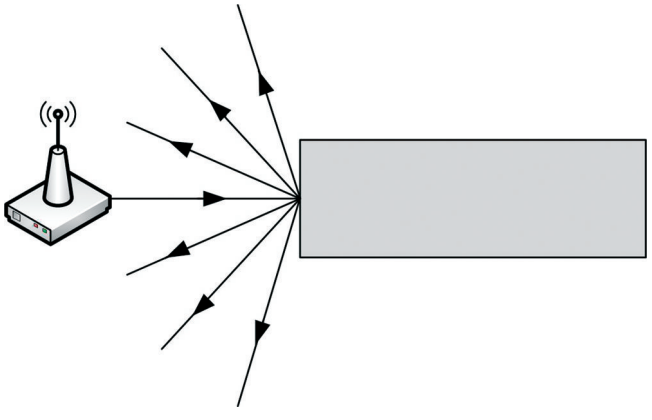


Figure 6.4 Example of scattering of an electromagnetic wave on an object.

always be certain that the signal exceeds the background level; otherwise, the same is undetectable. Noise must also be taken into consideration, inevitably present, which worsens the quality of the received signal.

In this sense, the signal-to-noise ratio or SNR is very useful as it allows the quality of wireless connections to be characterised. The SNR, as we have seen, is calculated by dividing the signal value by the noise value. This relationship is often expressed in dB. An SNR of 3 dB means that the SNR is 2:1, that is the signal is double the noise. This ratio doubles every 3 dB of the value of SNR; this means that if 3 dB corresponds to a ratio of 2:1, 6 dB corresponds to a ratio of 4:1 and so on. Therefore, for every increase of 3 dB, the signal doubles its power. In wireless devices currently being used, the SNR may vary between 12 and 17 dB that correspond to ratios of the order of 20:1 on the reception side and, on average, 80:1 on the transmission side.

6.2.3 The main players that operate on wireless

The main players in the wireless sector are the same players that operate in the world of telecommunications and that have already been discussed.

6.3 Risks and threats in the wireless industry

This section will illustrate and resume the general objectives of information theory and the methods of measuring risk and understanding of the threats.

6.3.1 Objectives of the information theory

The risks and threats present in the wireless industry are more or less the same as for the entire telecommunications sector, with the disadvantage that, in this case, the communications themselves are more vulnerable because the same transit in space and can be intercepted by anyone.

The basic concepts will be illustrated and resumed further in the following.

When the issue of information security is at stake, three fundamental concepts must be taken into consideration:

1. confidentiality;
2. availability;
3. integrity.

These objectives will help to clarify what should generally be protected and the reason for doing this.

Before evaluating any risk in the field, it is important to understand the definition of each of the above objectives.

6.3.1.1 Confidentiality

Attacks against confidentiality of information are related to the theft of or unauthorised access to data. This can happen in many ways, such as the interception of data while it is in transit or simply the theft of devices on which the data are stored. The objective of compromising confidentiality consists of obtaining proprietary information, the use of credentials, disposing of secret, financial, health-related or other type of information.

Attacks on the confidentiality of wireless transmissions are conducted by simple analysis attacks of signals which propagate through space. All the wireless signals that propagate through space are vulnerable to such analysis activities. This means that in the wireless industry, there can be no absolute

confidentiality because it is always possible to receive a signal and analyse it or record it to analyse at a later time. The use of cryptography can help reduce the risk to an acceptable level. It has already been discussed in Chapter 2 that the same can be made suitably secure by operating according to suitable criteria. However, vulnerabilities remain due to human factors, such as the management of keys, which must be properly organised; the failure to do so possibly compromises the entire communication process.

6.3.1.2 Availability

Availability allows a legitimate user to access confidential information after it has been properly authenticated. When availability is compromised, access may even be denied to legitimate users due to malicious activity such as denial-of-service (DoS) attacks.

Reception of radio frequency signals is not always possible, particularly if there is a subject that wants to avoid this. In this sense, the use of jammers is a significant problem, which is of considerable concern to national governments.

6.3.1.3 Integrity

Integrity implies the unauthorised modification of information. This could also mean changes to information while the same is in transit in space or while it is being stored on some type of support. To protect the integrity of information, efficient techniques for validation must be implemented. These techniques can be integrity checks or digital signatures.

Wireless networks operate in a manner free from manipulation of systems. If integrity is not taken into due consideration, it is always possible for an attacker to alter data during transmission. This can fool the recipient into thinking they are involved in an exchange of confidential data, while in reality, exactly the opposite is happening. Wireless networks have had to adapt, with time, to this type of threat by pushing for the creation of new increasingly secure and more efficient standards.

6.3.2 Analysis

Analysis involves seeing, recording or intercepting a signal against the will of third parties. Unfortunately, all radio frequency signals are subject to interception, and this is due to the fact that signals propagate freely in space. These circumstances allow anyone within a radius of propagation of the signal to intercept the same. One of the few available techniques to prevent the loss of confidentiality is the use of cryptography. If a signal uses encryption, confidentiality can be guaranteed until the type of encryption used is unassailable. Risk analysis on a radio frequency signal represents an inherent risk of the means used that cannot be eliminated. The only way to reduce this risk is the use of an efficient system of confidentiality control.

6.3.3 Spoofing

Spoofing, as has already been discussed in Chapter 5, is the act of mystification, which consists of impersonating a customer, a user or a device that is authorised to have access to a resource that is protected by some form of authentication or authorisation. When spoofing occurs on wireless networks, it implies that an attacker is able to produce an access point to obtain authentication information to be used straight away or subsequently. Another way to perform a spoofing attack is to conduct a man-in-the-middle attack. In this case, as has already been seen, the attacker is positioned between the user and the network. This attack can be conducted by deceiving a valid access point or diverting a session. Once this type of attack has been successfully conducted, the attacker can use the

authentication information acquired from the legitimate user to send it to the network as if such a request were originating from the actual legitimate user.

6.3.4 Denial-of-service

DoS represents, as has already been discussed in Chapter 5, the effect that is obtained as a result of an attack aimed at making a device or network unable to communicate. Attackers look for and discover continuously that sending appropriate packets makes a network unable to respond, restart or even close down. In the case of restart, the attacker can attempt to move traffic passing through a given device during the same restart phase. In fact, during this phase, the attacker hopes that the device itself has not yet uploaded all the procedures, rules and the information that will be used to render the same secure.

DoS attacks in respect of a wireless network can be simply conducted with small portable jammers because the maximum power output from a wireless device is relatively reduced both due to the national laws under which the device itself operates and to avoid reaching excessively high emission levels, which could endanger the health of the users that use it. This means that it is not particularly complex to cover the signal emitted from a wireless device by resorting to the use of a suitable jammer.

Another type of DoS attack that can be conducted against wireless networks is due to the relative scarcity of management frames that are used by the network itself. These frames allow anyone who is able to analyse a wireless signal to perform a DoS attack by replicating suitably certain management frames. In general, these types of attack attempt to deceive traffic management informing the device of the authorised user that the same cannot remain connected to the network.

6.3.5 Malicious codes

Malicious codes, as has already been discussed in Chapter 5, are able to infect and corrupt network devices and may take the form of viruses, worms, Trojan horses, etc.

The following sections provide a summary of malicious codes.

Viruses infect the devices and are not, in general, able to replicate and exit the devices themselves alone. Once a device has been infected by a virus, the same virus can only replicate within the device itself. This means that all the risks that arise from viruses only consist of the infection of the device that has received it.

Worms, unlike viruses, are able to regenerate and propagate outside the infected device, involving an entire organisation or propagating even within the Internet.

Trojan horses are programs that contain malicious codes that are activated the moment the programs themselves are installed or run.

Malicious codes relating to wireless devices are new threats that may affect many new types of wireless devices, such as mobile phones, handhelds and laptops, and a relatively recent threat that has been growing in intensity with the expansion of wireless networks.

The other form of malicious code related to wireless devices is spam. Although spam is not destructive, organisations are generally forced to invest a considerable amount of time and money to combat it, making the effects very similar to that of malicious codes. The issue of spam has created a security market characterised by its products, solutions and services developed for countering this threat.

6.3.6 Social engineering

Social engineering is often called low technology hacking. It implies that someone is able to use the weaknesses of people or the security policies of organisations to access certain resources. Some of the most famous attacks on computers and networks have been conducted using this type of approach to

capture confidential information. The real risk is represented by the skill level possessed, and there are a number of actions that can be performed to prevent this type of attack.

6.3.7 Rogue access points

Rogue access points represent a significant threat for every organisation. If organisations do not organise themselves at the appropriate time, employees themselves may be tempted to activate their terminals in wireless mode, opening a significant fault in the security of the internal network as the same, not being fully conversant with the security mechanisms, can leave the access points open, enabling anyone to enter the same network.

6.3.8 Security of cellular telephony

This section will explain the general problems of security of cellular telephony with reference to the information below for due insights.

Mobile phones have considerable advantages with respect to the other wireless communication systems from the point of view of security.

It has already been seen that cell phones use radio frequency transmissions on two separate channels:

1. channel for voice communication;
2. channel for the transmission of control signals.

Control signals are used to identify and authenticate a mobile phone within a radio-mobile cell through the transmission of suitable identification codes by the phone itself. When the radio base transceiver station receives such codes, it evaluates whether the applicant is a legitimate user, by comparing the above-mentioned codes with those contained in an internal archive. Once the cellular telephony supplier verifies that the received codes belong to the legitimate user, the latter is granted access to the network.

Like all wireless devices, cell phones are subject to interception and spoofing. In the field of cellular telephony, such activities are called call control and cloning. Another risk associated with mobile phones is the possibility of re-programming phones, transforming them into microphones for interception capable of capturing voices and sounds in a given environment and sending them to any remote location.

Call control is an action that is easy to conduct, particularly for phones that use analogue technology (which is now virtually disused everywhere). This is due to the fact that analogue technology uses plaintext voice transmission using modulation frequency.

With the advancement of digital technology, this problem has been reduced to a minimum, if not eliminated altogether, because within the same manager of digital telephony, calls are handled in a manner that is entirely digital and secure. Problems arise if the mobile phone links into roaming on another carrier. In fact, in many cases, if the member operator and hosting operator do not use the same digital technology, calls are transferred from one operator to another in an analogous manner and in plaintext, posing significant risks from the point of view of security.

Another problem is, as has just been stated, the possibility of using the cell phone as a remote microphone or microspy. In this case, anyone with the right technical knowledge can send a maintenance request signal on the control channel of the cell phone itself, placing the phone in diagnostic mode. In this mode, all the conversations that are captured by the phone are sent on the voice channel allowing remote listening, and this happens without any indication of activity on the cell phone display. The only way to be aware of this activity consists of trying to make a call; in this case, the phone is not able to call itself until it exits diagnostic mode. There are two ways to make it exit

diagnostic mode, either remotely by sending an appropriate command or by turning the phone off and on again.

Regarding the issue of cloning, it can be tackled in different ways with increasing difficulty with the progress of cellular communication technologies.

6.3.9 Hacking and hackers in the wireless industry

It is said that radio frequency transmission is characterised by inherent risks, such as interception, disturbance of signals (jamming) and spoofing of signals. As radiofrequency travels in space, interception of the signal is a relatively simple operation to perform if you have the right equipment. In this sense, spectrum analysers can detect radio transmissions by showing the frequency of the signal. Depending on the frequency, an attacker may be able to identify the transmission that they are interested in intercepting. Most of the radio frequencies of the electromagnetic spectrum are reserved for specific uses. Once an attacker is able to find a signal and map it within a reserved spectrum, he is able to know who is transmitting and in some cases also the reason for transmission.

Looking more closely at the field of wireless networks, it can be seen how they have the same risk of use of radio frequencies in addition to the specific risks of the sector. In a manner similar to mobile phones, the greater the number of persons who use them, the greater the number of persons who spend time understanding how they work and how it is possible to reduce the security defences they contain. Because they are now used by a large number of people, not all having adequate technical knowledge and security awareness, they are activated, in most cases, without taking into account the security parameters of the same, making the networks themselves vulnerable to attacks from outside.

6.3.9.1 The motivation of wireless hackers

When committing a crime, in most cases, there is always a reason. In the case of the wireless industry, in many situations, the reason for violating a wireless network is the need to have access to the Internet free of charge, both for harmless reasons, such as sending a simple email, and for more serious reasons, such as the exchange of information for the execution of a criminal act, or even worse, a terrorist act. Hence, it can be inferred how important it is to apply security to a radio frequency signal. The perfect knowledge of possible attackers can help in the identification of correct actions for reducing the risk that can be implemented.

6.3.9.2 War driver

When it was realised that a majority of people were using wireless devices or access points without activating any protection mechanism, many agents took action to exploit these security flaws.

This new trend to identify and catalogue, according to the level of security, wireless networks currently present was called *war driving*.

War drivers use equipment and software to identify wireless networks and their access guards. In most cases, once wireless networks are identified, located and catalogued, information is exchanged via the Internet with other players that perform the same activities. This information is enriched with many geographical coordinates obtained by a global positioning system (GPS) receiver to ensure being able to locate, without difficulty, the networks found. In this way, anyone can download updated maps of wireless non-secure networks from the Internet in their home town, get close to them and use them for purposes that are not particularly lawful, passing the relative responsibility for what has taken place to the legitimate owner of the wireless network.

Subjects that perform this activity are not only young people but also older people driven by more disparate motivations.

6.3.9.3 War walker

In *war walker*, the same activity is performed as *war driver* only that, in this case, movement is on foot instead of by car. In this case, the person who performs this activity takes with him, on foot, all the necessary equipment to find and identify non-secure networks. This activity is mostly performed by people who do not possess a car, represented, for the most part, by young persons.

6.3.9.4 War chalking

War chalking is closely linked to the two search and identification activities performed by car or on foot. In this case, the results obtained, are not published on any site on the Internet, but a suitable symbology was developed involving symbols being written directly on the walls of buildings where non-secure networks have been identified. The symbols commonly used are shown in Figure 6.5.

6.3.9.5 War flying

In *war flying*, search activity and identification is performed using a small air plane flying at relatively low altitudes, the same being able to move at relatively high speeds and thus to cover considerable distances in a short span of time. This activity is in any case expensive because it is linked to the use of an air plane.

6.3.9.6 Bluejacking

Bluejacking is a relatively new term involving the hunt for devices based on Bluetooth, a wireless communication technology explained in Chapter 1.

It consists of sending anonymous messages to devices that have the Bluetooth port enabled.

A person who wants to perform *Bluejacking* usually heads for a very busy place, such as an underground train station, and analyses such places in the search for devices that have Bluetooth enabled. Once a device is identified, the subject sends an anonymous message with more disparate purposes, which, however, can cause a sense of discomfort in receiving because personal information, such as the clothes that you are wearing and so on, may be contained.

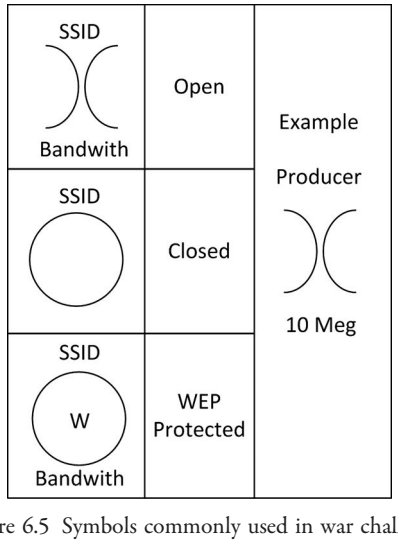


Figure 6.5 Symbols commonly used in war chalking.

6.3.9.7 X10 driving

In *X10 driving*, subjects move with a scanner for X10 wireless cameras, which are normally used by people for various reasons. Because these cameras are low cost, the plaintext signal emitted from the latter can be received without difficulty from one scanner characterised by relatively low cost. In this way, the privacy of individuals is heavily violated.

6.3.9.8 Cordless phone driving

Cordless *phone driving* is an activity to find wireless phones that are normally used within the home or office. This activity was relatively frequent and easy to perform when using wireless analogue phones and operating at the frequency of a few hundred megahertz. In this case, it was possible to intercept calls currently in progress and even connect with the telephone base to make phone calls.

A new generation of phones was subsequently created, which used encrypted digital technology, mostly operating through expanded spectrum modulation at a frequency of about 2.5 and 5 GHz.

6.3.9.9 War dialing

The terms *war driving*, *war chalking* and *war walking* are new expressions and derive from an old hacker term *war dialing*. The subjects that performed *war dialing* tried telephone numbers with their own computers to find a number to which a modem replied, exploiting the fact that these modems were usually connected to computers or networks characterised by relatively low or even non-existent levels of security, and, in this way, attackers could enter the systems. This designation has been extended to the current wireless networks and attackers, in this case, move with various means to search for networks that could be violated. The terms *war dialing* and *war driving* are substantially similar as they describe a search activity for non-secure networks to crack.

6.3.9.10 Tracking of war drivers

In any case, *war drivers* leave traces on networks that they have attempted to attack and these traces can be used to try to reconstruct the identity of the persons who have attempted to perform the attack.

This identification activity is performed by specialist investigators who possess all the necessary technical skills.

In the first instance, the investigators go to the site and acquire all the information that is stored in the devices and servers that have been attacked. Subsequently, this information is analysed in adequately equipped technology laboratories. Once the information sought is found, it is possible to determine the message authentication Code (MAC) address of the network card used by the attacker to violate the wireless network. This MAC address is unique for each card in the world and is inserted into the card by the manufacturer at the time of production. Certain particularly expert attackers are able to alter this MAC address, but the majority are not able to perform this operation. Once the MAC address has been acquired, investigators can check, at the locations of possible suspects, if the same possess a device characterised by the same MAC address found on the network that has been violated. In the most critical cases, such as those involving terrorist acts, investigators can contact the manufacturer to try to trace the movement of the card by the manufacturer, including the distributor and end customer. Such activity, although easy to perform, however, requires a relatively long period of time to be performed. Once the distributor has been reached, it is necessary to go to the customer that has purchased the incriminating card. The first task that can be done is to identify all the customers that have purchased the network card model sought in the hope that they have left their traces in purchase of the same, such as the use of a credit card or any type of purchase transaction that did not require the use of cash. If the transaction took place in cash, as performed by an attacker who does not

want to be identified, it is always possible to use the video surveillance system, if present, to examine the images relating to the date and time when the purchases of the same incriminating network card model were performed.

It should be remembered that all network devices, as they emit radio frequency electromagnetic waves, can be plotted without great difficulty.

6.3.10 Radio frequency identification

Radio frequency identification (RFID) is a promising technology at the growth and development stage, which was created to trace products from the time of manufacture to the time of leaving the factory until the time they are purchased by the end customer. It is clear that this technology, on the one hand, is very useful for the purposes of audit and, on the other hand, poses significant problems with regard to the protection of the privacy of individuals.

An RFID system consists of a suitable label containing a coil-wound antenna and all the components necessary for communication. It is attached to the product that will be traced. This label, also called tag, on receiving energy from an external reader, sends back the information contained therein. A tag can also be read, without any difficulty, by readers of different manufacturers. Tags can also be rewritten, without any problem, by other producers that acquire the product and want to enter the appropriate information in the same tag.

In any case, the majority of RFID systems used are of single write type, in the sense that the information contained within them can be written only once, typically during production, and can never be cancelled.

RFID have a variety of applications in all areas, from tracking to implementation of product inventories.

Using radio frequency electromagnetic waves, they are not immune from the risks of the products or devices that operate with the same mode.

Moreover, they have, as has already been stated, significant risks in terms of the privacy of those using them, precisely because of their traceability.

6.4 Wireless technologies in the physical layer

The physical layer of wireless technologies has been widely explained in Chapter 1. Various methodologies are used, all characterised by greater efficiency and security as new standards were gradually produced. It is also been seen that most of the standards were produced by the Institute of Electrical and Electronics Engineers (IEEE) under the name of 802.11(x) and all with the intention of ensuring the best ability to send radio waves as far away as possible and with the best possible quality.

Most of the standards used employ spread spectrum technology at a physical level, in all its variants that have been discussed previously. It can also be seen that spread spectrum technology uses a wider frequency band than that needed to send the desired signal, ensuring greater immunity to external interference and increased security. In fact, a signal that uses spread spectrum modulation has features that are very similar to background noise, and therefore can be detected, intercepted or disturbed only with great difficulty. Spread spectrum modulation was first used in the military field. This technology spreads a signal on a relatively broad bandwidth by spreading its content within the same band, unlike narrowband modulation that transmits a signal peak on a well-defined frequency. The situation indicated above is shown schematically in Figure 6.6.

The military used modulation spectrum because it made it possible to transmit a signal over a relatively broad bandwidth, making it extremely difficult for potential eavesdroppers to detect the same signal as it was indistinguishable from the background noise by which it was covered. For this reason,

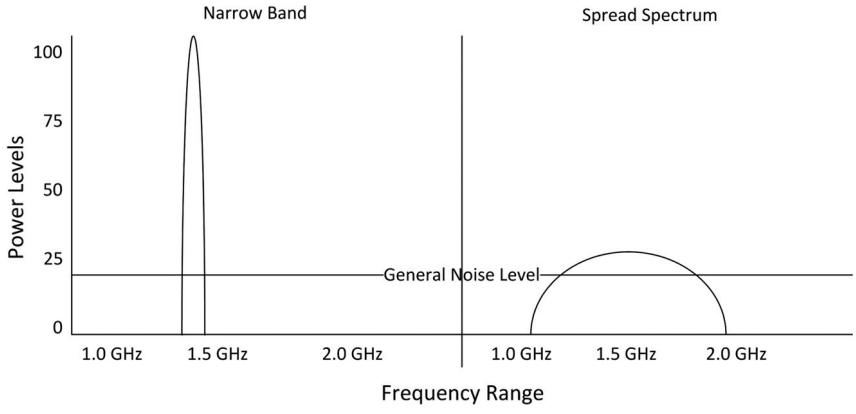


Figure 6.6 Examples of signal spectra for narrowband modulation (a) and spread spectrum modulation (b).

spread spectrum modulation was kept confidential until the 80s when the US Federal Communications Commission (FCC) implemented certain rules to make this technology available to the general public. The release of this technology was designed to encourage research and development, with the aim of leading to the discovery of techniques for more secure wireless transmission. This activity reached its climax with the publication of the family of 802.11 (x) standards that use spread spectrum technology intensively.

FCC has always tried to keep most of the frequencies available, releasing the minimum required radio frequency for civilian communications. The use of narrowband technology allows preservation of a large amount of bandwidth and permits numerous broadcasting stations to use a range of frequencies that are relatively limited. Moreover, narrowband technology requires considerable emission power of the signal to enable it to be above the background noise at the desired reception distance, allowing its detection. Because of the limited use of frequencies, this signal can be easily disturbed and is very sensitive to environmental or voluntary interference. If spread spectrum technology is being used, the risks mentioned above are relatively low. Hence, spread spectrum technology was seen by the military world as a robust and reliable means of communication.

6.4.1 The industrial, scientific and medical band

In 1985, FCC changed part of the adjustment of the radio spectrum, allowing devices and wireless networks to operate in the so-called industrial, scientific and medical (ISM), band. This allowed the development of the wireless market being able to rely on dedicated bands that enabled transmission. Usually, to operate in bands of the electromagnetic spectrum, authorisation is required from government bodies; these available bands vary from nation to nation. Instead, ISM bands are bands that are freely available throughout the world to ensure the interoperability of wireless products at an international level, and these bands coincide in most countries. The only restriction is the maximum power that can be emitted by the devices, which is relatively limited, reducing the radius of action of the majority of devices themselves to a few tens of meters.

The ISM spectrum can be used by anyone as long as the devices used comply with certain guidelines agreed at an international level.

6.4.2 Modulation techniques used

The modulation techniques most often used are frequency hopping spread spectrum (FHSS), direct sequence spread spectrum (DSSS) and orthogonal frequency division multiplexing (OFDM).

These techniques have already been described in Chapter 1 and will not be further discussed in this chapter.

6.5 Frame management in the wireless industry

The management of frames at the level of data link layer is of vital importance for the proper operation of wireless networks. This has already been addressed in terms of basic concepts in Chapter 1, and in the following will be further analysed, given its importance.

The data link layer is responsible for managing the information to be transmitted by dividing it, appropriately, into frames. These frames are sent at the physical level to be transmitted in the form of radio frequency electromagnetic waves from the antenna of the wireless device.

Very often, it is thought that there is a certain similarity between Ethernet technology at the level of wired network and Ethernet at the level of wireless technology. This is not true because there is a difference in the way the information is handled at the MAC level. In wired Ethernet, the maximum frame size is 1,518 bytes, while in wireless Ethernet, the maximum frame size is 2,346 bytes. Additionally, there are also other differences between the wired version and the wireless version.

There are basically three types of frames used in 8012.11 a/b/g standards. They are control, management and data frames. Control frames organise the data flow; management frames allow users to turn the network on or off while data frames contain the actual data. Figure 6.7 shows the general structure of a frame.

If one analyses the control section of frames, it can be seen how the 2 bytes dedicated to this activity contain a high amount of information. The parameters used are:. . .

1. *Protocol version*, which identifies the frame as a wireless frame. This field may only use one parameter.
2. *Type*, which serves to identify the frame as control frame, management frame or data frame.
3. *Subtype*, which serves to identify itself as one of the different types of control, management or data.
4. *To Distribution System (DS)*, which indicates that the frame is intended for a distribution system that identifies a wireless network.
5. *From DS*, which indicates that the frame comes from a distribution system.
6. *More fragments*, which indicates that there are more incoming frames. This means that several fragments or parts of this fragment follow the present fragment.
7. *Retry*, which indicates to the receiver that the frame has just been sent and that the same has been informed of its retransmission.
8. *Power management*, which indicates that the device is in energy-saving mode. Access points are never in power-saving mode, and this value, in their case, is zero. Mobile devices, on the contrary, can go into energy-saving mode and use this frame as an indicator.
9. *More data*, which informs the mobile device that the access point has received the multiple data frames and that it has begun to fill its memory buffer with these frames.
10. *Wired equivalent privacy* (WEP), this field is used to indicate that the WEP encryption is/is not

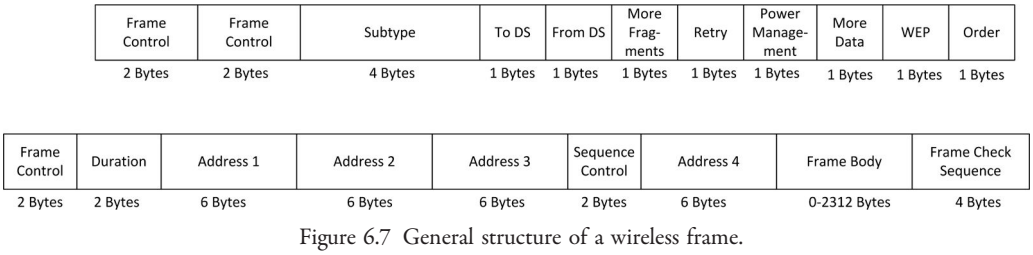


Figure 6.7 General structure of a wireless frame.

currently being used. This encryption will be discussed in more detail in the following.
11. *Order*, this field is used in the management of the quality of service.

6.5.1 Beacon

The access points, often, identify themselves and their operating configuration by issuing the so-called beacon frames. Devices, instead, send beacons only when operating in ad hoc mode. These beacons contain the data rates supported by access points, the channel on which they converse and any WEP support. Depending on the setting, access points can also emit, in broadcast mode, the service set identifier (SSID) (which is discussed later).

Beacon frames are used by wireless devices to understand who can provide the service in a given area.

6.5.2 Probe request

Probe requests are very similar to beacons, in the sense that both are used for identification and configuration information. Unlike beacons that transmit their identification in space to each device that is in their coverage area, the probe is used by wireless clients to locate an access point with the same identification settings. If the client's probe presents the same correct SSID, then the two devices decide if their settings allow them to start the authentication process. This probe request is characterised by two parts:

1. SSID;
2. supported data rates.

Probe requests are used to find the access points that do not send beacons. Because the access points themselves can operate in this mode, there must necessarily exist a mechanism to find the latter, and the probe request was designed for this purpose.

6.5.3 Probe response

An access point that receives a probe request responds to such a request. It only responds if the SSID corresponds. The probe response is used to identify the configuration parameters of the access point. The probe response provides the following parameters:

1. data rates;
2. SSID;
3. information on the channel.

This package is similar to beacons with the difference that it is sent if a given client sends a probe request that corresponds to the id information of the access point.

6.5.4 Authentication

The authentication process depends on the configuration and the level of security that has been set. Authentication must take place before connection to the wireless network. This frame is characterised by an identification number designed to identify and authorise a wireless device. An authentication frame is characterised by the following four parts:

1. Authentication algorithm, which represents a part of the frame that is used to indicate which security configuration is being used.
2. Transaction authentication number that identifies which frames are part of a given authentication

transaction. It is used when an access point is providing multiple authentication services.
3. Status code, which is used to identify the reason why the transaction has failed or has been successful.
4. Challenge text, which is a support for security and represents a packet of plaintext that is sent for security reasons.

6.5.5 Association request

Once a wireless device has been successfully authenticated, it can begin a process to access the network called association. In this situation, the wireless client must send the device's configuration information in order to enter the access point. The access point controls such information and takes relevant decisions. The outcome of these decisions does not depend on the outcome of the authentication because the latter has already taken place. It only checks that the configuration parameters are the same for both parties who wish to communicate. These configuration parameters are SSID, channel and power.

6.5.6 Association response

The association response, generated from the access point, communicates to the wireless device that the same was connected to the network. The association response frame is closely related to the association request frame. The only difference between the two is an identification attribute in the frame. Once the wireless device receives the frame, it knows that it has been connected successfully to the network and that it is ready to use network resources.

6.5.7 Disassociation and de-authentication

These two frames are the same, the only difference being the code. This code is included within the body of the frame and is used to identify the reason why the wireless device was excluded from the network. This code is characterised by the length of 2 bytes and is part of a frame. This code is capable of expressing 50,000 exclusion reasons even if, in practice, only 10 reasons are provided.

Both frames are used to expel a wireless device from the network. The main difference between the two frames lies in why a device has been expelled and which part led to the decision to expel the device. The disassociation message is used by access points to expel wireless devices that have not communicated for a certain period of time. This allows the access point to clean up the status of existing connections and to remove old clients that are disconnected without warning. The de-authentication frame is sent by the administrator to expel a wireless device from the network.

6.5.8 Carrier sense multiple access/collision avoidance

The carrier sense multiple access/collision avoidance (CSMA/CA) is a technology that has already been discussed in Chapter 1 and that operates at the data link level. It is very similar to the Ethernet CSMA/CD. However, there is a difference between the two. In fact, in wired networks, the loss of a certain amount of bandwidth in order to increase the transmission speed of network frames is considered an acceptable compromise, and therefore there is a limit on detecting collisions between frames to send them again at a later time. In a wireless network, the loss of bandwidth over time to detect the event of a collision is not considered to be acceptable, and for this reason a mechanism is used that attempts to avoid collisions and is shown below in detail, together with the four critical elements used in this mechanism.



Figure 6.8 RTS frame structure.

6.5.8.1 Request to Send

The first step of the process is the initial request. It is called request to send (RTS). The client begins RTS when he/she needs to send data to the access point or to another network resource via the access point. The client quantifies the amount of time needed to complete the transmission as part of the initial request. This is indicated with the bit named network allocation value (NAV). The access point uses that information to decide the time during which the client can communicate before re-requesting permission. The frame that is used in this step is shown in Figure 6.8.

As can be seen in Figure 6.8, a frame of 20 bytes is divided into five sections. First, there is the control section of the frame that identifies the frame as RTS. Then, there is the section concerning the duration that communicates to the access point the amount of time that is needed to perform the communication. Next, there is the section on the receiver's address that contains the MAC address of the transmitting station. Then, there is the address of the sender, which also contains a MAC address. The final part contains a control sequence of the frame itself and represents the wireless version of a similar control that is performed in Ethernet. Because the wireless represents a less reliable means with respect to the cable and therefore with the Ethernet standard that operates on it, a different technique is used to send a control sequence or checksum. This technique works by sending a checksum for the previous frame.

6.5.8.2 Clear to send frame

Subsequently, the access point sends out a clear to send frame (CTS). This frame is used to communicate to the client how much time can pass before having to close the CSMA/CA process. The structure of this frame is shown in Figure 6.9.

As can be seen in Figure 6.9, the RTS frame is divided into four sections. First, there is the control section, which announces the frame as CTS type. Then, there is a segment of the duration that is used to indicate the time granted to the wireless device to transmit data. Next, there is the MAC address of the receiver that has been copied from the transmitting address of the previous frame and, finally, there is the frame control sequence.

6.5.8.3 Data

The data section has nothing to do with handling of the same in itself, but it is necessary to understand fully the sequence of tasks that is performed when a wireless device transfers data. Data frames are sent until all the time ranges contained in CTS are exhausted. Once this is happened, data are no longer transmitted until a new CTS frame is received that communicates a new time range allowed for

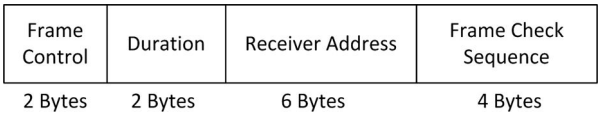


Figure 6.9 CTS frame structure.

Frame Control	Duration	Destination Address	BSSID	Source Address	Sequence Control	Payload	Frame Check Sequence
2 Bytes	2 Bytes	6 Bytes	6 Bytes	6 Bytes	2 Bytes	0-2312 Bytes	4 Bytes

Figure 6.10 Structure of a data frame.

transmission. To receive a new CTS, it is necessary to initialise the process again by sending an RTS. The structure of a data frame is shown in Figure 6.10.

As can be seen in Figure 6.10, the structure of a data frame is different from the structure of an RTS or CTS frame because the latter are management frames.

6.5.8.4 Acknowledgement

Acknowledgement frames are sent from different network protocols to ensure that the data received by the receiver have not been corrupted during the transmission. This frame is characterised by the same structure as that of RTS or CTS frames. The structure of this frame is shown in Figure 6.11.

As can be seen in Figure 6.11, the acknowledgement frame is characterised by an initial section that serves to communicate the nature of this frame. Then, there is a part of the duration that is used to inform the access point if the acknowledgement in question is the last of the current data transfer. Next, there is a section on the address of the receiver, which is copied from the MAC of the transmitter. Finally, there is a control section.

6.5.9 Fragmentation

Fragmentation is very widespread and used in wireless networks because of the considerable interference present in this type of transmission. Fragmentation is used to enable the wireless means to send more data. This is accomplished through the use of transmissions that are mostly characterised by smaller sizes. Because wireless is an extremely unstable means, sending very large frames and subsequent re-sending in the event of disturbance during transmission involves a considerable amount of time with respect to the situation where smaller volume frames are sent. In this sense, fragmentation is used to speed up the transmission. It is used to categorise the information to be sent into smaller parts that can be transmitted more quickly.

Fragmentation can cause problems as it makes the task of dividing the data and its reassembly once it has reached its destination more laborious. Fragmentation adjustment is a somewhat tricky option to achieve in that it must balance the problems generated by excessive fragmentation with those problems generated by reduced fragmentation.

In most cases, the same producers pre-set their products to provide the best performance, while still leaving the end user the option of adjustment depending on the operating conditions present in the environment where these products are to operate.

6.5.10 Distributed coordination function

Distributed coordination function (DCF) is used to query the means in order to check if the same is being used by someone else. In fact, before sending any frame, the network must be checked to ensure it

Frame Control	Duration	Receiver Address	Frame Check Sequence
2 Bytes	2 Bytes	6 Bytes	4 Bytes

Figure 6.11 Acknowledgement frame structure.

is not being used by someone else. The DCF performs this task by examining all the frames' MAC headers that are sent. When another device intends to use the network, it requests access, announcing the time it needs for the same. When it does so, its frames are marked with this indication that is called NAV.

6.5.11 Point coordination function

Point coordination function (PCF) has the same objectives as the DCF even if it is characterised by certain differences. In PCF, the device alerts the access point that it is able to respond to queries. This means that the PCF can operate within a wireless network controlled by an access point. Once the access point knows that the device can support queries, it interrogates the entire portion of the network, asking the various clients if they need to send frames. This allows the devices that need to perform data transfer to do so. The PCF must operate in a coordinated manner with the DCF.

6.5.12 Interframe spacing

To perform checking functions and to allow time synchronisation, there must be a time interval during which no communication occurs. This interval is called interframe spacing. This value is provided in all the wireless devices as part of the 802.11 standard. This parameter allows access points to organise and manage the priorities of the network traffic.

There are four types of interframe spacings:

1. short interframe space (SIFS);
2. point coordination function interframe spacing (PIFS);
3. distributed coordination function interframe spacing (DIFS);
4. extended interframe spacing (EIFS).

6.5.13 Service set identifier

The service set identifier (SSID) does not contain any form of security. SSID is mainly used for the identification of the network, as the name suggests. When a client device connects to a network, it must have a reference that allows him or her to know what network to connect. When wireless standards were created by the IEEE, the same was provided for several wireless networks that could operate in the same space, and in order to differentiate one network from anther, SSID was implemented. With the intensive use of wireless networks, this identification has become a necessity.

SSID may also be used to create multiple virtual wireless networks that are very similar to Virtual Local Area Network (VLANs) used intensively in the field of wired networks. When multiple virtual networks are used, every device shares the same propagation space of radio frequency electromagnetic waves, although each communicates on its own network. This mode is often used to accommodate guests within the network and allows different levels of security. This possibility is very useful in situations where more dated wireless devices are not capable of supporting the advanced security services provided in newer devices. In this case, two networks can be created:

1. a network characterised by advanced security services able to operate anywhere on the wired network;
2. a network characterised by a low level of security, capable of supporting older and weaker security services.

6.6 Local wireless networks and personal wireless networks

This section discusses some general concepts related to wireless networks that have already been discussed in Chapter 1.

Wireless networks were initially characterised by a relatively low *data rate* compared to wired networks with proprietary installations and were used for mobile applications.

Today's wireless networks are used intensively in any context: industry, commercial, health, sales, education, etc. This growth is due to the fact that wireless networks allow true mobility without having to depend on the restrictions imposed by cables. This degree of freedom allows being able to work without any problem, and in motion, in any environment where there is wireless coverage. These advantages have given a considerable boost to the market sector that has experienced and continues to experience remarkable growth.

Flexibility is a key element that characterises wireless networks and has extensively spread their use. Consider, for instance, historic buildings; in this context, it is very complex to lay the new cables needed to strengthen an existing data network. With wireless, on the contrary, it is not necessary to lay new cables, and all that is needed is installation of an access point, and the network is automatically strengthened. Consider also, buildings that are not owned, where it is very difficult, if not impossible, to install new cables or drill holes without the authorisation of the legitimate owners. Another very important area is that of civil protection, where in a very short space of time, in the event of a disaster, you would need to install a network of communication where there are no longer permanent infrastructures.

In relatively recent times, even phone companies have become involved in the market, installing access points in areas where there is potential traffic such as airports, train stations, hotels, bars, museums, etc.

Wireless networks can operate in two main ways, ad hoc mode and infrastructure mode, which are discussed in the following sections.

6.6.1 Ad hoc mode

In ad hoc mode, also called independent basic service set (IBSS), wireless devices communicate in peer-to-peer mode, having no need for an access point that operates as an intermediary. This mode is very useful when the number of users is relatively small. In this situation, the beacon is emitted directly from wireless devices. These beacons contain a timer synchronisation function (TSF) that ensures correct synchronisation of the devices. This synchronisation function is usually performed by the access point. Figure 6.12 shows an example of a wireless network that operates in ad hoc mode.

As can be seen in Figure 6.12, there are no access points. This mode has been provided to allow direct communication between two wireless devices without resorting to the use of third-party devices.

6.6.2 Infrastructure mode

Infrastructure mode is the most widespread mode in use of wireless networks. It is also called extended basic service set (EBSS). In EBSS mode, access points control all the traffic. Use of this network requires that an access point is connected to a fixed network for the forwarding of wireless traffic. To

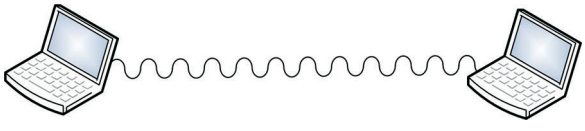


Figure 6.12 Example of a wireless network that operates in ad hoc mode.

operate within a wireless network, the mobile device must be equipped with an appropriate wireless network card. This network is equipped with a small antenna, generally integrated, for the transmission and reception of radio waves. This operating mode is shown in Figure 6.13.

The above described mode allows a group of users that are within the range of the access point to connect with the fixed network to which the access point is connected.

6.6.3 Bridging

Wireless networks can also be used to extend existing fixed networks by connecting through radio frequency, which operates as a bridge (hence the name bridging) between two distant sites containing fixed networks. This situation is shown in Figure 6.14.

Bridging can be used both to connect two fixed campus networks and to provide service on the last mile, completing the Internet service provider (ISP) service.

When you want to connect two remote sites, rather than resorting to the installation of a dedicated connection, or renting the same from a service provider, if the two sites are in line of sight, you can use a wireless bridge. In any case, even if the two sites are not in sight, it is possible to use more complex technology, involving greater cost, and to use satellite bridges to connect two sites that are located in any part of the world.

In the case of the last mile, it is sufficient to install a small antenna on the roof of your building that focuses on the closest ISP antenna to provide a wireless bridge, avoiding the limitations imposed by cable (relatively limited bandwidth, installation of the same, etc.).

6.6.4 Repeater

A repeater is an access point that uses radio frequency to send the signal to an access point connected to the fixed network, increasing the overall coverage of the wireless network, being able to reach users that are outside the coverage of the access point connected to the fixed network. An example is shown in Figure 6.15.

A disadvantage is that the repeater and devices use the same limited bandwidth, reducing the overall performance of the network.

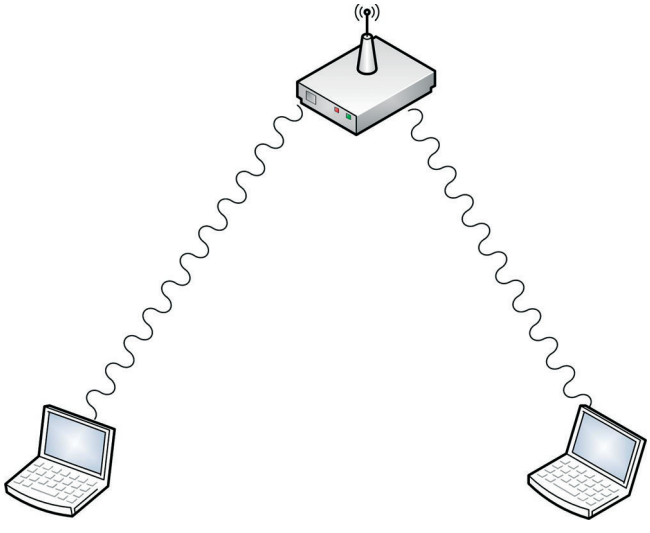


Figure 6.13 Example of a wireless network that operates in infrastructure mode.

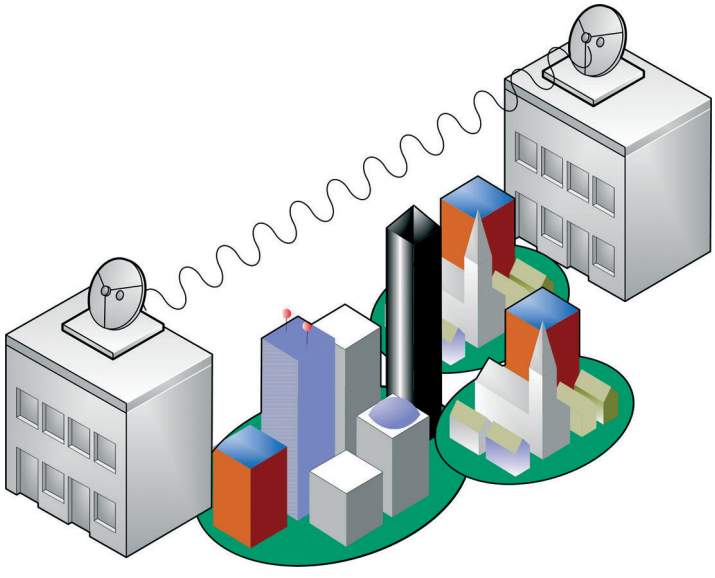


Figure 6.14 Example of wireless bridging.

6.6.5 Mesh networks

Mesh networks are networks in which the individual nodes that compose them operate as routers in fixed networks, always ensuring the connection paths to all connected users. Their great advantage is that not all access points are connected to the fixed network. In a manner similar to repeaters, certain access points operate in this mode while others are connected to the fixed network.

These types of networks are increasingly being used at the city level, using access points both to provide wireless service and to operate in a similar fashion to fixed network routers.

6.6.6 Wireless LAN standards

In the early days of development of wireless networks, there were no well-defined and internationally recognised standards, posing serious problems for the interoperability of the products available on the market.

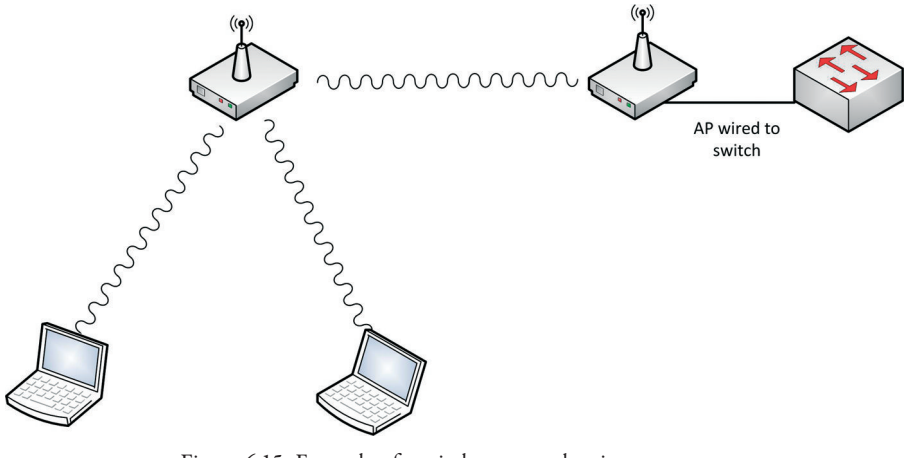


Figure 6.15 Example of a wireless network using a repeater.

For this reason, the industrial world turned to IEEE to develop a set of standards to guarantee the interoperability of the standard products. The IEEE developed a series of standards, most of them under the name of the 802.11 (x) family, which have been amply discussed in Chapter 1 and are not repeated here for the sake of brevity.

6.6.7 Personal area networks

Personal area networks (PANs) are networks that act in a relatively limited area and are used for the connection of personal devices, hence their name. They have already been explained. They are generally standardised by the IEEE, always to ensure the interoperability of the different products.

6.6.7.1 Bluetooth

Bluetooth has already been discussed, in general, above. This was born and conceived as a standard developed by different manufacturers that was subsequently standardised by the IEEE as 802.15.1.

Bluetooth Special Interest Group (SIG), created in 1998, is in charge of controlling the specifications of this standard and is composed of the principal industries of sector communications and of many other stakeholders in this standard.

The security architecture of Bluetooth, as specified by SIG, includes the functionality of authentication and encryption. All security functions are carried out at data-link level. There are four elements that are used to secure transmission:

1. a unique device address composed of 48 bits;
2. a pseudo-random 128-bit private key used for authentication;
3. an 8 to 128-bit private key used for encryption;
4. a pseudo-random 128-bit number generated by the device.

Bluetooth specifications provide three modes of operational security of the protocol:

1. Mode 1 – not secure: in this way, security is not ensured by the protocol.
2. Mode 2 – reinforced security at the level of service: in this mode, security is guaranteed after the establishment of the channel.
3. Mode 3 – reinforced security at the level of data link: in this mode, security is guaranteed before the establishment of the channel.

Bluetooth can operate in only one security mode at a time.

In addition to the three levels of security, it can operate with two levels of trust:

1. trusted;
2. not trusted.

Trusted devices are those that have a constant ratio and have full access to all the services, while not trusted devices are those that do not have a constant ratio or that otherwise are catalogued as not trusted and therefore have limited access to services.

The security architecture of Bluetooth is shown in Figure 6.16.

There are vulnerabilities in the security architecture of Bluetooth that can affect confidentiality, authentication, availability and privacy.

Confidentiality is a problem in all communications that occur by means of electromagnetic waves in space. In particular, Bluetooth does not require the encryption of all transmissions, and in many cases, this task is left to the application layer.

With regard to authentication, this can be a problem because with Bluetooth, it is the device that is authenticated, not the user. A stolen device could be used in a malignant manner if the user has not correctly configured the security of the device itself by using, for example, an adequately complex PIN.

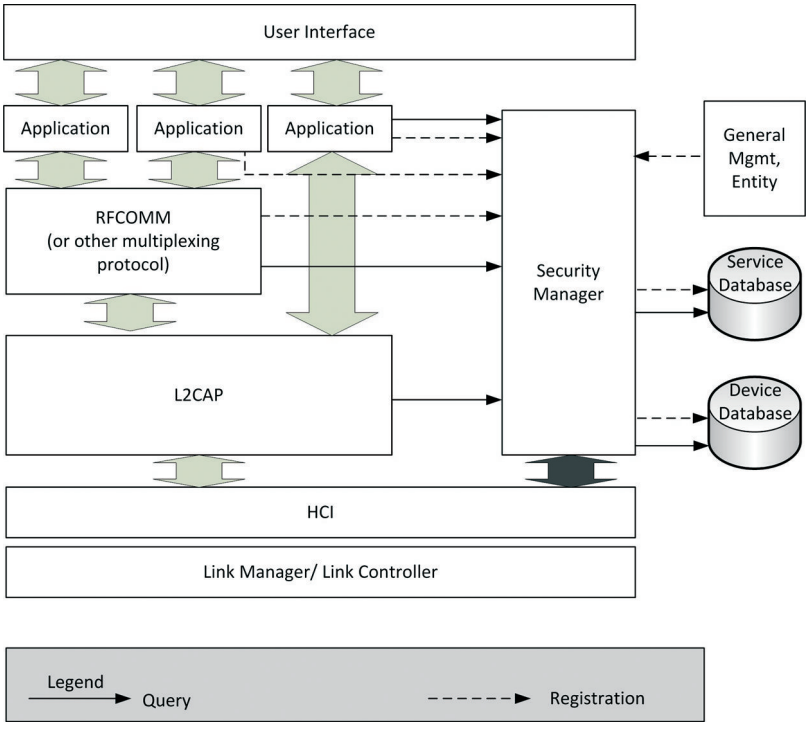


Figure 6.16 Security architecture of Bluetooth.

This situation becomes particularly problematic if the device is used for making payments. In this case, if the payment system is not equipped with recognition systems, such as a video camera connected to a video recording system, the rightful owner can never demonstrate not having made the payment. In such a situation, it is therefore very important to employ user verification systems.

With regard to availability, the very fact that Bluetooth operates by means of electromagnetic waves in space can be a problem because of the possible interference that may be experienced from other devices that operate on the same ISM band such as those based on 802.11.

In terms of privacy, this can represent a problem for Bluetooth users. In fact, it is said that a Bluetooth device is characterised by a unique identifier that is read by other devices that are in its range of action. If a device moves from one coverage area to another, its movements can be tracked, representing a problem for the privacy of the device owner. This problem is similar to the owners of cell phones with the difference that the displacement data of the latter must be stored by managers in a manner that varies from nation to nation, always ensuring maximum confidentiality of the movements of users.

The Bluetooth architecture, as has been said, does not correspond exactly with that of the International Organization for Standardization/Open Systems Interconnection (ISO/OSI) model. For an at-a-glance understanding of the differences, the two stacks are shown in Figure 6.17.

Below are the correspondences between the two models:

1. Physical layer that is responsible for the means of communication. This function is executed, in Bluetooth, through radio protocols and baseband.
2. Data link level that provides for transmission, framing and error checking. This functionality is performed in Bluetooth by the link controller.
3. Network level that provides for the transfer of data through the network in a manner, which is independent of the means and network topology. This functionality is performed in Bluetooth by

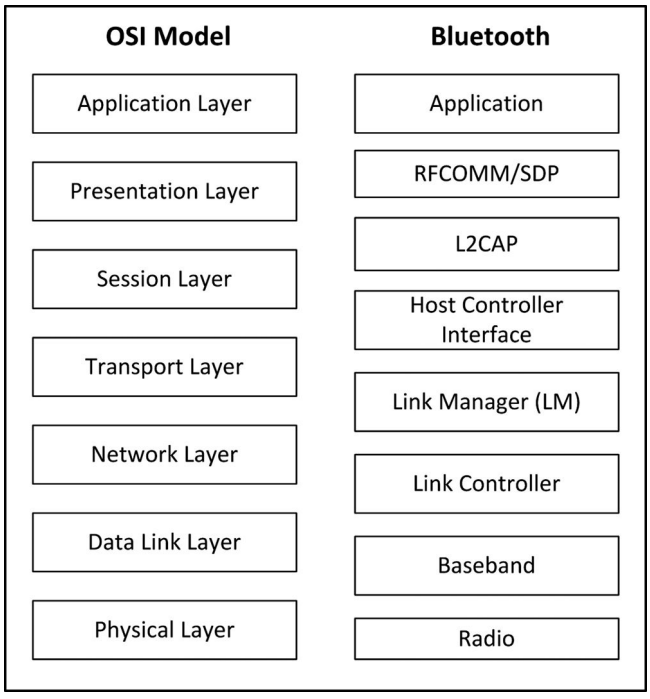


Figure 6.17 ISO/OSI model and Bluetooth architecture.

the upper part of the *link controller* and by a part of the *link manager* (LM) that manages the multiple connections.

4. Transport level that controls the division of data transferred through the network to the level provided by the application. This functionality is performed in Bluetooth with the upper part of the LM and a part of host controller interface (HCI), which provides the transport mechanisms;
5. Session layer that provides the services of management and control of data flows. This functionality is performed in Bluetooth by the *Logical Link Control and Adaptation Protocol* (L2CAP) and by the lower part of Radio Frequency COMMunication/Service Discovery Protocol (RFCOMM/SDP).
6. Level of presentation that provides a common representation for the level of application. This functionality is performed in Bluetooth mainly by RFCOMM/SDP.
7. Application level that is responsible for the management of communication between two remote applications. Such functionality coincides with those of Bluetooth.

A first level of security, in Bluetooth, is provided at the *baseband* level, which controls the connection in radio frequency between different units, also including, in control, modulation, demodulation, synchronisation and transmission. Modulation is used to maximise the number of bits conveyed by a cycle of use of the available bandwidth. *Baseband* also manages the physical channels, frequency hopping (FH), creation of packets, coding and correction of errors, encryption/decryption, consumption of energy and the search for other devices. Of all the objectives of this level, there are two, FH and encryption, that directly affect security. Bluetooth uses the technique of FH to reduce interference, reduce energy consumption and increase security. It has already been stated that the FH technique was first utilised in the military field and was then transferred to the civil context. This technique is not immune from DoS attacks that can occur imbuing the ISM band with noise signals distributed throughout the entire band.

In addition to the above-mentioned purposes, in order to also avoid interference with the devices that operate in the same band and in the same area, Bluetooth changes the frequency of transmission in

an almost random manner 1,600 times per second or, equivalently, once every 0.625 ms. Because it is assumed that interference occurs only in a limited portion of bandwidth, it is extremely unlikely that two consecutive packets are disturbed as they are transmitted on different frequencies. Each packet that is disturbed is transmitted back to another frequency of the available bandwidth and the number of retransmissions is, ultimately, negligible compared with the entire data flow.

The master unit of Bluetooth within a piconet uses its internal clock to determine the hopping pattern for use within piconets. Within each Bluetooth device, there is a 28-bit clock that is always running and operating at a frequency of 3.2 KHz, which is double the frequency of hop that has an accuracy of 20 ppm (parts per million). This clock determines when a device may or may not transmit and when a device may or may not receive. When a piconet is formed for the first time, the slave devices determine the master clock. The difference between the master clock and their clock determines a value called offset, which is used by a Bluetooth algorithm called frequency selection module (FSM) to calculate the common hopping sequence that must always be in sync with the master unit. The hopping sequence depends on the least significant 28 bits of the address of the master device, on the clock information and the mode of the country in which the network operates (because not all countries can use exactly the same frequencies). To increase the security of the piconet, every device that transmits at a given instant on a given frequency sends before the packet, an access code to the channel that is generated by the address of the master device, and other devices only accept those transmissions that have been adequately identified and that belong to the piconet to which they belong.

In order to increase security, given that the transmission of electromagnetic signals can be intercepted by anyone, cryptography is also included if properly equipped, which is further discussed hereinafter.

Particular attention from the security aspect was also given to the so-called Service Discovery Protocol (SDP) that represents the mechanism through which the devices seek the services available in their area of coverage and the relevant access modes. SDP uses a model based on request/response where every transaction is performed by a request packet data unit (PDU) and a response PDU, not ensuring that a series of requests will be fulfilled with the same emission sequence.

Piconets represent an example of publicly available ad hoc networks, which are spontaneously created when Bluetooth devices activate a process of search and reciprocally authenticate each other. As with all forms of broadcasts that use electromagnetic waves, the signals used by Bluetooth can be intercepted, and in any case, communications can undergo a process of *spoofing*. It has been observed that the specifications of Bluetooth have two modes in which security measures are provided at the data link layer to help prevent intruders; however, these methods must be used in a flexible manner to ensure a certain level of consistent security with the provision of basic services.

One of the forms of security provided is FH within the ISM band. The master device, as we have seen, determines the pseudo-random hop schema that must be used for the duration of the piconet, allowing all the slave devices to vary the frequency at the same time.

Bluetooth devices that intend to enter the network are authenticated through a series of challenge communications that are used by each device to identify other devices. This mode prevents the occurrence of spoofing attempts and discourages access that is not permitted to data or to available features. In addition, the small size of the radius of action of Bluetooth devices provides a certain level of protection from attempts to intercept as a potential eavesdropper could be visually identified.

Once all the devices are identified through mutual authentication, the unit master of the piconet may require the use of encrypted communication. If this occurs, the master unit creates and distributes a temporary key that is used by all devices that are part of the piconet.

When two Bluetooth devices come into reciprocal range and intend to communicate, the LM requests establishment of a link-level connection. For devices that operate in modes 1 and 3, an L2CAP connection is created without further requests, immediately establishing the channel. For devices that operate in mode 2, a series of other operations must first be performed. In this case, the security archive

is interrogated to check if the device is authorised to access. If the device is not authorised, the same is rejected and the process ends. If the device is authorised, the same is granted access, and subsequently it is determined if the same should be authenticated or not. If authentication is not requested, an L2CAP connection is established. Alternatively, the device is authenticated, and then it is decided if encryption can be activated or not. The L2CAP connection is responsible for performing these last steps.

The security policies of Bluetooth are managed through an exchange of requests with the *security manager* that performs the following operations:

1. stores the security information relating to the service;
2. stores the security information for the device;
3. responds to requests for access for the implementations of the protocol or the application;
4. reinforces authentication and/or encryption prior to connection to the application;
5. begins or processes inputs from external security control units or external security control entities (ESCE), as devices or applications, to create trusted connections at the device level;
6. starts alignment (pairing) and asks for the user's PIN.
7. responds to requests for access of the protocol layers.

Bluetooth allows the adjustment of the levels of security both at the device and service level. The *security manager* manages the archives that contain information about the devices and services. Security is activated when a protocol or access service requests it. The *security manager* interrogates the service or the protocol to understand if authentication is required or not. In the event of a positive response, it performs these checks. Then, it checks if encryption is required; in the event of a positive result, it provides for management of the connection key and starts encryption. At each point of the process, if the necessary requirements are not met, the *security manager* denies access.

The classification of devices is performed on the basis of three categories and two levels of trust:

1. Trusted devices: those devices that have been previously authenticated and that are marked in the archive as trusted. They have a fixed relation and full access to services for which the relation has been set.
2. Devices that are not trusted: are those devices that have been previously authenticated but that are not marked as trusted in the archive. They therefore have limited access to services. These devices, in general, do not have a fixed relation with the device in question.
3. Unknown and not trusted devices: those devices that do not have security information in the archives.

In a similar manner, the archive of services contains information about authentication, authorisation and encryption requested for access. Services are divided into three groups:

1. Authentication and authorisation requested: in this case, the services only provide automatic access to those devices that have been previously authenticated and with which a secret key stored in the archive has already been exchanged. Manual authorisation via a PIN is however possible.
2. Requested authentication: in this case, any device that can be authenticated can access the services.
3. Open services: in this case, the services require neither authentication nor authorisation.

If they are not established differently, authentication and authorisation for incoming connections and authentication for outbound connections is provided by default.

With regard to authentication, it has already been seen that all the devices in a piconet must be adequately identified by the other subjects in the network before it can be operated. Slaves, as well as masters, can perform authentication. Four parameters are used in the process of authentication:

1. the device address (BD_ADDR, 48 bits long);
2. a private authentication key (128 bits long);

3. a private encryption key (8 to 128 bits long);
4. a random number (RAND, 128 bits long).

The LM exchanges messages in the form of PDU; it represents, as already stated, the network layer in communications between devices. PDU communication has precedence over the communication of users' data, but it can be delayed by interference and by the associated retransmissions. In the event of delayed transmissions, each device that has failed in communicating with others is ousted from the network within 30 s.

Authentication begins when a verification unit sends a PDU that contains a RAND to the caller. The caller returns a response that contains an encrypted version of the RAND of its Bluetooth address and a secret key. If the received response is as expected, the caller is considered authenticated. Optionally, the roles can also be reversed in order to perform mutual authentication.

When an authentication process fails, it is necessary to wait a certain period of time before a new attempt can be made. This operating mode ensures that a possible intruder cannot attempt a brute-force attack by trying all possible keys in order to be able to enter the network. In addition, for every failed attempt, the time required before the next attempt can be made increases exponentially. In every case, after a timeout period, the time interval returns to its original length. The length of these time intervals depends on the type of implementation.

The possible reasons for which authentication can fail include:

1. termination of the existence of the connection;
2. authentication not being supported in the contacted device;
3. unavailability of the key on the device in question;
4. disabling of authentication.

When no connection is established between two devices, a process of alignment (pairing) begins where an initialisation key is generated that is used for authentication. The key is generated by entering a simple PIN in each device. This method generates a temporary key that allows the authentication process to proceed as if a key link were present. Some devices that do not have a user interface may already contain a PIN to allow performance of this type of process.

Bluetooth uses a variant of the cipher called SAFER+ to authenticate each device. This cipher was originally developed by the Swiss Federal Institute of Technology and Cylink Corporation as a candidate of Advanced Encryption Standard (AES) but was not accepted as a world standard. In any event, it is publicly available. It was selected as a security algorithm for Bluetooth in 2000. It uses 128-bit keys starting from a plaintext of 128 bits even if the keys can be shorter by using an appropriate seed number derived from the PIN.

The encryption keys of the link used by Bluetooth devices are divided into three groups:

1. link keys;
2. sub-keys;
3. encryption keys.

All the above keys are generated from one of the five algorithms contained in SAFER+.

Devices that use encryption must be previously mutually authenticated using trusted responses. Using a link key generated through the authentication process, an encrypted mode request can be made in two ways:

1. point-to-point encryption between two devices;
2. encryption of all the broadcast packets for the piconet.

The maximum key length is 128 bits, but depending on the specific legislation on cryptography in different countries, this length can be reduced. Once the request has been accepted by all participant

devices, key negotiation begins. When this process ends positively, encryption begins via the transmission, in broadcast, of an encrypted signal.

The process continues with the master device prepared to receive the encrypted data, slaves prepared to send and receive encrypted data and finally, the master prepared to send encrypted data. All the encryption of transmissions is symmetrical.

There are three possible modes of different encryption if the master key is used:

1. no encryption;
2. non-encrypted broadcast traffic, but individually routed traffic that is encrypted;
3. encryption both of broadcast traffic and point-to-point traffic.

If the unit key to the unit or a combination key is used, broadcast traffic is not encrypted, while individually routed traffic can be encrypted. The master and slave must agree on the mode of use. The master sends a request on the mode to the slave that can be accepted or rejected. If the initial mode is rejected, the master may try again, proposing a different mode.

Before two devices begin to exchange encrypted traffic, the same must negotiate an appropriate key length. The key length varies from 8 to 128 bits. This variability depends on:

1. specific legislation on the cryptography of the country in which the devices operate;
2. possibility of future expansion without having to redesign the security.

In addition, the user cannot adjust the length of the key within a specific unit; it must be set at the time of manufacture to prevent the end user from manipulating it.

Key negotiation is very similar, in substance, to the process of negotiation of the encryption mode. Initially, the master sends a suggested key length to the slave. This initial length is the greatest permitted length. If the slave accepts it, the encryption process begins. If, on the contrary, the slave cannot support the length of key suggested by the master, it sends a counter-proposal. If the counter-proposal is accepted by the master, the process of encrypted communication begins. It is seldom the case that the master and slave are unable to find a shared key length. In the latter case, the negotiation ends and the encryption process cannot be commenced.

Bluetooth uses a 128-bit symmetric flow cipher named E_0 that is used to guarantee the confidentiality of data during transmission between two or more devices. This cipher is composed of three parts:

1. the key generator that uses four parameters: the encryption key, the Bluetooth address of the device, the clock and a RAND;
2. the key flow generator that uses a series of linear feedback shift registers (LFSR);
3. an XOR function that is used to blend together the plaintext with a key to obtain the ciphertext, or to extract the plaintext from the ciphertext using the key.

Each data packet is separately encrypted, while the access codes and the packet headers are rarely encrypted. The cipher used is subject to attacks of the divide-and-conquer type using a vulnerability that is manifested by the operation of re-synchronisation of the cipher after each packet is transmitted or received.

This cipher is, furthermore, subject to a series of other attacks that do not make it particularly sturdy and secure.

Bluetooth is, unfortunately, subject to intentional attacks by radio frequency noise (jamming), which is an illegal activity in most of the countries in the world. To do this, signals must be emitted that do not predominate over the Bluetooth signal. Given the limited nature of the radius of action of the same, the jammer should be in the immediate vicinity of the piconet or emit, however, with sufficiently high powers that must increase with the increase in distance between the jammer and the piconet itself.

There are currently four areas of vulnerability of Bluetooth:

1. attacks on the address of devices;
2. key management;
3. attacks on the PIN code;
4. absence of support for authentication.

Applications that run on devices may decide to make the device itself connectable or exposable. This represents a vulnerability because in this way it is possible to trace the device through its specific address.

The exchange of keys is also a risk in that until a secure connection is established between the devices, all exchanges, including those of the variables used to calculate the keys, are performed in plaintext. The only defence that can be in place during the calculation of the keys is the repeated application of the PIN code, even if this is a weak security measure. This problem could be solved using the public key encryption, but this would add complexity to a system that is already complex.

Another vulnerability of Bluetooth is that the device is authenticated and not the user, as has already been stated. In this sense, a device that is stolen or lost could be used for fraudulent purposes. To avoid this, security at the application level could be implemented using more robust or public-key system symmetrical ciphers.

6.6.7.2 Infrareds

Infrareds are not properly considered a PAN technology and are not standardised in the 802.15 family. It is only their brief coverage distance, however, with line of sight that allows them to fall within the scope of PAN. They were initially standardised in the 802.11 family. They are very useful for transferring files between devices within optical range and up to distances of a few meters. Because of their high directivity and limited scope, the risk of interception by attackers is reduced to a minimum, ensuring a high level of security.

6.6.7.3 Ultra-wide band

The technology for an ultra-wide band (UWB) allows transmission at a short distance through a relatively extended frequency band, as the name itself suggests.

Because this system uses a relatively extensive band, it is more likely that in such a band, there are devices that emit radio frequency. In this sense, UWB is able to take account of such interference and ensure maximum functionality. It has already been said that the military world is very interested in UWB technology because of the low power used and its high resistance to voluntary interferences (jamming).

The UWB standard is currently being researched by the IEEE.

6.6.7.4 Zigbee

Zigbee was created by the 802.15.4 standard. This standard was created for solutions with a low speed of communication that can guarantee long duration of batteries. Another strength is the ability to operate with reduced complexity.

The 802.15.4 standard, like most of the IEEE standards, only operates at the physical and MAC levels. The Zigbee Alliance entered the formulation of the standard and extended it from the network level up to the application level. In this sense, the management of security and topology, routing, management of the MAC, the research protocols and Application Program Interfaces (APIs) available to allow programmers to create libraries of functions and specific applications is provided for.

The Zigbee standard can operate on three different frequencies:

1. a valid frequency throughout the world;
2. a valid frequency for Europe alone;
3. a valid frequency for the United States alone.

The first frequency is at 2.4 GHz. It is located on the ISM band where Bluetooth, 802.11, 802.11b and 802.11g also operate. This means that this frequency would prove to be very crowded and would experience interference from all other devices that use microwaves at this band. The advantage of the use of this frequency is the interoperability at a global level, by the high *data rate* that is reachable and by the high availability of channels. The Zigbee standard, which operates at a frequency of 2.4 GHz on the ISM band, is able to reach 250 Kbps with 16 channels available.

The second frequency, only valid for Europe, is 868 MHz. At this frequency, the *data rate* is much lower than the frequency of 2.4 GHz, reaching 20 Kbps with only one static channel available.

The last frequency, valid only for the United States, is 915 MHz on the ISM band, with a *data rate* of 40 Kbps and 10 channels available.

6.7 Wireless WAN technology

WAN-type wireless network technology operates using radio frequencies and microwaves on relatively large areas and hence is addressed in this chapter. The fundamental systems have already been explained in Chapter 1 and will be partially discussed below.

6.7.1 Cellular phone technology

It has already been seen that cellular technology has evolved over three generations since its debut in 1980. Changes in technology have seen an increase in the levels of security of the same.

The first generation (1G) was characterised by the use of analogue voice that was transmitted in plaintext with significant problems from a security perspective.

The second generation (2G) used digital technology, higher quality and a primary form of encryption.

The third generation (3G) also uses mobile phones as data terminals, being able to provide many services even if with a relatively limited data exchange rate.

Work is currently taking place on fourth-generation (4G) cellular phones able to ensure high-speed data connections and the use of multiple wireless technologies in such a manner that the mobile terminal can choose the best wireless network to use from all those available in the area of coverage.

6.7.1.1 The first generation of cellular phones

As already stated, the first generation of cellular phones was based on terminals that transmitted voice in an analogous manner and without any form of encryption, while exposing communications to considerable problems from the point of view of security of the same in that communications could be heard without difficulty by using a normal device called scanner. Plaintext transmission also generated another significant problem called cloning. In fact, at the time of authentication of the mobile phone with respect to the network, the phone's data were transmitted in plaintext and then exposed to the possibility of interception by cloners. Cloners, once they had acquired such data, entered it in another cell phone (clone) that could thus easily enter the network, placing the costs of fraudulent communications on the bill of the user as rightful owner of the cell phone.

6.7.1.2 Second generation of cellular phones

Towards the end of the 1980s, the introduction of the second generation of cellular phones began that passed from analogue technology to digital technology, allowing an increase in the capacity, security and quality of communications.

In 1989, the Cellular Telecommunication Industry Association chose time domain multiple access (TDMA) technology over the frequency domain multiple access (FDMA) technology proposed by Motorola, making the former a reference of the second-generation cellular telephony. These techniques have already been described above.

6.7.1.3 Code division multiple access

Code division multiple access (CDMA) is a technology that was developed for the US army in the early 1960s. CDMA was developed to provide a technology that was attacker-resistant and that was characterised by a certain level of security, making it immune to interception. In the early 1990s, the Telecommunication Industry Association adopted the CDMA technology as digital technology for access to cellular telephony, using for the first time a technology that was only for military purposes.

We have seen that this technology works by assigning a digital code for each bit of the vocal signal that is sent, in a coded manner, in space using electromagnetic waves. All calls that are carried out under coverage of a single radio base station use the same frequency. This means that at the same time, numerous calls overlap by generating a single signal. The code used is known only to the cellular telephone and to the base radio station. This code is used to collect from the signal only general information about the call of interest, allowing the mobile phone and the radio base station to separate the communication of interest from any other call. This code is to be very robust against attacks, while guaranteeing a high level of security and a high robustness against eavesdropping.

6.7.1.4 The GSM standard

The Conference of European Posts and Telecommunications (CEPT), in 1982, created the Groupe Special Mobile (GSM) group with the aim of developing a standard cellular network that was valid for the whole of Europe. In 1989, the European Telecommunications Standards Institute (ETSI) took control of the GSM group and ended the first standard in 1990. The following year saw the start of the development of GSM networks throughout Europe. GSM was the first second-generation digital network of the time. GSM is currently, the most widespread cellular technology throughout the world.

GSM uses both an FDMA and TDMA system of access. It takes the available bandwidth and divides it into 124 sub-carrier frequencies using FDMA. In total, 125 sub-carriers are used but the first is used to protect the GSM from interference. The remaining 124 sub-carriers are divided into 8 time slots using TDMA, allowing 4 devices to use each channel. This is due to the fact that each device uses two channels, one for receiving and one for transmitting. The system gives the impression of receiving and transmitting at the same time even if not using the two channels available at the same time.

The GSM standard is very complex and is described in a volume of approximately 8,000 pages.

With regard to the GSM phone, it is also called mobile station (MS) and is equipped with a subscriber identity module (SIM) card that contains all the information necessary for communication with a specific mobile phone manager. The phone also contains the so-called Mobile Station International Subscriber Directory Number (MSISDN), which is a unique identification code that informs the network about the operator of the service. This code is used as authentication code.

With regard to the radio base station or base station (BS), it is used by the phone as a reference for performing its own communications services. The BS operates as a true wireless access point and is equipped with an antenna, for receiving and transmitting signals, and a control module, which manages all of the services. Before you can make a call, the cell phone and the radio base station

exchange a series of information. When a phone is accessed, it attempts to find its location information. This information may be requested by phone or requested of the cellular network. This activity is called mobile station roaming number (MSRN). This number is based on the location. To obtain the MSRN, the local gateway mobile switching centre (GMSC) is contacted, which is a device that processes all calls. It performs different functions from those performed by the controller of the radio base station that manages only the connections between the mobile terminal and the radio base station and vice versa. Once the controller of the radio base station receives a request for MSRN from a mobile terminal, it forwards the request to the local GMSC. The GMSC uses MSIDN to find the operator gateway. Once the gateway has been found, the other gateway sends the same the so-called visitor location register (VLR) or the current location of the mobile terminal and receives the so-called home location register (HLR). At this point, both the gateways know where the device comes from and where it is located. Once this process is completed, the telephone call can take place.

The operation of making a phone call is very similar to the functions of control and location. In this case, the phone sends the number that it intends to call to the radio base station. The radio base station receives this request and sends it to its controller that in turn, sends it to the local gateway. The local gateway checks the number and searches for the gateway of numbers to call. Once the gateway has been found, an HLR and VLR request is made in such a manner that an MSRN can be found. Once the other gateway responds, it communicates the gateway address where the number requested can be found. Once this has been done, the gateway that handles calls locates the other number and sends the call.

Within the GSM, the general packet radio service (GPRS), which represents a service of high-speed data exchange, was subsequently developed. It allows packet-based communications to take place on a GSM network. It is very different from circuit switched (CS) networks that use most of the cellular networks. GPRS uses GSM channels or, more properly, its TDMA slots. This allows multiple mobile terminals to use GPRS without interfering with voice communications that occur in the same area. In this situation, voice always has priority over data traffic. This means that in areas with high voice traffic, the speed of GPRS data traffic undergoes an inevitable reduction. The *data rate* of GPRS is 170 kbps that in practical cases reaches approximately 25 to 275 kbps. GPRS uses the concept of classes that allows it to use a number of different channels for the uploading and downloading of data, enabling optimisation of the data flow for applications that exchange the same asymmetrically.

The aspects of security built into GSM are very important. The GSM security model is based on shared information that is both on the SIM card of the phone and in the system of the phone operator. This confidential information, called Ki, is a 128-bit key, which is used to generate a 32-bit response.

When a mobile terminal enters the network, it connects to the local gateway according to its position. This gateway searches the base gateway and receives a random challenge and a signed response from its base HLR. This is done by sending two other pairs of challenges and a signed response that is used in the following. The local gateway, once the information is received, only sends the challenge to the radio base station. When the radio base station sees this challenge, it uses the shared secret to generate a signed response to be sent to the local gateway. To create such a response, it uses the cryptographic algorithm called A3 with the shared secret and the challenge. Once the local gateway receives this information, it compares it with the information received from the operator. Once this has been done, the authentication process is completed.

In addition to a process of secure authentication, a process of secure communication is also provided in the GSM. In this case, the second couple of challenges and the signed response is used. This information was created using an algorithm different from A3 and it is called A8. This algorithm is used to create a key that is used with another algorithm, called A5, to encrypt the data. It operates in a similar manner to authentication. The mobile terminal uses the challenge and the shared secret key contained within the same to generate a new key. This process uses the A8 algorithm to generate a session key named Kc, which, in turn, is used together with the number of frames to generate a single stream key for each frame.

In GSM another algorithm is used called COMP 128. It is used for both A3 and A8 on most GSM networks The COMP 128 algorithm generates both the signed response and the session key just once. The key length of COMP 128 is 54 bits instead of 64 bits, which represents the length of the key of the A5 algorithm. In this sense, 10 zeroes are added to the key that is generated from COMP 128. This means that the space of the keys used to protect the key is not 64 bits but 54 bits, decreasing the overall level of security.

With the passing of the years, various attacks have been conducted in relation to GSM. It is also important to remember that the A3, A5 and A8 algorithms were not made public against the normal procedures that suggest making public the cryptography algorithms in order to be able to test the level of security from the outside world through attacks of any kind, ensuring, essentially, that the same cryptography algorithms are truly secure. In 1998, the members of Smart Card Developer Association showed that it was possible to violate the A5 algorithm, which is used for authentication in the GSM system, in a few hours using a normal personal computer (PC) characterised by the facilities available at that time. Subsequently, it was demonstrated that it was possible to carry out an attack against the A5 algorithm in less than two minutes using a standard PC. As a result of this, it was realised that the GSM system is characterised by a relatively low level of security.

6.7.1.5 The SMS service

The term "SMS" (acronym for short message service) is commonly used to indicate a short text message sent from one phone to another. The correct term would be "SM" (short message), but the custom of indicating a single message with the name of service is now widespread.

The service was originally developed on the GSM network; however, it is now also available on other networks such as universal mobile telecommunications system (UMTS) and certain fixed networks. It is possible to send an SMS to a mobile phone even from a computer, via the Internet, and from a fixed telephone line.

Among the main advantages of the SMS, underlying the extraordinary diffusion of this service as a system of communication, is the low cost compared to a long phone call (the advantage in reality is often non-existent because a conversation via SMS consists of several SMS over a period of time, one in response to the other) and the possibility of making communication asynchronous, that is reading the message at any time subsequent to the receipt.

The message has dimensions that vary from system to system. Table 6.1 shows a summary of the size of the messages according to the cellular telephone system.

Table 6.1 Size of the messages according to the cellular telephone system.

Cellular system	Kind	Length of messages (in characters)	Area of use
GSM 900	Digital	160	Everywhere
GSM 1800	Digital	160	Everywhere
GSM 1900	Digital	160	North America
NAMPS	Analogic	14 alphanumeric, 7 to 32 numeric	North America
CDMA	Digital	256	North America
IDEN/NEXTEL	Digital	140	North and South America
TETRA	Digital	256	Europe
Iridium	Satellite	200/20	Whole world
Globalstar	Satellite	160	Whole world

From the transmission point of view, the data units of the SMS message (six different PDUs are used) are included in the GSM control channels so that it is possible to receive or send a message even during a conversation. This also implies that the operator has a minimal cost for sending SMS messages.

The standard provides two different types of messages: point-to-point (SMS/PP) type, used in communication from one terminal to another, and the cell broadcast (SMS/CB) type, originating from a cell and distributed to all terminals under its coverage.

It has already been seen that a cellular network operates on the basis of a division into cells of the territory and is designed to provide the service of communication to a large number of customers. This division is made necessary because the user of a GSM network is mobile and as such requires mobile points of access.

Returning to SMS messages, there will be, in the case of *cell broadcast,* a message that is sent to all the MS present in a cell. All users that have subscribed to the service and that will be present in the cell will be able to receive the *cell broadcast* message. This service is one-way, non-wait, that is, a confirmation of receipt by recipients. It is used to transmit information on traffic conditions, weather forecasts, marketing information, etc.

In *point-to-point* service, a message can be sent from one mobile device to another or from a PC to a mobile device and vice versa.

The messages are stored and transmitted by a message centre called *SMS centre* (SMSC). The message centre is the electronic equivalent of the regular postal service because it stores and forwards messages as soon as these can be routed. Every GSM network can support one or more message centres to sort and route messages. Each SMSC controls, organises and sends the message to the operator; it also receives and passes every confirmation message to any GSM device of any network. In practice, an SMS can pass through different networks without any obstacle.

There are several ways in which an SMS can be sent, depending on the interface supported by the message centre of the GSM network. A user may need to call an operator to send the message rather than directly create it through the keyboard on a mobile device.

Some message centres together with certain companies have developed their own protocols for sending SMS. As a result, most GSM networks now offer access to their own message centres using these protocols on different hardware interfaces: modem dial up, X[25] and again the Internet.

With regard to the basic architecture of the SMS service, the main components of the structure of an SMS network are shown in Figure 6.18.

When an SMS message generated by a mobile device has to be routed, the SMSC forwards this message to the SMS-*GatewayMSC* (SMS-GMSC), which interrogates the HLR database for information on routing and sends the SMS message to an appropriate MSC. The MSC component that receives the message delivers it to the MS.

Considering from the other direction, in the case where a message to an MS must be delivered, the addresses of the MS of destination are derived from global information from the SMSC. If routing takes place outside the public network, for example on the Internet, the message will be routed through an appropriate *short message service-Internet working MSC* (SMS-IWMSC).

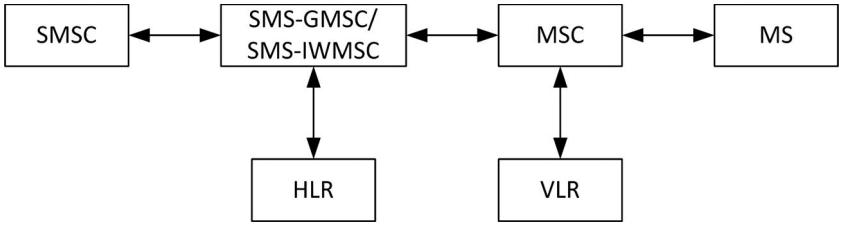


Figure 6.18 SMS network architecture.

Figure 6.19 shows more clearly and in greater detail the network architecture, where the following elements are present:

1. *Short Messaging Entity (SME)*: it is the entity through which you can send and receive an SMS message (for example, a mobile phone or an SMS Gateway). The (SME) can be present within a fixed network.
2. *SMSC*: it is responsible for the transmission, storage and delivery of an SMS message between an SME and another phone (or other SMSC).
3. *SGMSC*: this is an MSC capable of receiving an SMS from an SMSC to interrogate HLR for routing information and, therefore, to send the message to another MSC for receipt by the recipient mobile phone.
4. *HLR*: this is the database used for permanent storage of subscribers' data with their designated service profiles. By querying HLR, SMSC obtains the routing information necessary for sending the SMS message to a subscriber.
5. *MSC*: it has the function of switching vocal and data traffic to the GSM network.
6. *VLR*: it is a database that contains the information of users that are temporarily assigned to the operator because of roaming features.

The SMSC uniquely identifies each SMS message by adding a *time-stamp* to it (message time-stamp) in the field *SMS-DELIVER TP-Service-Centre-Time-Stamp (TP-SCTS)*. The accuracy of the time-stamp is to the second, in relation to the arrival of the message. In the event two SMSs arrive at the same time, the SMSC ensures that a different time-stamp is used for the two competing messages in order to avoid collisions.

The MS is enabled to send and receive Transport Protocol Data Unit (TPDU) messages, receiving, possibly, a confirmation message for correct receipt. In addition, the MS is responsible for notifying the network when it has space available to receive messages.

With regard to the protocol architecture of the SMS service, the layers of the protocol for SMS are shown in Figure 6.20.

As can be seen in Figure 6.20, the *short message-transfer layer* (SM-TL) serves the *short message-application layer* (SM-AL), allowing the exchange of SMS with a "*peer*" so as to be able to receive confirmation of receipt. The *SM-TL* layer exchanges TPDU with the "*peer*" entity.

The *short message-relay layer* (MS-RL) transports the TPDU through the *short message-link layer* (SM-LL).

With regard to the types of TPDU for the SMS service, there are 6 types of TPDU that are dealt with by the *SM-TL* layer which are:

1. *SMS-Deliver*: transportation of a message from an SMSC to an MS;

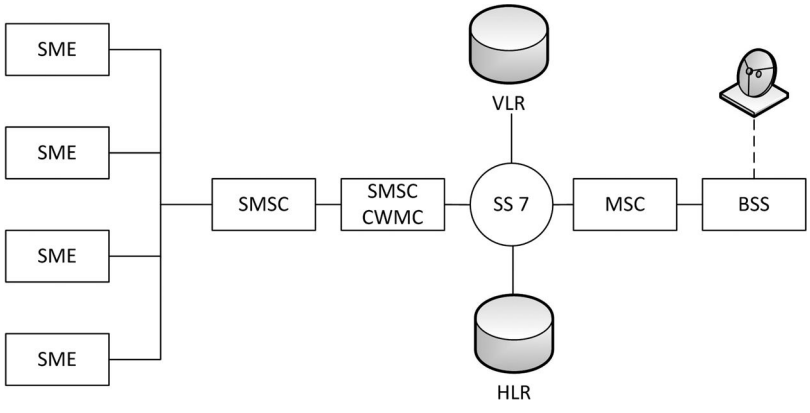


Figure 6.19 SMS network Architecture.

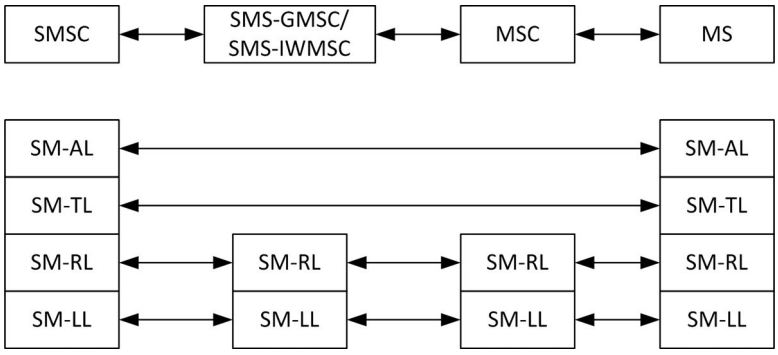


Figure 6.20 Protocol architecture of the SMS service.

2. *SMS-Deliver-Report*: transportation of information concerning the cause of an error;
3. *SMS-Submission*: transportation of a message from an MS to an SMSC;
4. *SMS-Submission-Report*: transportation of information concerning the cause of an error;
5. *Status-Report SMS*: transportation of the status report between SMSC and MS;
6. *SMS-Command*: transportation of a command between MS and SMSC.

The elements of *SMS-Deliver* and SMS-*Submission of* a TPDU are shown in Figure 6.21. The most important elements are:

1. TP-data-coding-scheme: the element of the scheme for encoding data (TP-DCS) is used to identify the encoding scheme used by the data of a user. This scheme can be 7 to 8 bits or Unicode (16 bits) as defined in GSM 03.38.

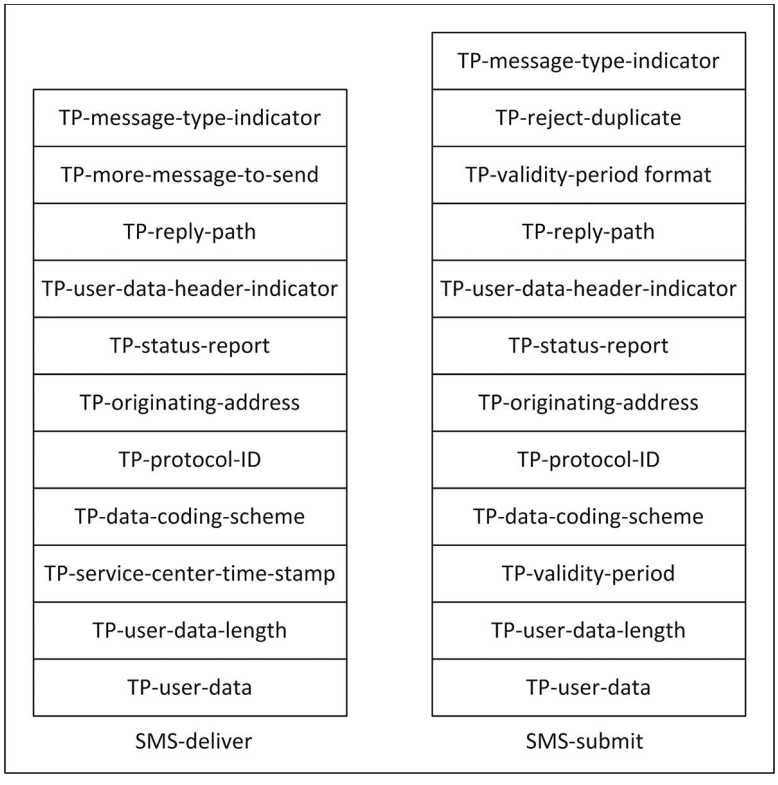


Figure 6.21 Elements SMS-Deliver and SMS-Submission of a TPDU.

2. TP-valid-period: the TP-VP element contains information, which allows an MS to specify the period of validity relating to the sending of an SMS. This value specifies how long the SMSC must keep a message "alive" before it can be deleted.
3. TP-more-message-to-send: the SMSC uses the TP-MMTS to inform the MS that one or more messages are waiting to be delivered.
4. TP-user-data-header-indicator: the first bit of the TP-UDHI indicates when the TP-UD includes an additional header such as SMS.
5. TP-protocol-ID: the TP-pid element is used by both the MS and the SMSC to identify the protocol used by the upper layer, networking with some types of telematics devices such as Telefax (group 3 or 4) and Ermes.
6. TP-user-date: this element is used to convey the SMS message. It can store up to 160 octets of data for SMS/PP messages. In addition, the same also carries a header, following the indications of the TP-UDHI field. In this case, the space taken by the insertion of a header, reduces the amount of space that the TPDU data can carry.

Figure 6.22 shows a representation of the schema of a TP-UD element with a schema data coding of 7 or 8 bits.

Figure 6.22 shows how the following components are present:

1. the *user data length* (UDL) field which provides the length of the user data field;
2. the *user data header length* (UDHL) field which provides the length of the header;
3. the *information element identifier* (IEIx) fields that provide identifiers of information elements, used to locate concatenated SMS;
4. the *information element length* (IELx) fields that provide the length of user data that follow (*IED*);
5. the *information element data* (IEDx) fields contain data.

Each field is composed of one octet. The last field, which is intended for user data, contains the actual message that may be of 7, 8 or 16 bits. In the case where the same is of 7 bits and the header does not cover all the bits until the beginning of the field containing the true message, "*padding*" is

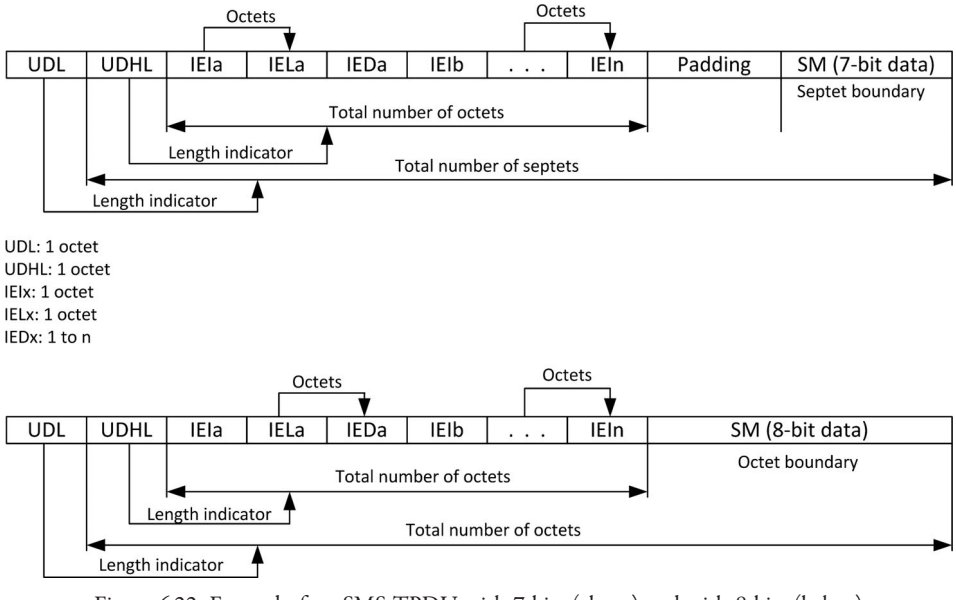


Figure 6.22 Formed of an SMS-TPDU with 7 bits (above) and with 8 bits (below)

performed to fill in the blanks. This arrangement makes it possible to have compatibility downwards; that is to say that even old cell phones that do not support TP-UD can view the message correctly.

The IEI field is the one that allows you to manage concatenated messages. This field contains all the information necessary to ensure that the receiving entity can correctly reassemble the concatenated message. It is structured in the following manner:

1. First octet: this is the message reference number that identifies the message within the same transaction.
2. Second octet: this specifies the number of messages constituting the concatenated message; this number cannot exceed 255.
3. Third octet: this identifies the sequence number in the event that a message is part of a concatenated message.

With regard to the routing of the SMS, Figure 6.23 shows an operating diagram in which a user A of a network is sending a message to user B of another network, performing international roaming. User A will use the SMSC of network 1 to send its SMS messages.

The mobile device interacts with the local network, and the SMS message is encapsulated in an SCCP packet, together with the SMSC. The SCCP packet is forwarded and exchanged until it reaches the recipient message centre (path 1).

Routing of the packet must be implemented at every point of sorting of the SCCP packets present on the path until destination. As soon as the SCCP packet delivers the message to its destination, a message of confirmation of receipt is sent using another SCCP message (path 2). To deliver the message to user B, the SMSC must access the HRL database of its local network. A localisation request is sent through the message centre via an SCCP packet, based on the mobile number of user B (path 3). As such, the SCCP international protocol routes the request packet to an appropriate HRL database (path 4). After this, the message centre sends the message to the *VisitorMSC* (VMSC) of user B on the basis of information received by HRL (path 5). Now the *VMSC* device can query the VRL database (paths 6 and 7) and then deliver the message to user B (path 8). To confirm correct receipt, the message centre sends an SCCP packet of successful delivery (path 9).

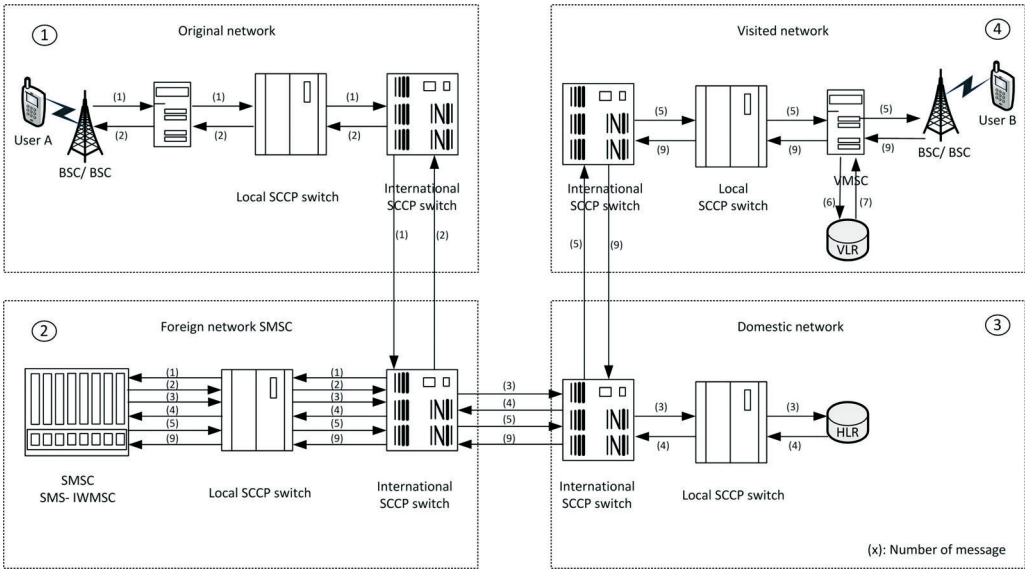


Figure 6.23 Example of international routing for SMS on a GSM network

With regard to SCCP packet loss, the SCCP routing protocol is based on a *global title* (global routing information) used to make the packets switch. Such information should also benefit the SMSCs. Routing information is placed in international switches that use SCCP, which route the international level of packets. Some international switches control only the international prefix (e.g. 39 for Italy) and forward the packet to the next "*hop*". Others also control the network prefix. If the prefix of a message is not present in the switch routing table, it is refused. Another reason for refusal of an SMS can be derived from the implementation of the service centre that is not compatible.

Packet loss can be schematically represented as follows:

1. receipt by the sender of a message of *message failed* despite the message having reached its destination;
2. receipt by the addressee of a duplicate message; this could be caused by an excessively low setting of the timeout by the message centre;
3. in the worst of cases, no message will be delivered to anyone.

With regard to the protocols for the sending of SMS, the ETSI has developed a protocol for sending SMS messages as part of the GSM standard. These specifications identify three interface protocols, *Block Mode*, *Text Mode* and *PDU Mode*, for the transfer of SMS between MS, through an asynchronous interface. Other proprietary protocols such as *Text-Based* and *TAP* will also be analysed.

With regard to *Block Mode*, this is a binary protocol that encapsulates the TPDU of an SMS, used to convey the message between MS and SMSC defined in GSM 03.40. This protocol provides for the ability to find errors and as such is very useful in applications where the connection is not reliable. *Block Mode* is well suited for remote control because the application builds a binary string consisting of a header and the TPDU relative to the message (SMS-TPDU). When an application uses this protocol, the same has access to a set of features such as:

1. sending of a message;
2. deleting of a message from a phone;
3. viewing messages in the phone;
4. transfer of messages from phone to application;
5. adjustment of the phone so the application is notified each time a message arrives.

With regard to Text Mode, this is a protocol which is based on characters, specifically based on *AT* commands modified for GSM. AT commands are alphanumeric strings; a use known to many is that oriented to modem configuration. *Text Mode* is particularly suitable for intelligent terminals or terminal emulators rather than for software applications based on a commands structure. The application passes the message in plaintext to the phone that builds the relevant TPDU. This means that this protocol offers much less functionality than the other two protocols for sending SMS. The *Text Mode* protocol only supports new message notification.

With regard to *PDU Mode*, this is very similar to *Text Mode*, with the difference that the first delegates to the application the responsibility to build the TPDU of the message. In addition to the benefits of using AT commands, this protocol allows construction of more complicated TPDU, that is to say that it can transmit binary data and not only characters.

There are, however, other protocols including Text-Based. Usually, this type of protocol is the owner, in other words, it is developed as an interface between an SMSC and a digital cellular network operator. The advantage of text-based protocols is that the user does not need a special *client* to send messages; it is possible to converse with appropriate message centres using any terminal emulation software.

As always, there are also certain disadvantages: protocols based on text offer limited support for a set of extended characters. The user is only enabled to send messages and to receive confirmation of sending. The SMSC is not enabled to notify if the message has been received correctly by the recipient.

Another protocol is Telocator Alphanumeric Protocol (TAP), which contains much more functionality than the protocols based on text, showing a greater flexibility.

In its full implementation, the protocol allows the user to perform the following operations:

1. send a message and receive confirmation of acceptance;
2. send a message and receive information on the first attempt at delivery;
3. query the current status of the message sent, according to mode 1 or 2;
4. delete the message sent with mode 1 or 2;
5. replace the message sent with mode 1 or 2, unless this has already been delivered to the mobile device;
6. update the message sent with mode 1 or 2. If the message is still in the message centre, SMSC is updated; otherwise, a new message is sent to the mobile device.

TAP is a *session-based* protocol. Each session provides a log-on, a number of transactions and a log-off.

With regard to the security aspects of the SMS service, this service is not characterised by a high level of the same, for reasons which will be illustrated below, unless use is made of dedicated programs able to use encryption.

SMS traffic is controlled, in its integrity, by means of a Cyclic Redundancy Check (CRC) to ensure that variations during transit do not occur.

The same traffic is encrypted using the SIM within the phone and by resorting to an algorithm, called IA5, which more than an encryption algorithm, is an encoding format that is very similar to ASCII. In this sense, if the SMS traffic is intercepted, it can be read without great difficulty, making this system very vulnerable from the point of view of security.

The SMS system can still be made secure by using robust encryption algorithms such as those that are used for banking applications, which use dedicated applications that employ SMS for the transmission of information.

Another big issue of SMS is the fact that the same, for security reasons, must remain registered, for a certain period of time that varies from nation to nation, in systems of mobile telephony operators, at the disposal of the security forces. In this sense, the confidentiality of information exchanged via SMS depends directly on the confidentiality assured by the information systems of mobile telephone operators, which may vary from operator to operator.

6.7.1.6 MMS service

The multimedia messaging service (MMS) is a telephone messaging service. As the name suggests, its peculiarity is the possibility of transmitting messages containing multimedia objects (images, audio, video, formatted text). The acronym "MMS" is commonly used.

The MMS standardisation was mainly performed by 3GPP, Third Generation Partnership Project 2 (3GPP2) and Open Mobile Alliance (OMA). The Third-Generation Partnership Project (3GPP) is a collaboration agreement, formalised in December 1998, between institutions involved in telecommunication system standardisation in different parts of the world. They currently form part of the 3GPP: The Association of Radio Industries and Businesses, Japan (ARIB), China Communications Standards Association (CCSA), The European Telecommunications Standards Institute (ETSI), The Alliance for Telecommunications Industry Solutions, USA (ATIS), Telecommunications Technology Association, Korea (TTA) and Telecommunication Technology Committee, Japan (TTC). The original target of 3GPP was to produce specifications for a third-generation mobile system based on GSM core network and universal terrestrial radio access. After 3GPP, there was a need to check and improve the technical specifications for GSM including more modern access technologies such as GPRS and EDGE.

MMS proposes itself as the successor of the SMS service, the latter only allows the transmission of unformatted text and is not to be confused with EMS, which is a simpler extension of SMS.

Multimedia messages consist of monomedia elements that are combined and synchronised with each other. Monomedia elements can be text, audio, voice, still images or video.

Unlike SMS messages, travelling in general on signalling channels of the GSM network, MMS messages are transmitted via a data connection. For this purpose, packet switching systems (GPRS and EDGE, for example) are often used.

These techniques do not involve the activation of a dedicated communication line (fixed) between one computer and another but allow for the simultaneous execution of multiple communications (paths) between sender and recipient because of segmentation of the information to be transmitted in information units or packets, thus maximising the efficiency of utilisation of the transmission media used.

The delivery of an MMS message can be immediate or deferred. In the first case, the telephone of the recipient retrieves the message as soon as the network informs the same of its existence; in the second case, before recovery, the permission of the user is sought.

Multimedia content created by a certain phone may not be compatible with other phones. In such cases, the standard ensures that the network operator (or the service provider) takes responsibility for the content adaptation. This is not obligatory but allows greater interoperability between different phones.

The configuration of phones will require different parameters: in order to reduce the problems that derive from this, systems have been developed for automatic configuration. For example, it is often possible to visit the website of your operator to ask for the parameters to be sent to your phone; the operator sends an SMS message containing the special parameters requested which the phone can automatically save.

Even if there is no maximum size for MMS messages, this can be set by the capacities of the terminals used and by decisions of the service providers.

The MMS environment has various characterising elements:

1. MMS relay/server: this is essential for the management, receipt and sending of messages, their control (cancellation, programming) and their transfer between different messaging systems. It is also used to convert email and fax into MMS.
2. MMS user database: here the data regarding the user profile, access to the service, configurations and messaging rules are stored.
3. MMS user agent: this is the application layer that allows users to view, compose and manage multimedia messages and manage their profile.
4. MMS Vas applications: these are the applications that offer added value to users.
5. External server: this is a server outside the MMS architecture but it is connected to the server via another messaging system such as email, SMS and EMS.

Multimedia messages can draw on different languages. For text, the most important are US-American Standard Code for Information Interchange (US-ASCII) and ISO; for voice, AMR; for audio, MPEG or MIDI for synthetic sounds; for images, JPEG, TIF and Bitmap; for video, H263; and for vector graphs, SVG-tiny.

With regard to communication, finally, this is based on a process identified by the following elements:

1. issuer or source of the message;
2. encoding or transformation of the idea;
3. message or language, set of symbols and signs;
4. decryption by the recipient of the message;

5. intended receiver or otherwise of the message;
6. feedback or response or reaction of the recipient to the message.

In the case where written communication becomes multimedia, a number of advantages can be encountered:

1. In encoding, the process is faster and more spontaneous.
2. For messages, there is a greater variety of signals to be addressed and the message is made more effective.
3. In decoding, the message is better received.
4. In feedback, with greater possibility of listening and reaction, a message becomes more engaging.

This latter element of multimedia communication is essential because if the messages are selected on the basis of moods, values and perceived needs, multimedia facilitates this selection.

6.7.1.7 The UMTS standard

3G systems are designed to provide global mobility, ensuring a series of services such as telephony, messaging and broadband data. The International Telecommunication Union (ITU) has, in its time, begun the process of setting the standards for the third generation, called International Mobile Telecommunications 2000 (IMT-2000). In Europe, ETSI is responsible for the standardisation process of UMTS. In 1998, the 3GPP was formed to continue the work of preparation of technical specifications.

UMTS represents the third-generation digital cellular radio mobile system.

Conceived as a global system comprising both land-based and satellite components, it is able to support a data transmission speed that can reach 2 Mbit/s, thus making a series of multimedia services possible such as the ability to send faxes and email, access to the Internet and to download and transmit data packets without the need for a fixed terminal, to participate in video conferencing and to use video telephony.

UMTS, unlike GSM, for the transmission of data using circuit-switched technology, integrates circuit and packet data transmission that allows diversified services to be obtained, such as virtual continuous connections to the network and alternative methods of payment (for example, payments proportional to the number of bits transferred or to the bandwidth employed).

With regard to the system architecture, the basic elements are shown in Figure 6.24.

The UMTS architecture is composed of three components: the core network (CN), the UMTS terrestrial radio access network (UTRAN) and user equipment (UE).

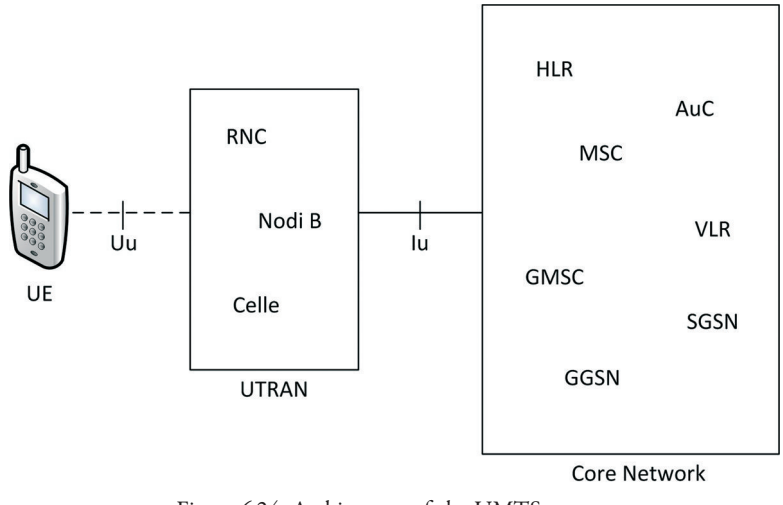


Figure 6.24 Architecture of the UMTS system.

HLR, Authentication Center (AuC), MSC, VLR, GMSC, SGSN, Gateway GPRS Support Node (GGSN) are acronyms of the main entities that constitute the CN, while radio network controller (RNC), B nodes and cells are entities that constitute the UTRAN.

UE represents the equipment of the user, in other words, the complete mobile phone card.

The UE consists of mobile equipment (ME) and one or more UMTS Subscriber Identity Module (USIM), applications that contain the functions and data needed to identify and authenticate the user, such as the international mobile subscriber identity (IMSI) that serves to uniquely identify the user, although the user may not know the value.

The CN represents the network infrastructure in which transmission occurs exclusively via cable. It is divided into the CS domain and PS domain. These two domains differ in the way they manage user traffic. The CS domain consists of the set of all the entities in CN that offer, for user traffic, a CS-type connection. A CS-type connection is a connection in which the required resources are granted when the connection is established and are released when the connection is released. The PS domain, instead, consists of a set of all the CN entities that offer for user traffic a PS-type connection. A PS-type connection carries the information of the user via bit packets.

The UTRAN represents the radio interface of UMTS. It consists of a set of radio network subsystem (RNS) connected to the CN through the Iu interface. The RNS represents the point of user access to the UMTS network in that it is responsible for the granting and release of specific radio resources. It performs two roles:

1. It establishes and manages a connection between the User Equipment (UE) and UTRAN.
2. It provides the radio resources requested during this connection.

Each RNS is controlled by an RNC, connected to a set of B nodes, that is logical nodes responsible for receiving and transmitting in one or more cells, respectively, from and towards the UE. In other words, the RNC controls the use and integrity of all the radio resources.

A B node, on the other hand, is a logical node responsible for reception and transmission in one or more cells, respectively, from and towards the UE.

With regard to the security aspects, the following will be discussed below:

1. the identity of the user;
2. the cryptographic functions;
3. the parameters for the authentication procedure generated in the AuC;
4. the authentication procedure.

With regard to the identity of the user, identification of the user means determining its IMSI. The network identifies the user either by asking the user directly for the IMSI or determining it through the temporary identifier Temporary Mobile Subscriber Identity (TMSI). For security reasons, the IMSI is requested from the user only when the latter cannot be identified by its TMSI temporary code; this is to prevent an intruder being able to intercept the IMSI along the radio path while the user is connected to the network. The TMSI code, unlike the IMSI, is protected in two ways:

1. It is encrypted via the f8 algorithm;
2. Its value is changed at least every time the user moves from one location area (LA) (or routing area (RA)) to another.

With regard to the cryptographic functions, the 3GPP considered it necessary to equip the UMTS system with certain security features, performed via the use of functions and cryptographic algorithms. The cryptographic functions used in the authentication procedure are as follows:

1. f0: RAND (pseudo) random number generation function;
2. f1: feature for network authentication; this generates a MAC or XMAC (expected MAC);
3. f1*: function for the authentication of re-synchronisation massage; it generates a message

authentication code-synchronisation (MAC-S) code or expected MAC-S (XMAC-S);
4. f2: function for user authentication; it generates a user response (RES) code or expected RES (XRES);
5. f3: function for the generation of the cipher key (CK) that is given in Input to the f8 function described below;
6. f4: function for generation of the integrity key (IK) that is given in input to the f9 function described subsequently;
7. f5: function for the generation of the anonymity key (AK) used in normal operations.

The cryptographic functions that are used for the encryption of user traffic and of certain signalling messages are:

1. f8: function for data confidentiality; this generates a keystream;
2. f9: function for data integrity; this generates a message authentication code – integrity (MAC-I).

The 3GPP has established the standardisation of only functions f8 and f9.

Function f8 is for data confidentiality, while f9 is for data integrity. They were specially designed for UMTS by the participants of 3GPP. The work for the implementation of these functions started in August 1999 and ended in November of the same year. They were published by ETSI, in their first version, on 23 December 1999. ETSI also implemented their standardisation.

With regard to the parameters for the authentication procedure generated in the AuC, the operational schema is shown in Figure 6.25.

As can be seen in Figure 6.25, the parameters that are generated in the AuC for the authentication procedure are as follows:

1. Random challenge (RAND) is a 128-bit random number generated by function f0 in the following way: f0 (internal state) = RAND.
2. Sequence number (SQN) is a 48-bit number. In each authentication vector AV, created for the authentication procedure, an SQN number is inserted. The SQN numbers are mainly used as sequential numbers with respect to which AV authentication vectors of a certain user are kept in order; they thus identify in a unique way each AV vector of the user. The AuC maintains an SQN_{HE} counter for every user; so, the SQN_{HE} is an individual counter. In the AuC, the value of this counter is used by a special procedure to generate the SQN numbers. The USIM also maintains a counter. It is the authentication SeQuence Number Mobile Station; (SQNMS) counter which contains the largest SQN number that USIM has accepted up to that moment.

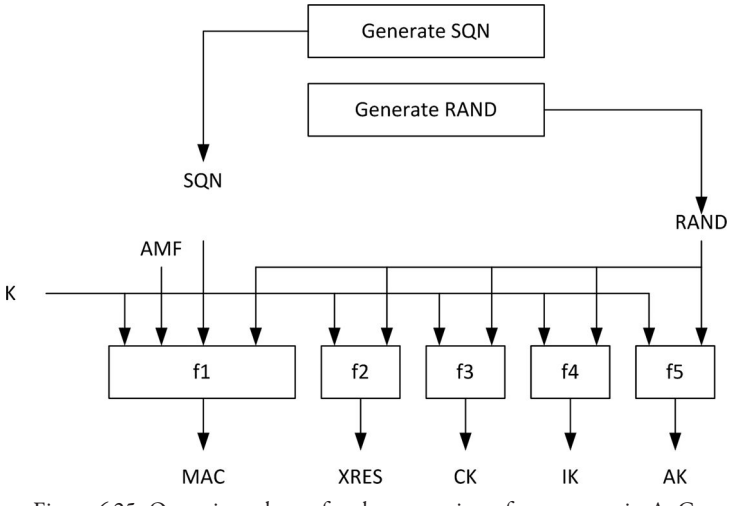


Figure 6.25 Operating schema for the generation of parameters in AuC.

3. Authentication management field (MFA) is a 16-bit parameter whose use has not been standardised.
4. MAC is a code of 64 bits generated by f1 function in the following way: $f1_K$(SQN || RAND || Authentication Management Field (AMF)) || MAC, that is f1, which has, as inputs, the key K and the parameters SQN, RAND and AMF concatenated in the manner indicated, provides MAC as output.
5. XRES is a MAC parameter that has a length ranging between 32 and 128 bits. It is generated by f2 function in the following way: $f2_K$(RAND) = XRES, that is f2 with in input K and RAND provides XRES as output.
6. CK is the 128-bit key for encrypting the data generated by f3 function in the following way: $f3_K$ (RAND) = CK, that is f3 with in input K and RAND provides CK as output. The key CK is given in input to the function for the confidentiality of data, f8.
7. IK is the 128-bit key for data integrity generated by f4 function in the following way: $f4_K$(RAND) = IK, that is, f4 with in input K and RAND provides IK as output. The key IK is given in input to the function for data integrity, f9.
8. Anonymity Key (AK) is a key of 48 bits generated by the f5 function in the following way: $f5_K$(RAND) = AK, that is, f5 with in input K and RAND provides AK as output. The key AK is used to keep the value of the SQN number hidden. In fact, at the beginning of the authentication procedure, AuC sends USIM an SQN number inserting it in a parameter called AUthentication Token for Network authentication (AUTN); however, in this parameter, for security reasons, the SQN number is not inserted directly but is the result of the XOR operation between SQN and AK. After their generation, the parameters SQN, AK, MFA, MAC are concatenated in the following manner: AUTN = SQN (+) AK || AMF || MAC, and AUTN parameter is thus obtained. Finally, the authentication vector AV is obtained from the concatenation of the parameters RAND, XRES, CK, IK and AUTN: AV = RAND || XRES || CK || IK || AUTN.

With regard to the authentication procedure, it is the manner by which the user and the network reach mutual authentication on the basis of a secret shared by AuC and USIM. The shared secret is a 128-bit long key K that is known and available only in AuC and in USIM. Every user is assigned a different key K.

Furthermore, through the process of authentication, a new key pair CK and IK is established for algorithms f8 and f9, respectively.

It is the VLR/SGSN register that starts the authentication procedure after the first registration of the user following:

1. a service request (a phone call, a connection to the Internet, etc.) by the user;
2. a request for updating the user location;
3. a request for connection to the network;
4. a request to disconnect from the network, etc.

A user that is stationed in a LA/RA controlled by a VLR/SGSN register is identified by the latter. From the user's identity, that is from its IMSI, the VLR/SGSN register finds in its database the ordered array (with respect to the SQN sequential numbers) that contains the vectors of AV authentication of that given user. From that array, the VLR/SGSN register takes the next AV vector and sends to the user (i.e. to the USIM) the RAND and AUTN parameters contained in this vector. In fact, the VLR/SGSN register maintains, for each user, an array containing the vectors of authentication AV and distinguishes such arrays on the basis of the IMSI of the user. In each array, the vectors of authentication AV are ordered with respect to the SQN numbers.

The VLR/SGSN register, as well as any other network entities, manages the vectors of AV authentication according to the first-in/first-out (FIFO) logic for which reason, of the AV vectors stored for a given user in its database, it will choose and use for the next authentication procedure the first that it has received from AuC. If the VLR/SGSN register has completed the vectors of

authentication AV of a given user, it requests n of them from the AuC, sending to this entity the IMSI code that uniquely identifies the user. The AuC, from the IMSI code traces in its database the AV vectors generated for that given user and sends the VLR/SGSN register an ordered array (with respect to the SQN sequential numbers) of n vectors AV. If in the database, there are no AV vectors for that user, the AuC generates n of them and then sends them in an ordered array (with respect to the SQN sequential numbers) to the VLR/SGSN register, which stores them and uses them for the next n authentication procedures that relate to that user.

6.7.2 GPS technology

GPS technology represents a spatial system owned by the US military that is managed by the US Air Force. It is used throughout the world for precision localisation, navigation and timing. This system uses three different components:

1. satellites;
2. earth-based control centres;
3. receivers.

Most of the applications aimed at ordinary users consist of providing information to drivers or to mariners concerning the road or the route to be followed.

Satellites are the first component of vital importance. There are 24 of them which orbit at approximately 11,000 miles above the earth. These satellites transmit a unidirectional signal, stamped temporally, over the whole of the earth's surface that is located below them at a given time. The time information is provided by four atomic clocks that are located within each satellite. The defence department has officially named these satellites NAVSTAR, which stands for Navigation Satellite Timing and Ranging. These satellites were launched in successive stages: in the first phase, in 1978, the first four satellites were launched, while in the last phase, the launch of the latest satellites was completed in 1995.

The second component is the control aspect, represented by five tracking stations throughout the world. The main control centre is located at Schriever Air Force Base in Colorado. The other four control centres do not have a defined name and are located in the Pacific Ocean (Hawaii and Kwajalein), the Indian Ocean (Diego Garcia) and the Atlantic Ocean (Ascension).

The localisation part is performed on the GPS system, while all other GPS services use other technologies to transmit the GPS co-ordinates. It should be emphasised that, in itself, the GPS system is not a tracking system. GPS receivers use the dedicated unidirectional transmission satellites to determine a position. No part of the GPS infrastructure is able to know who is using the signal, when and where. On the contrary, a GPS receiver cannot use the same system to transmit its position. In order for the GPS system to become part of a tracking system, it must be combined with a transmitting device. A cellular phone equipped with a GPS receiver can be used to trace its location and its movements. These phones create problems from the point of view of privacy and for this reason, in most cases, the same will allow the user to block the information relating to a location.

Having now explained the GPS system in general terms, the aspects of security cannot be addressed.

The GPS system was made available to the general public during the Reagan administration, for commercial uses. In the first phase, the system was programmed to send two signals:

1. the precision positioning service (PPS) or precision code (PC);
2. the standard positioning service (SPS) or coarse acquisition code (CA).

Sending of the double signal was designed to prevent enemies from identifying a perfect location. The PC code is encrypted by the military with a reserved cryptosystem using special keys and that can only be received with special receivers.

The PC code increases its accuracy by using a second carrier that allows receivers to measure the small delay due to the different distance of propagation of the signals emitted by various satellites in space.

The coarse acquisition code is available to the public. It is characterised by less accuracy and greater ease of interference. From the earth, it is easier to acquire the CA code than the PC code. In this sense, the CA code is first acquired by the military that subsequently authenticates using its cryptographic key and then use it as a reference to link into the PC code. The CA code is processed through a technique called selective availability (SA) that allows the government of the United States to change the accuracy of the GPS signal at any point of the earth and at any time. This possibility allows the United States government to interfere with an enemy navigation system, reducing accuracy in the event of conflict.

Even if the Defence Department has sought to reduce the accuracy of the GPS system used for commercial purposes, the manufacturers of GPS devices have found a legal loophole concerning the increase in accuracy. In this sense, they have developed a system called differential GPS. This technique uses the knowledge of the exact position of the BS and compares it with the location information of the BS provided by CA. Once the level of error has been determined, it can be applied to the position of the receiver, making the information provided by the latter more accurate.

As a result, many other techniques were developed to increase the accuracy of GPS. In May 2000, President Clinton decided to put an end to the restrictions of the accuracy of GPS, thereby making all commercial GPS devices much more precise. This led to the end of the development of new techniques to artificially increase sensitivity and a proliferation of new commercial products based on GPS.

The greatest vulnerability of the GPS system is the ability to interfere with the same, blinding the relative devices that use them.

In this sense, it is possible to use appropriate jammers, the magnitude of which depends directly on the area within which you want to interfere with the GPS. From a military point of view, it is clear that the larger the jammer, the greater the chance of locating and destroying it. In this sense, the real risk comes from portable jammers that, through widespread nationwide distribution and through the issuance of 500 to 1,000 W of electromagnetic power in the operating band of GPS, are able to interfere with the GPS signal in a relatively large area.

Currently, the European Union is also creating its own satellite positioning system called Galileo.

6.7.3 TETRA technology

Terrestrial trunked radio (TETRA), originally trans-European trunked radio, is a technology specifically developed at the level of the European Community by ETSI for communications of security and civil protection.

It represents a trunked radio system.

The TETRA system is composed of a series of radio base stations (or BS, such as in the GSM or UMTS cellular telephone system) connected to a central unit that manages and controls the service of user mobile terminals.

The system is, therefore, composed of a control centre, called master site (MS), and a variable number of BS located in the area of interest.

Each BS can support four radio channels per radio frequency carrier and can operate simultaneously on different carriers.

The MS can be directly connected with standard phone lines to interface with them.

Radio units are characterised by small dimensions and weight and by powers of controllable emission, always ensuring the best quality of communication between the radio units and the nearest BS.

The mobile system allows service, at the same time, to a predefined number of:

1. users;
2. user groups;

3. simultaneous calls;
4. simultaneous phone calls.

The frequencies used are indicated in Table 6.2.

In the TETRA system, frequencies are assigned dynamically, as required, allowing efficient management and dynamics of the system.

The digital technology used provides the following benefits:

1. better quality of voice communication;
2. greater speed of transmission and receipt;
3. less dependence on the level of the received signal;
4. greater security of conversations because of the cryptographic algorithms used;
5. the option of using mobile terminals not only as phones but also as data terminals to transmit and receive any kind of information;
6. localisation of mobile terminals because of the built-in GPS receiver.

Each radio connection is divided into 4 different channels that can be used individually or together depending on the transmission band required.

The TETRA system allows a multi-level authentication of users (user-mobile system; mobile system-fixed network; network-network; user-user), using cryptographic algorithms with a high degree of security. It supports a multi-traffic profile that allows voice and data services at the same time on the same terminal. Voice traffic is based on TDMA transmission technology, while the data traffic is based on packet data optimised (PDO) technology. The PDO technology also allows full compatibility with the TCP/IP protocol and related services.

The mobile system uses two types of logical channels:

1. control;
2. traffic.

The control channels carry signalling information. Radio terminals, when not engaged in a communication, continuously listen to the control channels where they receive information regarding the network (availability of neighbouring cells, available services, status of the channels, etc.) and can make their requests to initiate a particular service.

Traffic channels carry the information related to voice and data. Traffic channels (from 1 to 4 for each frequency used according to the service requested) are assigned to mobile terminals by the system and issued by the terminals or network when the service is ended (Figure 6.26).

The TETRA system offers the following voice services:

Table 6.2 Frequencies used by the TETRA system.

Number	Couple of bands of frequency (MHz)	
	Band 1	Band 2
Emergency systems		
1	380 to 383	390 to 393
2	382 to 385	393 to 395
Civil systems		
1	410 to 420	420 to 430
2	870 to 876	915 to 921
3	450 to 460	460 to 470
4	385 to 390	395 to 399.9

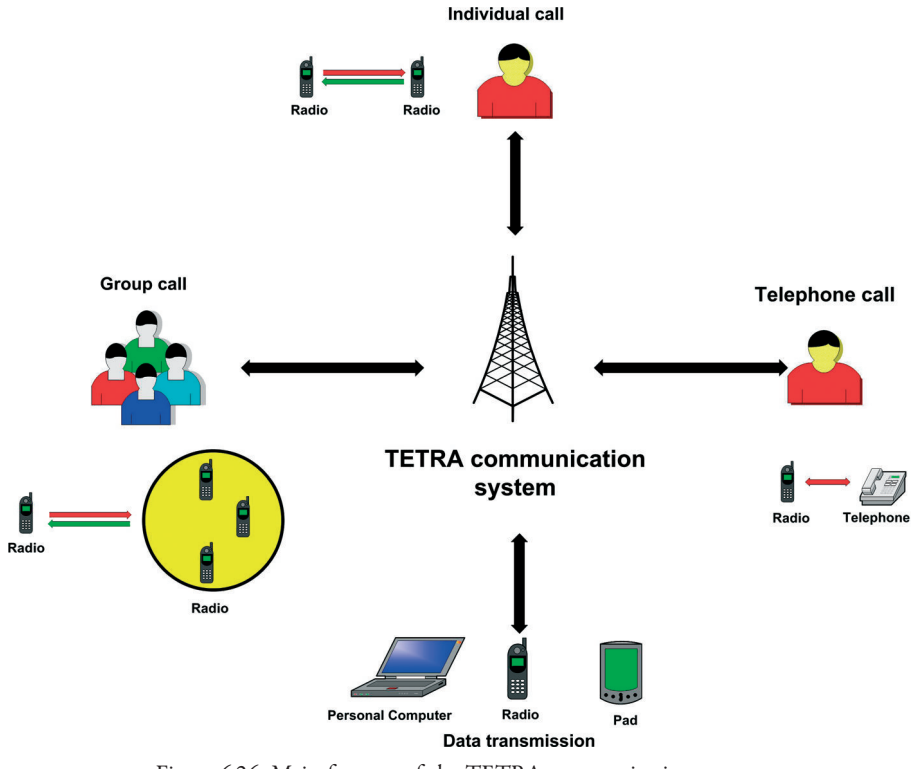


Figure 6.26 Main features of the TETRA communication system.

1. Individual calls: this service is equivalent to the communication through a mobile phone (one user calls another user).
2. Group calls: a user calls a predefined group. Each member of the group can talk and listen to the other members of the group. The group is defined in a flexible manner, that is, each user can be added or deleted from the group at any time.
3. Direct call: two or more radio terminals communicate directly without the support of the radio base station.
4. Public call: this represents a one-directional point-multipoint call in a certain area. The areas and users can be dynamically defined.
5. Emergency call: this allows a high-priority call to be made by pressing an emergency button on the radio unit.
6. Inclusive call: this allows us to call or to include within a call one or more additional users.
7. Open channel: a group of users can converse on a specific radio channel and all users can listen or talk at any time.

The TETRA system offers the following data services:

1. transmission of status: that allows the transmission of short and predefined messages from the network to the mobile units and vice versa;
2. short data service: this allows the sending of predefined messages to individual users or groups of users;
3. data transmission using a CS mode;
4. data transmission using a PS mode (X25, TCP/IP).

In addition, the mobile system is characterised by high security through:

1. using mutual authentication (radio unit-radio base station and vice versa);
2. encrypted communications using both static and dynamic keys;
3. "end-to-end" encrypted communications support;
4. disabling of lost or stolen radio units;
5. organisation of data directly through the IP network using encrypted protocols.

6.7.4 Wireless Application Protocol

The *Wireless Application Protocol* (WAP) was designed to enable easier communication between wireless devices and the Internet. WAP was developed with the creation of the WAP Forum in 1997. This forum included 350 producers.

WAP was designed to increase productivity and services and to increase the speed of installation of infrastructures, the simplicity and the cost thereof.

Because wireless technology is characterised by a series of advantages, such as portability and ease of use and installation, its disadvantages, such as security, are attacked in a violent manner by malicious entities.

Wireless handheld devices are characterised by a series of initial limitations such as the relatively small display, the limited processing capacity, the data entry devices that are not particularly easy to use, the relatively low passband.

The WAP forum was not directed at re-engineering products but at promoting standards for the technology used to develop applications, services and platforms that were able to operate through wireless networks. These standards were designed to ensure maximum interoperability between systems and allow the development of applications by third parties. WAP is able to significantly reduce the inconveniences related to the low passband available. The initial specifications of WAP were published in 1998 to create a global wireless protocol based on existing Internet standards such as XML and IP.

6.7.4.1 The ISO/OSI, TCP/IP and WAP models

The stack of WAP protocols is composed of:

1. application level;
2. session level;
3. transaction level;
4. security level;
5. level of transport.

The level of application, operating as a wireless application environment (WAE), is able to provide an environment to develop and run applications and services for handheld wireless devices and includes a microbrowser, an interface for a markup language and push technology to transmit data. It also provides multimedia capabilities.

A comparison between the TCP/IP, ISO/OSI and WAP models is shown in Figure 6.27.

The session level acts as a WAP Session Protocol (WSP) and manages the exchange of content. It is able to ensure the states shared between network elements through multiple requests and to manage negotiations to take advantage of the better client capabilities.

The transaction level operates as a WAP Transaction Protocol (WTP) and allows the transaction process, although with some doubts surrounding reliability. WTP provides streaming data, hypermediality and the transfer of messages.

The level of security, which is sometimes considered as optional, operates as a wireless transport layer security (WTLS) and guarantees authentication, confidentiality and secure connections between applications; WAP 2.0 provides the facilities on privacy, authentication, integrity checks and non-

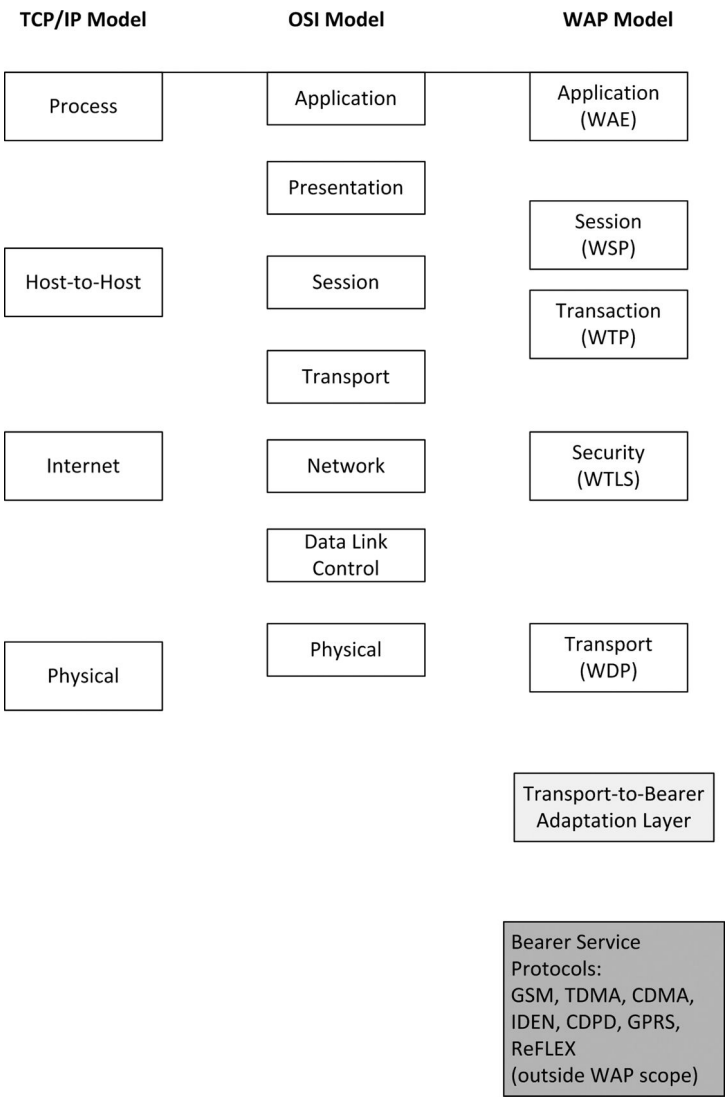


Figure 6.27 Comparison between the TCP/IP, ISO/OSI and WAP models.

repudiation capacities. The most important characteristics are the use of cryptographic libraries for signature and various forms of authentication on different levels. At the transportation layer, both WTLS and TLS are supported; identification and authentication of the user by wireless identity module (WIM); WAP 2.0 provides public key infrastructure (PKI) services enabled through the help of servers. Secure transportation takes place at the transportation level.

The level of transportation operates as a WAP Datagram Protocol (WDP) to protect the higher levels from services that belong to the operator and to provide a consistent set to be selected in a set of available services. Services are available outside of WAP and are those with which the protocol itself intends to communicate. The success of WAP lies in its ability to interact with these services.

6.7.4.2 The effectiveness of WAP

HyperText Transfer Protocol (HTTP) cannot be used with WAP due to the limitations of existing bandwidth. For this reason, WAP clients communicate through WAP gateways, which are devices that

are within the service providers to adapt wireless protocols into Internet protocols. Gateways translate the requests that come from wireless devices into Web standard requests using protocols defined by the WAP specifications. After this, the gateways operate as Internet clients, sending requests to the servers that provide the WAP content. The gateways adapt the format of the information requested and send it back to the wireless client.

A WAP configuration baseline is shown in Figure 6.28.

6.7.4.3 WAP security

A high level of security is required from WAP to ensure that a series of applications can be supported such as online currency conversions, email access, bank account transactions and stock exchange transactions.

It is precisely due to the sensitivity of these operations that security is of vital importance. The first bastion of WAP security is WTLS that ensures the confidentiality, authentication, non-repudiation and integrity of data but is partially vulnerable to viruses, worms, DoS attacks, Trojan horses and so on.

Some of the problems could be solved by using cryptography, but unfortunately, the portable wireless devices are not equipped with the considerable computing resources necessary for cryptographic applications.

Wireless communication inevitably requires a certain level of trust. Users must be able to count on the fact that the network that they use is equipped with adequate security measures. In addition to WTLS, there are two other concepts related to WAP: login and *white spot* security.

With regard to login security, also called calling line identification (CLID), it enables the server to identify the end user. Because there is no well-defined CLID method for these systems, the user alone must establish a higher level of security than normal.

With regard to white spot, this represents the brief moment of time between decryption of the signal and the subsequent encryption to be sent again. This activity takes place when the WAP terminal encrypts the information received and the WAP gateway decrypts it to send it to the network remote servers. This situation is represented schematically in Figure 6.29.

Until WAP 2.0 was developed, the security services of the layer were considered as optional and with the development of WAP 2.0, such services are considered mandatory.

Figure 6.30 shows the typical sequence of secure exchange of data in wireless systems and in wired systems.

Another significant problem is that of viruses, which, however, may not be very complex and detailed precisely because of the limited computing resources typical of portable wireless devices.

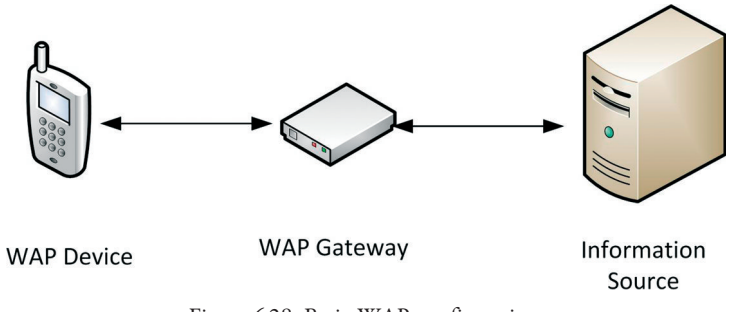


Figure 6.28 Basic WAP configuration.

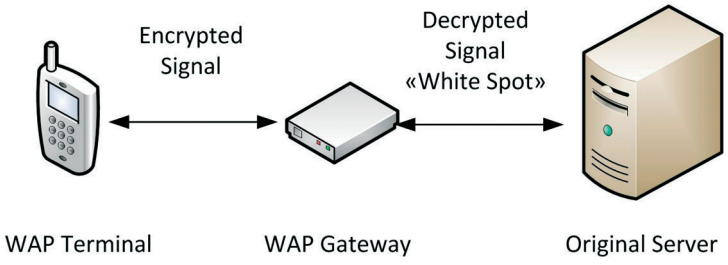


Figure 6.29 Schema of possible security solutions adoptable in WAP.

With regard to authorisation, this ensures that only authorised users can have access to information requested. Full functionality of the services requested depends, in large part, on the authentication mechanism and subsequent control procedures.

With regard to non-repudiation, this guarantees that the person concerned cannot deny having performed a well-defined transaction. To do this, it is necessary to use digital authentication and signature forms.

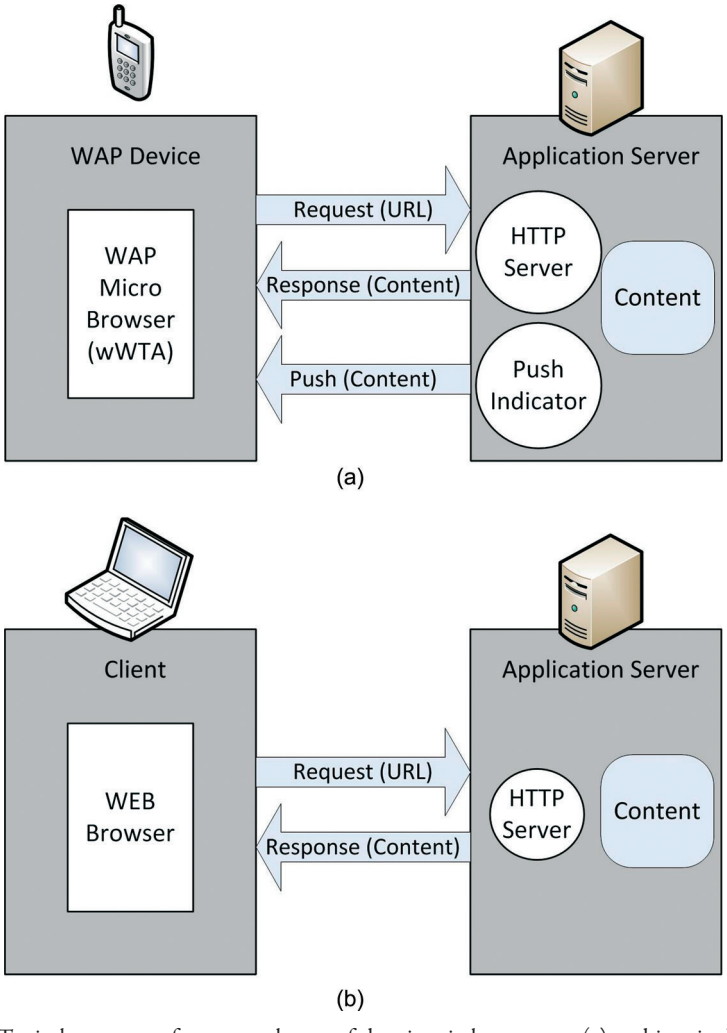


Figure 6.30 Typical sequence of secure exchange of data in wireless systems (a) and in wired systems (b).

With regard to authentication, this uses digital certificates.

The security of sessions can be ensured by using authentication and Secure Socket Layer (SSL) for WAP. The main vulnerability derives from the WAP gateway due to the problem of *white spot*.

Greater security is assured by wireless identity module (WIM) that has a function similar to the SIM. It ensures security features of the WAP terminal and is able to withstand sabotage in a manner similar to smartcards. The WIM is equipped with an encryption, a decryption and an authentication capacity.

A typical example of WAP system is shown in Figure 6.31.

Some applications are already equipped with internal encryption and, at the session level, SSL ensures message integrity, while PKI and certification ensure authentication and confidentiality.

6.7.4.4 Security architecture in WAP

The security architecture of WAP is divided into several models of levels. The most popular models are one- or two-level ones. A solution at an individual level requires a WAP-Web connection but appears to be too insecure, and for this reason the two-level architecture is preferable, with a gateway that is located between the WAP and ISP device. There is also a three-, four- and five-level architecture.

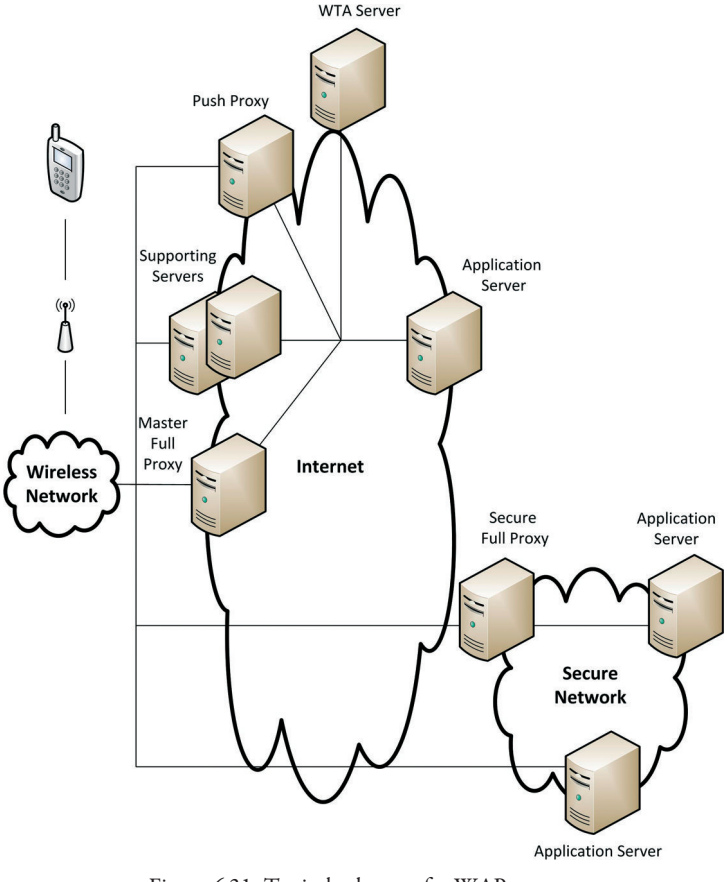


Figure 6.31 Typical schema of a WAP system.

6.8 Wireless antennae

All wireless devices require an antenna in order to be able to transmit or receive radio signals. Without antennae, wireless devices would not be able to perform any transmitting or receiving actions or in any case, it would make the activity very difficult.

This section analyses the basics of wireless antennae as an integral part of the same and the effects both on their functionality and on the relevant aspects of security.

6.8.1 Introduction to antennae for wireless devices

Understanding the type of antenna that is used and the relative mode of operation is of vital importance in order to fully understand wireless networks, the relative reliability and relevant security. Improper positioning of the antenna of an access point causes performance degradation in the area covered by the same access point. This section addresses the key concepts to help ensure the correct operation of wireless systems.

6.8.1.1 Polarisation

It has already been seen that electromagnetic waves, of which radio frequencies and microwaves used in wireless systems are part, are composed of two fields: electric field and magnetic field. These fields are mutually perpendicular and are characterised by an exchange of energy from one field to another during propagation that occurs through oscillation of the same. The position and direction of the electric field with reference to the surface of the earth is called wave polarisation. In order to ensure a correct and optimal exchange of energy between the antenna and the electromagnetic wave, the metal part of the antenna itself must always be parallel to the electric field. Understanding the concept of polarisation is of the utmost importance to guarantee an optimal coupling between the antenna and electromagnetic field in order to ensure maximum transfer of energy between the two and maximum efficiency of the wireless system as a whole.

As such, you have:

1. horizontal polarisation, when the electric field is parallel to the ground;
2. vertical polarisation, when the electric field is perpendicular to the ground.

6.8.1.2 Gain

An isotropic radiator is an antenna that radiates electromagnetic energy in all directions equally, both vertically and horizontally.

Given that, often, there is a requirement to send the electromagnetic energy emitted in very specific directions, particularly shaped antennae are used, characterised by differing gains in the various spacial directions.

The gain of an antenna is measured in dBi, a value that derives from the comparison of the behaviour of the antenna in question in relation to an antenna consisting of an isotropic radiator.

To have an immediate idea of the radiative behaviour of an antenna, the so-called radiation diagrams are used that characterise spatially the propagation directions of the antenna. An example of radiation diagram is shown in Figure 6.32.

It has already been stated that the gain of an antenna is often referred to as an isotropic radiator and is measured in dBi. A half wavelength dipole antenna can also be referred to and is measured in dBd. The two values are closely related numerically because if you want to have a value expressed in dBi starting from a value expressed in dBd, it is sufficient to add 2.14 to the value in dBd. The value in dBi is very important as it is used as reference by FCC to define the limits of the equivalent isotropic

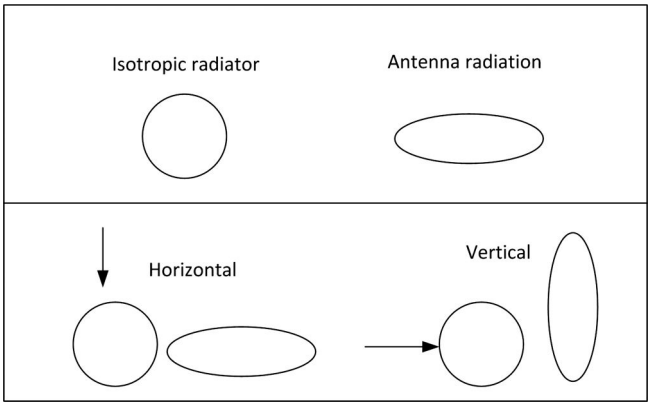


Figure 6.32 Example of a radiation diagram for an isotropic radiator and for a general purpose antenna (a). Radiation diagrams for horizontal and vertical polarisation (b).

radiated power (EIRP). In practice, manufacturers use values expressed in DBD to characterise their antennae.

EIRP represents the total effective power from a device. EIRP is a reference to limit the maximum power emitted by a wireless device as institutions such as FCC impose EIRP limits that cannot be exceeded by wireless devices. EIRP takes into account both the power output from the device and the gain of the antenna of the device-to-antenna connection system.

6.8.1.3 Beam width

The *beam width* is the width of the gain of an antenna. When the antenna radiation diagram is restricted, gain in the direction perpendicular to the restriction is increased. Two factors are associated with beam width: the vertical and horizontal factors. Vertical beam width is measured in degrees and is perpendicular to the surface of the earth. Horizontal beam width is measured in degrees and is parallel to the surface of the earth. Beam width is an extremely important factor to take into account in order to ensure the wireless coverage of certain areas.

6.8.1.4 Path loss

Path loss is the attenuation that an electromagnetic wave undergoes while it propagates in space. Path loss is very important when using the wireless solutions to provide bridges; in such a situation, it is of fundamental importance to take this into account in order to ensure that the signal reaches its destination with a width sufficient to be able to be detected.

6.8.1.5 Multiple Paths

It has already been stated that, very often, an electromagnetic signal is subject to various types of interference between reflection and refraction. When this happens, often the signal itself is divided into multiple signals that reach the receiver at different times due to the delay in covering different distances. This situation, called *multipath*, can cause significant problems for the receiver that has to do with the main signal and with a series of delayed replicas of the same that reach it. This situation is shown in Figure 6.33.

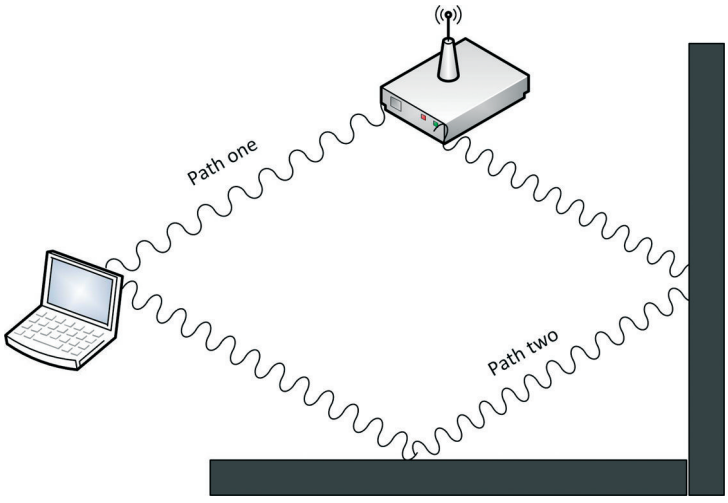


Figure 6.33 Example of multipath.

6.8.1.6 Diversity of antennae

To avoid multiple paths, a method called diversity is used. Diversity refers to the use of two antennae for each radio device, instead of only one. This mode reduces the effects of multipath, being able to choose, in turn, which of the two antennae receives the best signal. This means that the antenna used varies from time to time. The two antennae are physically separated even if they are of the same type. The distance between the two antennae must be appropriate and is usually indicated by the manufacturer of the device. This situation is shown in Figure 6.34.

6.8.2 Fresnel zone

The Fresnel zone is the area around the wireless signal that must be free from obstacles in order to ensure the best conditions for propagation of the signal itself and, therefore, the best reception. The

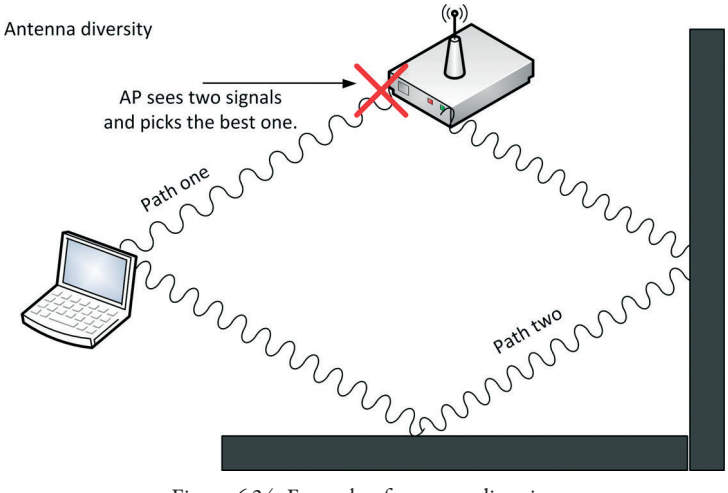


Figure 6.34 Example of antenna diversity.

wireless signal, depending on the frequency and type of antenna used, can occupy relatively extensive portions of space and any fixed or mobile object can be within that space, interfering with the signal propagation. When wireless bridges are mainly involved, which require an optical range, it is very important to take into consideration the Fresnel zone, which, if occupied by obstacles, can cause serious problems for the propagation and reception of the signal even if the transmitting and receiving antennae are in line of sight. The Fresnel zone is also influenced by the curvature of the earth.

6.8.3 Types of antennae

There are several types of antennae. In the wireless industry, there are substantially three types of antennae:

1. directional antennae;
2. omnidirectional antennae;
3. home-built antennae.

These types of antennae are discussed below.

6.8.3.1 Directional antennae

Directional antennae are characterised by a radiation diagram mainly directed along a main direction. This characteristic is very difficult to achieve because antennae, by their nature, tend to radiate in all directions.

Directional antennae are used when a narrow beam must be generated but that must reach relatively long distances, as in the case of long corridors. The most common directional antenna is the Yagi antenna, which is characterised by a beam width of 17° and has the shape of a triangular cone.

In the creation of bridges for long distance, parable antennae are usually used. In these antennae, the beam width and the shape of the main lobe is very similar to the Yagi antennae with the difference that in this case, the signal can reach a greater distance with the same input power into the antenna.

Another very popular antenna is the patch antenna, which represents a middle path between the directional antenna and the omnidirectional antenna, as its beam width is equal to 180°. This is also the reason why the patch antenna is applied to walls, ensuring that all the electromagnetic energy radiates from the opposite side in relation to the same wall, thus avoiding unwanted signal radiation outside of the building in which the antenna is installed.

6.8.3.2 Omnidirectional antennae

Omnidirectional antennae are differentiated from directional antennae because of the beam width and shape of the radiation pattern. All omnidirectional antennae are characterised by a range of 360° because the same radiate electromagnetic energy in a uniform manner across all the space that surrounds them.

Omnidirectional antennae are suitable to be used in wide open spaces, where it is necessary to ensure uniform and homogeneous coverage. If an omnidirectional antenna is applied to a wall, the wall itself tends to hinder the spread of the antenna within the same, nullifying the properties of the same omnidirectionality.

6.8.3.3 Home-built antennae

Home-built antennae have found a great diffusion due to the relatively high price of professional antennae. They are, for the most part, used by the hackers who carry out the activities of *war driving*, war *walking*, etc., shown above.

The Internet contains a huge number of sites that show how to build an antenna using materials that are readily available such as jars of known commercial products, which provide physical dimensions and construction materials that are well defined, and thus the possibility of assembling antennae characterised by the same properties.

6.9 The implementation of wireless networks

The process of installing a wireless network is relatively simple when a single implementation is involved. When it is a matter of an entire organisation, this process must be performed by following well-defined procedures, thus avoiding the risk of implementing a network characterised by low quality, reduced performance and a low level of security.

This paragraph explains the basic steps to be followed in order to create an efficient network, maintaining costs and ensuring a high level of security.

The first decision to be taken, when a large-scale wireless network is to be created, is to decide whether to implement the same inside or outside the organisation. When an external subject is opted for, the risks are transferred from the inside to the outside, but ultimately responsibility for the final outcome always falls with the organisation. If the outsourcing process is managed appropriately, every financial risk is suitably placed on the external subject, ensuring the success of the required result with the estimated costs. This approach is often difficult to apply for relatively small organisations, whereas it is easily manageable for large organisations.

If there is doubt surrounding the objectivity of the external subject selected for the development of the wireless network, a consultant can always be hired, who can also be an employee of the organisation who has all the technical expertise of the sector and is updated on all the products available on the market. This consultant is a third party who is impartial and who acts as a mediator between the needs of the organisation and the needs of the external developer ensuring high quality of the final product. If the consultant is able to analyse the project and the devices to be installed, he is able to guide the external developer towards creating a quality wireless network, from the viewpoint of both performance and security.

6.9.1 Requirement acquisition

First, it is necessary to understand the reason why an organisation aims to implement a wireless network. In most cases, the reason is clear, an increase in productivity and efficiency of the same organisation.

Subsequently, it is necessary to understand where the wireless network is most needed, and to try to establish whether it is necessary to extend the same network in all the places where the organisation operates, or if it is sufficient to install it in well-determined spaces. It is also important to understand the business needs of the various subjects that operate within the organisation to be able to develop a wireless network to meet their needs.

A very crucial need to consider is to understand if the wireless network should possibly support a video surveillance system, whose cameras require a high bandwidth to ensure the vision of a video stream in real time.

Once all the needs of the organisation have been established, it is possible to evaluate the requirement to provide a wireless network and the places where such a network is actually required. Other factors of vital importance are the time during which the creation of a wireless network is required, the final cost and the implementation mode.

6.9.2 Cost estimate

Once the needs of the organisation and the technical aspects have been established, the costs can be estimated. When it involves a wireless network, costs may fluctuate. Even in buildings designed by the same designers and produced by the same manufacturer, there may be totally different behaviours with regard to the propagation of electromagnetic waves.

The estimation of costs is the second step required for the implementation of a wireless network. This is due to two main reasons. First, before having an idea of costs, it is necessary to perform a thorough analysis of the site that, in turn, requires time and resources; this means that there are additional costs which are added to the costs incurred for the construction of the network itself. Second, it is not possible to make a decision if there is a lack of clarity surrounding the costs involved.

The process of estimating the costs necessary to create a wireless network can be very challenging to perform for a variety of reasons. The best and easiest way to perform this is dividing the area to be covered by the mean area of coverage of a single access point in order to know the number of access points, on average, necessary to ensure uniform coverage of the entire site. It is very important to remember that the coverage area of a single access point varies with the wireless technology to be used for the access points. The choice of technology to be used is a result of an assessment of the needs of the organisation, an operation that must be performed as the first phase of development of a wireless network. Thanks to this operation, it is possible to have a summary estimate of the costs. Even if this operation does not provide a precise figure, it provides, however, a first estimate of the costs that can be used as a reference for the assessments of feasibility before proceeding with the next step.

In the phase of cost estimating, to ensure precise achievement of the result, it is necessary to take into account other additional costs such as the fixed network, which serves to connect the various access points and the cost of labour required to install the access points themselves. Many additional costs could become evident during the design phase, raising the total cost of the wireless network; for this reason, it is necessary to try to take into account all the possible costs at the phase of economic evaluation, avoiding surprises at the subsequent phases of development of the same network. Therefore, the support of an industry expert that has some experience in the implementation of wireless networks and that can ensure having taken into account all the factors that contribute to the final cost of the network is suggested. To avoid surprises due to the fact that the access points have performance that is reduced compared to those stated by the manufacturer, the purchase of one or more access points of the technology and of the desired manufacturer is recommended to be able to perform testing directly within the various areas of the site of interest of the real performance of the same access points. Once the real performance has been measured, including the coverage area, it is possible to calculate the actual number of access points required, dividing, as already seen before, the total area to be covered by the effective area of coverage of the single access point.

6.9.3 Evaluation of investment

Evaluation of the investment is made by the respective managers to whom must be explained the usefulness of the wireless network, as the latter are not technicians. The key element is definitely the increase in productivity. In this sense, statistics can be used to increase productivity in the various sectors that are easily available everywhere and that have been certified by the major organisations in the sector.

A key point is the so-called return on investment (ROI), which quantifies how the wireless network can also produce a reduction in labour costs and increase productivity, as employees are always connected to the network and to one another and to the organisation wherever the same may be within it. There are many advantages of wireless networks, and to facilitate quantification of a reduction of the costs in ROI, it is necessary to distinguish the benefits from cost reduction, or the increase in production and to do this, it is possible to refer to data and statistics available to the public and to the organisations that operate in the same area, making evaluation of the investment reliable and well informed as it is based on objective parameters and tested by others.

Among the advantages of the creation of a wireless network, it is also possible to consider an increase in security, as long as the wireless network is well designed and structured from this point of view. In this sense, in addition to the intrinsic security of the network, the same network allows the support of security systems, such as video surveillance, that can reach in capillary fashion every area of the organisation, even the most remote ones.

6.9.4 Site analysis

Once the project of creating a wireless network has been approved by the management, it is possible to proceed with the first phase of the entire manufacturing process, as to this point, intellectual type work has been performed.

In this sense, it is necessary to perform a thorough inspection to acquire all the fundamental elements that represent the input data for design of the wireless network.

When analysis of the site is being performed, the essential requirements must be complied with. First, it is necessary to use the suitable analysis devices, that is an access point, of the type to be installed on the site, to use a set of different antennae, a portable battery and a vertical mast necessary to place the antenna on the same level as the various access points that make up the wireless network. This equipment is used to test the various types of antennae in various areas in order to find the type of antenna most suitable to ensure the maximum coverage in various areas of the organisation. At this phase, it is also necessary to have the planimetric drawings of the site on an appropriate scale. Furthermore, it is very important to have a spectrum analyser to check whether there is a particular interference within various areas of the site. If a bridge solution is being considered or it is necessary the coverage of very extensive external areas, the use of a GPS device for correct referencing of the external parts of the site can also be opted.

6.9.4.1 Execution of site analysis

When analysis of the site and its on-site inspections are being performed, a series of points must be respected. It has already been stated that it is absolutely essential to be equipped with the appropriate tools in order to be able to perform the assessment tests directly in the field.

The key element is the access point that must be the same model as the one to be installed on the site. If a model different from the one to be installed is used, the problems of coverage may be checked at the time of practical implementation of the network. Before choosing the model, it is necessary to select a wireless technology; all the major manufacturers now have products that are able to operate with all the wireless technologies.

The next step consists of powering the access point, which is a critical step because it is inconceivable to use a power cable that is able to follow the access point during its movements within the site to perform analysis and evaluation of the same. For this reason, two equal and charged batteries must be used; the use of two batteries always allows certain operational autonomy, one battery being always charged, whereas the other is operative and vice versa.

Another very important object is a telescopic pole that is used to hold together the access point, the antenna and the connection wiring. The height of this pole must be adjustable so as to be able to place

the access point and the relative antenna at the actual height of installation. These poles are normally available on the market. The pole should be lightweight and telescopic and characterised by a bright colour to avoid treading on it when the same rests horizontally on the ground.

It is also necessary to use a set of antennae that allow assessment of the different areas of coverage of the access point in the various areas of the site. It has already been seen that there are different types of antennae for access points, each having different performance and characteristics. The use of too many types of antennae should be avoided because this could cause problems in the management of the maintenance of a wireless network that uses a multitude of different antennae. For this reason, it is necessary to select three to four types of different antennae for each of the main types: patch, omnidirectional, Yagi and ceiling mount. This ensures maximum flexibility without dramatically increasing complexity.

Another device to be used during the analysis operation of the site is a final wireless device, which can, for example, be a normal laptop with wireless capability. Such a device should be of the same type as that used by employees of the organisation for normal work activities. This is not always possible but is recommended to obtain the best final results. While using a wireless card connected to a laptop, it should be ensured that the same is not characterised by high/low emission power and high/low sensitivity and instead it is characterised by average performance.

It is also necessary to have available maps of the site where the exact positioning of future access points can be indicated, reporting with precision the exact point of installation and relevant height.

An object very useful at this stage is a spectrum analyser that is used to check any type of interference that might interfere with normal operation of the wireless network in certain areas. In fact, because wireless networks use dedicated and free frequencies that do not require any authorisation by government bodies, the same frequencies can also be used by other devices, crowding the band and degrading the performance of devices that operate on it. It is not necessary to use excessively expensive spectrum analyser models; even the simplest and least expensive models are able to show the status of the radio frequency band that is to be used to provide the wireless network at the site in question.

The team that performs analysis of the site should be composed of at least two people. The work should be planned beforehand; rather than moving around the site with all the instrumentation with no defined program, it is advisable to establish in advance the various areas where access points and antennae are to be installed, and then to go to these areas and check what has been programmed. The team, once the various estimated points have been achieved, activates the access point connected to the first antenna and then performs testing of coverage with the wireless device, ensuring not only the presence of the field but also that the normal activity of transmission/receipt is taking place. This verification must be performed for the various antennae available to find the one that is most suitable for the area of interest at that moment. Once the area of coverage for a given antenna has been identified, this area must be reported with accuracy on the site map, together with the exact point of installation of the access point and its height. It is very important to show on maps both the levels of power emission, if it is adjustable on the access point, and the channel used. The best channel must be verified by the spectrum analyser, choosing the frequency that is most free from interference.

Once this operation has been performed for all the areas of interest at the site, it is necessary to convert paper-based information into electronic information, which facilitates its processing, reproduction, transmission and storage in order to be always available in the future.

The documentation must therefore indicate:

1. location of the access point;
2. antenna type used;
3. positioning of the antenna;
4. emission power of the access point;
5. channel used.

With regard to the positioning of the access point, this information is of vital importance to allow the access point itself, once that element of the wireless network has been positioned, ensures the same coverage conditions that were observed at the analysis stage; displacement of the point by only a few tens of centimetres can vary, even considerably, the coverage of the area of interest. For this reason, this information must be well defined at the analysis phase and reported with precision on planimetric drawings.

With regard to the antenna, the two pieces of information concerning the type and positioning must be indicated together. This is very important because each antenna must be mounted in a well-defined manner to ensure optimal coverage. For this reason, the above information must be provided with precision in the planimetric drawings, allowing for the same coverage conditions found during the analysis phase to be repeated during installation.

With regard to the output power and the channel to be used, these are of fundamental importance to have a certain radius of coverage, with a particular quality of the wireless signal.

In most cases, manufacturers do not allow for the adjustment of the output power, producing devices that emit at maximum power. To ensure the maximum yield of the wireless network, it is recommended that devices featuring adjustable emission power are used. In this way, it is possible to adjust precisely the area of coverage when zones characterised by a high number of access points installed contiguously with one another are involved.

6.9.4.2 Technical standards

Technical standards are also very important. An initial standard is represented by the possession of all the technical skills by the staff to avoid neglecting details which, even if at first sight may seem insignificant, can lead to significant problems of yield for the wireless network once the standard has been implemented.

6.9.4.3 Financial controls

From a financial point of view, it is very important that the specifics of the network are correctly defined and detailed before dealing with the actual network provider. These specifications are the area of coverage, the maximum speed required, the quality of the signal, the reliability of the network, etc. If these specifications are identified at a preventive phase, surprises can be avoided following the implementation of the wireless network and disputes with the provider of the network will also be avoided.

The provision of specific guarantees for correct analysis of the site is always a critical element as boundary conditions may vary between the analysis phase and the phase of the wireless network, as well as influencing significant performance demands. In this sense, it is very difficult to assign liability in the event a certain degree of interference becomes evident, in a certain area, which was not present at the time of analysis of the site and which, however, degrades the performance of the wireless network in the above-mentioned area. In this sense, the presence of an external and third-party consultant is very important, who, with their experience, can verify the correctness of the implementation of the operation of analysis and justify that such events may happen unpredictably.

Most providers offer a limited warranty period, and there is a risk that once the warranty period has expired, the environmental conditions could vary, and interference not found during the analysis phase may appear that could affect the performance of the wireless network. For this reason the terms of the warranty should be studied with precision before signing a contract with the supplier. If this research is performed correctly, the supplier becomes responsible, together with the site analyst, for the entire financial obligations relating to the design, installation and certification of the same network, guaranteeing the addition of any access points where, at the testing phase, there is insufficient coverage.

It is also very important to prevent excess coverage, avoiding the wireless network from extending beyond the site, exposing the network to possible attacks external to the site.

6.9.5 Network design

The design of the network uses the information collected during the site analysis and translates them into project, determining the structure of the wired network that will serve all the access points required. At this stage, if there is already a wired network, and it is believed that this network can be used without degrading its performance, a wired network can be used, and the design work involves finding the easiest path for connection of the various access points with the closest wired network connection node.

It is also necessary to define the IP addresses that will be used by access points within the wired network and all the other parameters.

In this phase, the various paths are identified that must follow the cable to connect the access points to the fixed network and the positioning of any other communication devices needed for support, by the fixed network, of the wireless network services.

6.9.6 Device verification

Device verification can be carried out both on-site and at a central location.

At this stage, all the devices must be activated, set up according to the specifications identified during the analysis phase and their operation tested. Once each individual device has been tested, it is possible to proceed with a collective test. In this way, it is possible to verify the functioning of the entire network in normal conditions and under conditions of malfunction of one or several parts of it.

6.9.7 Development and installation

Once the various devices have been tested, the network can be installed. Since, in most cases, equipment must be added in the rack cabinets containing all the network devices, it is very important to verify at the design phase that there is still space available for the installation of new equipment in these rack cabinets.

With regard to the installation of access points, they must be installed in the same location identified at the analysis phase to ensure the same performance obtained during this phase. If this positioning is not carried out with precision, there is the risk of having coverage less than, or in any case different from, that quoted at the analysis phase. In many situations, to avoid errors during the installation phase, the site analysts leave appropriate well-highlighted markings at various positions of installation in order to allow installation of the access points, during the implementation phase, in the same position as the access points used during the analysis stage. The installation of access points must be performed by competent personnel, who are able to install the same and properly direct the relevant antenna. The information relating to orientation of the antenna must be generated during the analysis phase and must represent the information that is transmitted, with accuracy, at all stages of development of the same network in such a way that the above information is available with precision and clarity at the final installation phase.

At the installation phase, it is necessary to pay particular attention to avoid making very common mistakes such as affixing the omnidirectional antennae to a wall as the same should be installed at a certain distance recommended by the manufacturer to maximise their properties to irradiate the electromagnetic signal uniformly in all directions. Another mistake committed is the orientation of directional antennae. In fact, antennae such as Yagi are characterised by a well-determined radiation diagram that must be addressed carefully to ensure the desired coverage and must be verified during the

analysis phase. The same caution must be applied to patch antennae, avoiding installing them upside down. Another problem is the obstacles which may have been introduced at a later stage of the activity of analysis that can significantly alter the coverage of the area in which they are present. Another error is leaving an excessive amount of connection cable for the access point; this cable may form an induction ring with respect to the signal emitted from the same access point and altering the wireless coverage area.

6.9.8 Certification

Once the wireless network has been properly installed and is operational, it must be certified by a third party, different from the site analyst and the installer. This choice ensures that the certifier acts with impartiality, identifying any errors made during the site analysis phase and during the installation phase. The tester can be an external consultant or an employee of the organisation who possesses all the technical skills and experiences needed to perform this type of activity. The certifier must identify any gaps in coverage or any shortcomings in the wireless network as a whole. This phase is also used to verify that all the access points have been installed at the points provided for at the analysis phase, to attribute with certainty any deficiencies to the site analyst or the installer of the wireless network. Certification is performed including all the actors that were involved in the development activities of the same network. In this way, any problem that is easy to solve, such as the incorrect installation of an access point in the location provided at the analysis stage or the incorrect orientation of an antenna, can be resolved at the time, avoiding long and troublesome disputes.

At the certification phase, it is verified that all the coverage areas identified at the analysis phase are adhered to through the use of wireless devices that exchange data directly with the network. Subsequently, load tests can be performed, simulating a condition of maximum presence of users connected to the network, in such a way as to verify full-load performance.

6.9.9 Audit

The audit phase is very similar to the certification phase, in the sense that it is substantially a verification phase. The purpose of the audit is to verify that there has been no activity that has exceeded the standards of the organisation where the wireless network was installed and also to ensure that the network itself meets the security standards required by the organisation. The audit process also verifies that the network has not been altered in any way following its implementation and that it continues to conform to the standards of the organisation. Very often in peripheral sites of the organisation, employees tend to comply less with corporate policies to make their work easier and faster, very often neglecting security aspects, exposing dangerous flaws in the systems. For this reason, the audit phase is a critical and essential element for ensuring the network always maintains the standards that comply with corporate policies and that always ensure a high level of security.

6.10 Wireless devices

Wireless devices are divided into two large families:

1. access points;
2. mobile user devices

6.10.1 Access points

Access points, as has already been seen, are the heart of wireless networks. These devices allow the connection of mobile devices with the fixed network, ensuring wireless service in their coverage area.

There are different wireless technologies, already described in detail, and different equipment manufacturers that use these technologies. The choice of technology to be used depends on the objectives to be achieved in terms of speed of data exchange, coverage area, number of concurrent users, etc. There are, in any case, access points that are able to use different technologies at the same time, ensuring high interchangeability of devices that operate within their coverage area.

There are also access points designed for small offices and home and access points designed to serve large organisations.

6.10.2 Mobile user devices

Mobile user devices are the most varied and the most popular ones are mobile phones, due to the widespread distribution of cellular telephony.

However, there are many other devices that use the wireless network, such as laptops, tablet PCs, handhelds, various peripherals (printers, scanners, etc.) and dedicated devices for mobile applications.

Most wireless devices are mobile: this means that they are always exposed to the risk of theft and/or loss, with all the consequences this entails, first, the loss of the data contained in them and all the implications of security if the data are confidential data and have not been stored in a protected manner within the devices. For this reason, it is advisable to always use the dedicated cryptographic programs that encrypt the data contained in these devices, making it indecipherable in case of theft and/or loss.

6.10.2.1 Laptops

Portable computers, or laptops, were among the first user devices of wireless networks. They are preferable to desktop computers, or desktops, because they are now marked by the same performance but are smaller, portable and affordable. Once the wireless technology took root and began to be used almost everywhere, portable computers took great advantage of it in terms of the connectivity services offered.

Initially, laptops had no wireless connectivity and, for this reason, it was necessary to use dedicated cards. Nowadays, practically all laptops are equipped with multi-technology wireless connectivity so as to be able to connect to any access point characterised by different technologies.

Portable computers are exposed not only to the normal risks of use of wired networks but also to the risks of use of wireless networks.

An example of risk is the use of a hot spot, which is a dedicated zone where there is a wireless service, generally available free of charge or for a fee. It has already been stated that the level of security is greatly increased with the development of wireless technology and the first technologies were characterised by a relatively low level of security. If the technology used in hot spots is quite dated and your laptop is able to accommodate it, you are exposed to significant risks in communication as the transmitted data can be intercepted, handled, stored, etc.

Another risk is the non-use of cryptography in hot spots. In this case, transmissions take place in plaintext, and the traffic can be easily intercepted and used fraudulently (sensitive data, credit card numbers, username and password).

For this reason, it is very important to know the type of wireless technology used by your computer and its associated security algorithms that the same technology uses to ensure a high level of security of its communications.

6.10.2.2 Tablet computers

Tablet computers represent a product that is experiencing remarkable success and dissemination. They are a middle way between the laptop computers and personal digital assistant (PDA, see

section 6.10.2.3) and are characterised by high functionality in a device that interacts with the user directly via commands that can be activated on the screen. They are very useful due to their small size, reduced weight, high functionality and the ability to be used without a keyboard. This makes them very useful in all situations in which a determined job needs to be followed on the move. Think of the case of implementation of warehouse inventories or medical staff in a hospital that can act directly on the electronic folders of patients moving fluidly within departments.

As the tablet is a laptop computer, it is exposed to the same risks as the latter. A further problem of tablets is the possibility of signing documents by writing directly on the screen. This means that within the same is a digitised version of your signature that in the event of loss can be used, if not protected, to be affixed to other documents that may be signed by simply resorting to the digitised signature. This risk is not obviously linked to the wireless industry but is intrinsic to tablet computers.

6.10.2.3 PDA Devices

PDAs were introduced on the market around 1990 and were primarily intended for managers. Around 2000, these devices were equipped with additional features that made them more attractive to the general public. One of the features introduced is wireless communication that made the development of a considerable quantity of applications that require connectivity possible.

All PDAs work with dedicated hardware that allows, among other things, the use of different wireless technologies.

The market now offers a considerable number of PDAs and this can have positive and negative effects. With regard to the positive aspects, it must be pointed out that the great availability of devices allows one to be found that is suitable for individual purposes.

In relation to the negative aspects, it should be emphasised that within the same organisation, employees can use different PDA, posing serious problems regarding the use of application programs that need to be shared at a general level. This is why increasingly more support is being requested, and the assistance of staff of the information and communication sector in the management of business software on different PDA is becoming a very expensive and complex operation.

6.10.2.4 Portable scanners

Handheld scanners, which are very closely related to barcode readers, were among the first devices to require wireless communication, given their need to operate in mobility, and were produced before the standardisation of wireless technologies currently in use.

Older models operated at a frequency of 900 MHz and were not compatible with the 802.11 technology, whereas newer models operate in accordance with 802.11.

The great problem with these devices is the life of the battery, which is a particularly critical element, when the device must support and supply an 802.11-type wireless technology. This problem is especially apparent when wireless technology must be used that uses advanced encryption, as consumptions rise inexorably due to the fact that there is additional consumption owing to the implementation of the necessary encryption operations.

6.10.2.5 Smart phone

Smartphones have become extremely popular devices used by every age group. They can be used not only as mobile phones but also as an electronic diary, a Web browser and as a small computer, and everything takes place on the pocket devices.

Given their high functionality, some devices are able to integrate the functions of a PDA and in this case, reference is made to PDA phones.

There are numerous models available as well as there being many operating systems used by these phones. Also an extremely large number of applications are available, which can be downloaded from the network, capable of performing a wide range of functions.

It is clear that since these devices use wireless networks, they are exposed to risks. The risk increases with the number and type of wireless networks that they are able to use (cellular, Wi-Fi, Bluetooth, etc.). As all wireless technologies are characterised by specific risks, the device is exposed to multiple risks. For this reason, it is very important that the user possesses a moderate knowledge of wireless technologies used and of the respective security threats that the use of each of these technologies involves.

One of the greatest risks is viruses which, if they were originally designed to strike only computers, are now also designed to attack handheld and mobile devices. The first viruses were transmitted from computers to mobile devices during synchronisation operations that are usually carried out to duplicate the archive of the device in a fixed computer. These viruses are often intended to force the phone to make phone calls or to connect to the Internet. For this reason, many smart phones have already integrated an antivirus program which must be constantly updated to withstand the new threats that develop on the horizon and due to the development of new viruses. Lately, Trojan horses have been developed that are inserted within applications; once installed and the application is run, the Trojan horse becomes operational and can perform very varied tasks, from the mildest up to the most destructive activities.

Another feature that is increasingly integrated into smart phones is the GPS receiver. The built-in GPS smart phone is not a real GPS system but a hybrid integration. In fact, the GPS is not able to work correctly in indoor environments due to the difficulty of receiving a signal from satellites. The smart phone, on the contrary, is able to operate properly in indoor environments by resorting to a technology called GPSone that engages not only the satellite signal but also the signal of wireless networks, which also contains information about GPS. The smart phone is able to use both signals to produce the location information.

One of the biggest advantages of the integration of GPS technology is the possibility of locating the telephone in case of emergency. Of course, there are also navigation applications and other related applications such as the georeferencing of photographs and videos that are taken by the smart phone itself. The risks of using a GPS system on a smart phone are the same risks related to the use of a GPS device and that have already been discussed.

In addition, it should be remembered that, as smart phones are able to surf the Internet, they are exposed to the same risks of a fixed computer that surfs the Internet and that has been widely shown above.

6.10.2.6 Wi-Fi phones

Wi-Fi phones in fact use the Wi-Fi network to make voice phone calls and of course require an adequate bandwidth and an adequate signal quality to be able to make intelligible calls.

Many new phones implement Wi-Fi communication capabilities in such a way as to be able to change from the most expensive cellular network to the least expensive Wi-Fi network to make calls.

The problem of Wi-Fi phones is security implementations whose procedures unavoidably slow down the voice signal that must be processed, possibly leading to excessive delays and making it incomprehensible.

6.11 The security of wireless LANS

Wireless communications, by their very nature, have always been exposed to security risks even before the advent of the family of 802.11 technologies. In fact, the signals are emitted in space and are easily

subject to interception. Over the years, encryption systems were initially implemented, which were not always suitable for the situation.

This section discusses the various technologies used to ensure the security of wireless networks, starting from the first technologies implemented up to the latest technologies.

6.11.1 History of wireless security

Different attacks have been conducted in relation to wireless technology as the same has gradually developed over time.

This section analyses the methods of wireless security that have been created, violated and improved, helping in the understanding of the current methods of security.

Already by 2000, the first articles pointing out the security problems typical of wireless networks were beginning to appear. These articles also highlighted that the increase in length of the WEP key (which will be described later in this chapter) from 40 bits to 128 bits was not sufficient to protect the communication from certain specific threats.

In 2001, another article highlighted how it was possible to attack the 802.11 standard by resorting to cryptoanalysis in plaintext, which requires the knowledge of part of the plaintext and the corresponding encrypted version; by comparison and analysis of the two parts, it was proved possible to refer to the key used for encryption. In wireless networks used during that period, this key was used for the majority of communications through the network.

In 2002, an article was published on the security of the 802.11x standard, which presented the problems arising from the non-use, in such a standard, of two-way authentication. This lack makes it possible for hackers to use a man-in-the-middle attack to breach such networks.

In 2003, an article was published that analysed the WEP encryption method and demonstrated how it was possible to breach it. This showed the weakness of WEP and the need to replace it with algorithms that were more secure and most robust against attacks.

Over time, several other articles were published; all intended to demonstrate the security vulnerabilities of wireless networks. These articles have stimulated research and the development of increasingly sophisticated technologies characterised by a greater level of security.

6.11.2 Authentication

It has already been stated that when accessing a wireless network, authentication must be performed. The IEEE provides two types of authentication:

1. shared key authentication;
2. open key authentication.

6.11.2.1 Shared key authentication

Shared key authentication should be more secure than open key authentication but a small criticality in the manner in which the user key is validated makes it less secure. The operation of this type of authentication is explained below.

This mechanism uses the mechanism of response to a challenge. The first phase involves a connection to the network through the transmission of a probe frame by the wireless client. This frame is intended to search for available wireless networks and the associated transmission parameters. Once an access point receives this probe, it responds with a response frame to the probe that serves to inform the client of the communication parameters used. As a given area can be covered by more than one access point, to be sure that the client connects to the access point characterised by the stronger signal, the current value of the signal is indicated in the response frame. In this situation, the client receives the

responses of the different access points but only connects to the one that is characterised by a stronger signal. Once the client receives the response and verifies that the access point is in accordance with their own communication parameters, connection is made between the client and the access point. At this point the authentication phase begins.

At the beginning of authentication phase, the client sends a response authentication frame to the access point. This frame is evaluated and when the access point realises that it contains an authentication request, it sends a challenge packet to the client involved in the process. The challenge package contains a plaintext string of data; this packet must be encrypted by the client with its WEP key and sent back to the access point. The access point, once the response from the client has been received, checks the encrypted result and if such a result corresponds with what is expected, it allows the client to access the network. If the result is not the expected one, the authentication process fails and connection to the client is denied. Figure 6.35 shows the various operational stages of the process of connection and authentication.

6.11.2.2 Open key authentication

Open key authentication was considered less secure with respect to the process of shared key authentication. The aim of this process is to allow the construction of an open network that does not require knowledge of a WEP key.

The authentication process always begins with a probe request, a response by the various access points and determination of the access point characterised by the stronger signal. Open key authentication differs from the shared-key authentication in that the access point does not send a challenge to the client requesting a connection and connection is allowed once the initial phase is completed. If, in this type of authentication, WEP mode is enabled, then the client uses WEP encryption for all the data used in the communication. When the access point receives the data, it decodes that and sends over the fixed network. If the frames are encrypted with a key different from the

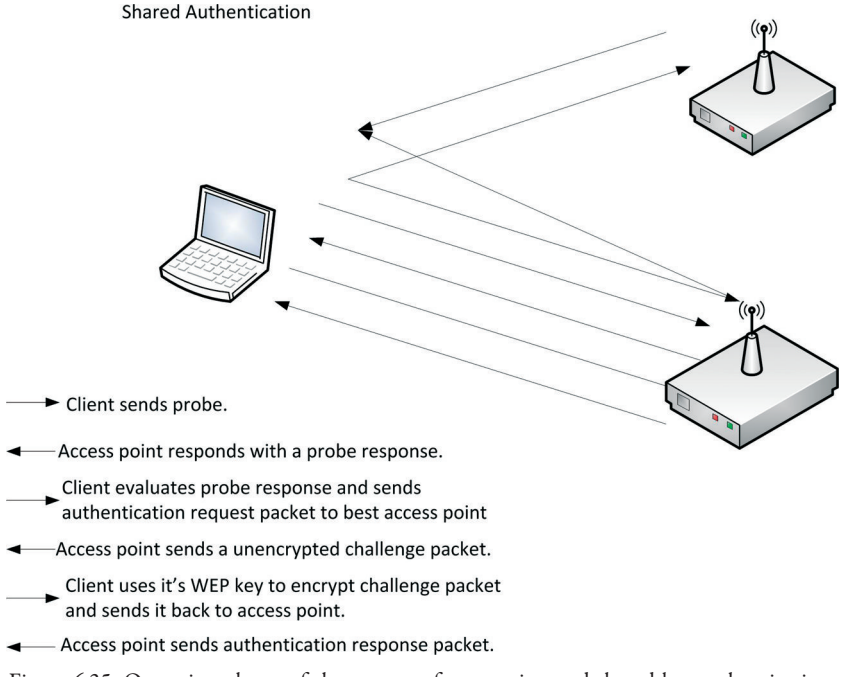


Figure 6.35 Operating phases of the process of connection and shared key authentication.

one used by the access point, then the decryption process provides incorrect results and the packets are lost. The open key authentication process is shown in Figure 6.36.

6.11.3 SSID

Most wireless networks transmit in broadcast the information relating to SSID. When the security aspects first started to be taken into consideration, SSID began to be inserted into beacon frames, obtaining as a result the masking of access points. In most wireless networks standardised by the IEEE, information relating to SSID can be acquired using traffic *sniffer*s. Even if SSID is masked, every time a client wishes to communicate, the access point sends all of its settings, including the information relating to SSID, as provided by the connection process.

Most of the producers insert a default SSID at the time of manufacture of the device.

6.11.4 Foundations of wireless security

This section addresses the basics of wireless security that allow for a better understanding of the encryption methods and advanced security that were subsequently introduced.

It has already been stated that the transmission of SSID is a vulnerability from the security point of view and, for this reason, transmitting it in broadcast is discouraged. Changing the default parameters, such as username and password in all of the access points, is also strongly recommended. In some products, there is also a default WEP key that is strongly advised to be changed. Other recommendations are to avoid Dynamic Host Configuration Protocol (DHCP) and to segment suitably the fixed part of the network that connects the wireless network. Another recommendation is to suitably inform users of the wireless network on security policies that describe accurately what can and cannot be done while using a wireless network.

6.11.5 WEP

WEP stands for *wired equivalent privacy* and is a standard to allow wireless networks to reach the same level of security as wired networks. WEP is a cryptographic standard that aims to provide a level of confidentiality of data equivalent to a wired network that does not use encryption for data

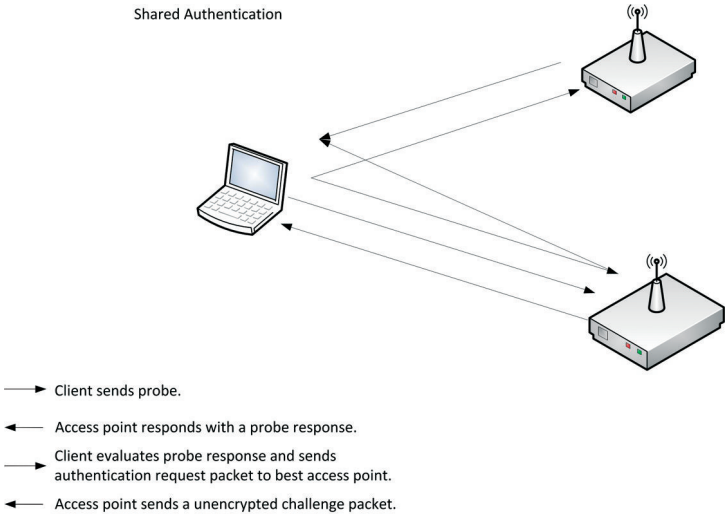


Figure 6.36 Operating phases of the process of connection and open key authentication key.

transmission. To reach this objective, WEP aims to ensure three fundamental points of security: confidentiality, availability and integrity. As WEP is used to prevent eavesdropping, it focuses on confidentiality. As WEP is intended to ensure only authorised logins to the wireless network, it is aimed at availability. As WEP is designed to prevent sabotage of the network, it is aimed at integrity.

The WEP protocol is used to encrypt communications between wireless clients and access points. This means that data is transmitted in plaintext until it is travelling within the wired network and that it is encrypted at the time of emission into space in the form of radio frequency or microwave electromagnetic waves. WEP is based on the cipher stream named RC4. This cipher is applied both to the body of each frame and to the CRC. There are two operational levels of WEP:

1. The first method is based on a 40-bit encryption key and a 24-bit initialisation vector, which leads to a total of 64 bits.
2. The second method is based on a 104-bit encryption key and a 24-bit initialisation vector, which leads to a total of 128 bits.

This protocol has been characterised by real problems, from the moment of its creation, such as the scarcity of design elements and key management that have made it an extremely deficient protocol from the viewpoint of security. These problems will be analysed below in detail and this section illustrates only the operational modes of WEP that allow for a better understanding of the security vulnerabilities of the same.

It has already been stated that WEP uses the stream cipher called RC4 to encrypt the data transmitted. This encryption process is divided into different phases. The first phase is to generate a seed value that is used to commence the process on the WEP key. After the value has been defined, it must be sent to the access point to allow decryption of the data transmitted. This value is composed of a 26-digit hexadecimal number.

This value is not used alone to generate a stream of encrypted data using WEP, but a technology to make the key random is also used. This technology uses an initialisation vector (IV) that is generated frame by frame. The IV generation technology varies from manufacturer to manufacturer. The standardised technology in 802.11b standardises the length of IV and provides that the same varies from frame to frame. In addition, recommendations concerning the increase or randomness of the IV sequence are not provided; this lack of information causes significant security problems for WEP. Once IV and the WEP key are combined, the frames can be encrypted. Using the RC4 stream cipher, the key and IV undergo a process of XOR together with the data to generate an encrypted frame. Subsequently, a copy of the IV is inserted in plaintext in the header of the frame itself. At this point, the frame is transmitted.

Once the other party receives the frame, it extracts the IV value and applies the default seed value to generate the same WEP session key that was used to encrypt the packet. The same RC4 process is applied in reverse to obtain plaintext from the encrypted text. Once this operation is completed, the CRC is taken and compared with the data to be certain that the same is still intact after the transmission process. The encryption process is shown in Figure 6.37.

As can be seen in Figure 6.37, data and the integrity checking are combined, and the same takes place for the WEP and IV key. The latter are inserted in a pseudo-random number generator. The output generates a stream of keys whose length is equal to the length of the frame payload plus the CRC. An XOR function is then applied on the result obtained during the previous phase, by generating a cipher stream. The next phase consists of the insertion of IV within the header of the packet and transmission of the same.

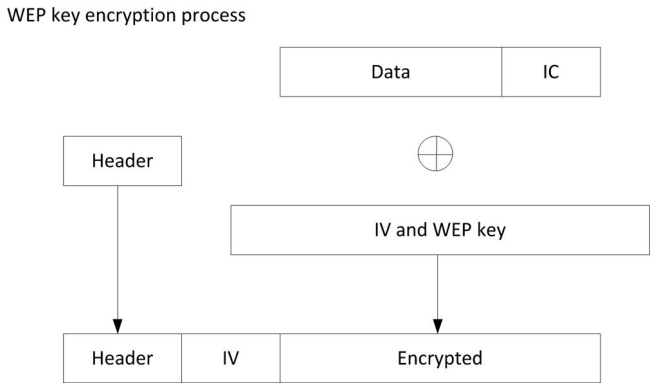


Figure 6.37 Encryption process using WEP.

6.11.6 802.1x

The 802.1x standard was approved by the IEEE in June 2001 and the American National Standards Institute (ANSI) in October 2001.

The 802.1x standard was developed for the authentication of all IEEE 802 networks. This means that this standard can operate over Ethernet, Token Ring, wireless, and on all other systems and devices that use it.

A certain amount of confusion has often been generated about the fact that encryption, in 802.1x, takes place outside of the same. It must be remembered that this standard is intended for authentication on ports: this means that the standard takes the authentication request, assesses whether the same is allowed or not allowed on the network and subsequently ensures or revokes access.

Many descriptions of how 802.1x operates are contained within other standards such as Extensible Authentication Protocol (EAP) and Remote Authentication Dial-In User Service (RADIUS, described in section 6.11.7). The 802.1x standard provides a mechanism that denies all access traffic to the network with the exception of EAP; once the latter approves request for access of the device to the network, the 802.1x protocol tells the switch or the access point to allow the traffic generated by the same device.

The schema of 802.1x is shown in Figure 6.38.

As can be seen in Figure 6.38, the 802.1x protocol provides for two other standards, EAP, with regard to the relationship between the applicant and the authenticator, and RADIUS, with regard to the relationship between the authenticator and authentication server. The 802.1x protocol receives requests from EAP, sends them to the RADIUS server and waits for a response. From the figure, it is clear that the authentication server, the authenticator and the applicant represent three main roles of 802.1x. Each of them performs a specific task in the process of authentication exchange and allows those devices that are successfully authenticated to access the network.

6.11.6.1 The authentication server

The authentication server is responsible for permitting or denying access to the network. It receives requests for access from the authenticator. When the authentication server receives a request, it validates it and returns a positive or negative message to the authenticator. This represents the final part of 802.1x authentication; the operations of this server are defined in the RADIUS standard.

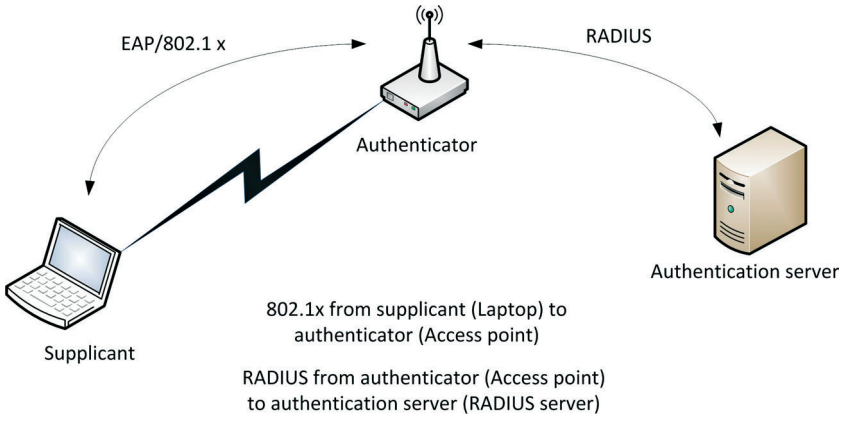


Figure 6.38 Schema of 802.1x.

6.11.6.2 The authenticator

The authenticator is the first element to which a device based on 8021.x authentication connects. It can be an access point or any other device that allows access to the network. The purpose of the device is to allow the passage of only EAP packets and to wait for a response from the authentication server. Once the authentication server responds with a message (positive or negative), the authenticator behaves accordingly. If the received message is negative, it continues to block the traffic until the same receives a positive message. If the received message is positive, then the authenticator allows the applicant to access the network.

6.11.6.3 The applicant

The applicant is the device that wishes to connect to the 802.1x network. It can be a laptop computer, a PDA, a smart phone or any other device capable of using wireless technology. When the applicant wishes to connect to a network, the same must pass through the authenticator. The authenticator allows the requester for only passage of the direct EAP traffic to the authentication server. This EAP traffic is the credentials of the user or of the device. Once the authentication server has evaluated these credentials, it returns the message of authorisation to access or denial of access depending on the same credentials.

6.11.6.4 EAPOL

The Extensive Authentication Protocol over Local Area Network (EAPOL) is a part of EAP even if the same is described in 802.1x. For this reason, it is described together with the latter protocol. This is due to the fact that the 802.1x standard allows the passage of certain types of EAP messages from the authenticator up to the applicant. Each message, to be authorised for transit, must comply with the 802.1x standards in terms of the type of message and format. The EAPOL standard refers to the frame structure used to send traffic from the authenticator to the applicant. This traffic is defined by a limited number of types of frames that represent the only frames that are allowed to transit from the access point to the client. IEEE has provided space for additional types of frame, although the current standard requires a limited number. These frames are:

1. EAPOL-packet, which is used to identify the package as an EAP packet;
2. EAPOL-start, which is used to initiate an EAP conversation or an 802.1x authentication;
3. EAPOL-logoff, which is used to terminate an EAP conversation or an 802.1x authentication;

4. EAPOL-key, which is used to exchange information between the authenticator and the applicant. This represents a security frame;
5. EAPOL-encapsulated-ASF-alert, which represents an EAPOL frame that is used to carry Simple Network Management Protocol (SNMP) information outside an authenticated port that is not of the 8021.x type.

The most critical frame from a security perspective is EAPOL-Key frame, which is used to send data relating to keys, such as dynamic WEP keys. The only key defined in 802.1x is the RC4 WEP key.

6.11.7 RADIUS

RADIUS is a protocol used in network environments for authentication, authorisation and accounting (AAA). RADIUS can operate on different types of devices such as routers, switches, servers, modems, Virtual Private Network (VPN) concentrators and other types of devices. The protocol works by generating an encrypted tunnel between the network device and the RADIUS server. This tunnel is used for all the information for AAA about the identity of the user, the areas of access permitted and the current location of the user. To begin an encrypted tunnel, a password or passphrase is needed, which is known as a shared secret. This shared secret is located in the RADIUS server and on the RADIUS of the device that participates in the process. Once the shared secret has been set correctly, secure communication can occur.

One of the advantages of RADIUS is the use of a user-shared archive that provides the AAA information for all types of devices. The RADIUS archive used to store information relating to username and password can be programmed to refer to the different types of directories.

This protocol allows administrators to centrally control user access to all devices in the network. RADIUS, in addition, also deals with the removal of the devices connected to the network when users leave without using the procedures of disconnection.

Once an organisation has developed RADIUS, the profiles of users' access can easily be removed when they terminate their working relationship with the organisation, unlike what must be done in the absence of RADIUS, which consists of the manual transmission of all the usernames and passwords to all the network devices, a very time consuming and expensive operation and not without errors because the same must be performed manually.

With regard to the use of RADIUS in wireless networks, it can be used as a method of access to manage the access points, to allow the administrator to perform the same operations that are performed on routers and switches. In addition, access points and the RADIUS server share secrets that are used to create the encrypted channels used to exchange authentication information.

RADIUS, moreover, as is provided for in 8021.x authentication, can be used as a mechanism for authenticating users, as this represents one of its operational characteristics. In this sense, the access points must share a secret correctly with the RADIUS server to allow the access points themselves to trace users' requests for access to the network; this means that users must negotiate their authentication only with access points and not with the RADIUS server.

RADIUS provides only four types of packets for authentication and other types of packets for accounting. These are:

1. Access-Request, which allows commencement of the RADIUS sequence;
2. Access-Accept, which informs the RADIUS client that the authentication supplied to the same is correct;
3. Access-Reject, which informs the RADIUS client that the authentication supplied is not correct;
4. Access-Challenge, which is used to challenge a RADIUS client on their authentication credentials.

The structure of a RADIUS frame is shown in Figure 6.39.

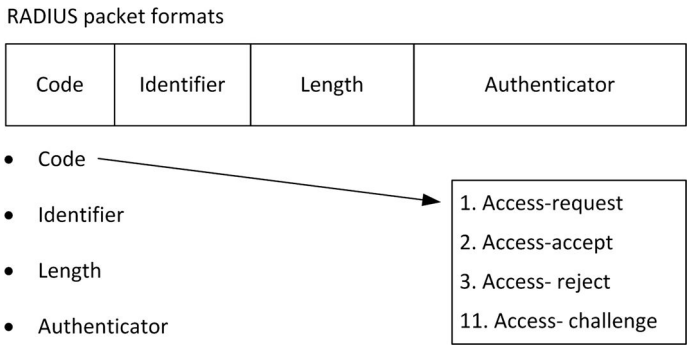


Figure 6.39 RADIUS frame structure.

As can be seen in Figure 6.39, all four types of packets have the same format. They are identified by a field code. This field can contain different numbers, even if it is limited to the following:

1. access request: 1;
2. access accepted: 2;
3. access rejected: 3;
4. access challenge: 4.

The next field in the frame is the identifier that is used for agreement between their various requests and various responses, thus avoiding confusion when the RADIUS server must fulfil multiple requests simultaneously.

The next field is the length field that is used to identify the length of the packet as the RADIUS packets can have, within the same, up to 2,000 attributes, and a mechanism to measure the length of the packets is therefore necessary, avoiding errors of various kinds.

The next field is the authentication field, where there is a password, properly protected by a hashing mechanism.

Because the RADIUS is a server, it is absolutely essential to implement all of the security systems within this, already seen previously, to protect the same, given its importance within the network; compromise by an external attacker compromises the security of the entire network.

6.11.8 EAP

EAP is a method of authentication to access a network.

Initially, the authentication protocol was used through a password or Password Authentication Protocol (PAP), which in a short space of time showed its limitations in terms of security. After this, the Challenge Handshake Authentication Protocol (CHAP) was developed that also presented its security limitations in a short space of time. At this point, the industrial world decided to develop a protocol that worked in the same way regardless of the type of authentication used (e.g. password, biometrics and tokens), without the need to write additional code; the result was the creation of EAP, which is able to use different authentication mechanisms without having to change its operating mechanisms.

When the EAP was created, the same was asked for compatibility with the (Point-to-Point Protocol (PPP). To ensure this compatibility, the support of PPP packets was included in EAP, allowing every device that used PPP to support EAP.

The EAP frame is composed of four fields:

1. code;
2. identifier;

3. length;
4. data.

The structure of this frame is shown in Figure 6.40.

The fields of the EAP frames are used for the proper functioning of the protocol. They are shown in detail in the following:

1. Code field: this is composed of 8 bytes and allows four types of codes. The request code (identified with number 1) that makes the initial request; the response code (identified with number 2) that informs the applicant that the request has been submitted; the success code that notifies outcome of the authentication (3 for positive outcome and 4 for negative outcome). To ensure a high level of security, frames must be exchanged by following the appropriate sequence. For example, if the successful frame is received from the client before the response frames, the client realises that the sequence has not been respected and deletes that frame.
2. Identifier field: this field is 16 bytes long and is used to associate the requests to the responses and is essential when there are multiple clients at the same time that send their requests. This field is uniquely associated with each client;
3. Field length: this field is 8 bytes long and is used to indicate the length of each EAP packet. This length includes the code, identifier, length, type and length of the data part.
4. Field type: this field is 8 bytes long and serves to identify the structure of each EAP packet of request and response. This EAP field allows implementation of different types of authentication and can contain four main types of codes to check the transmission of EAP credentials with the authentication server. These codes are identification, notification, Null AcKnowledge (NAK) and Message Digest algorithm 5 (MD5) challenge. The identification code is used to request the identity of a client and to begin the process of identifying the type of method intended for use for authentication. The notification code is used to request that the client perform a given action before proceeding. The NAK code is only used in response messages and serves to communicate to the client that the authentication type is invalid. There is also the option of sending an extended NAK that provides further details on the EAP type that is accepted by the authenticator.
5. Data field: this field contains data representing the EAP request and representing the authentication part. For example, if the EAP uses MD5, then the user must provide the username and password in this field to have access to the point.

It has already been said that one of the major benefits of EAP is the ability to support different types of authentication. This has allowed EAP to be constantly used, avoiding the typical fall into disuse of protocols when vulnerabilities and defects are discovered.

Currently, there are different versions of EAP, some of which are defined in the relevant standard while others are typical of producers.

The main versions of EAP are described below.

6.11.8.1 EAP-MD5

EAP-MD5 is one of the main types of EAP. This type uses the MD5 hashing algorithm to validate the credentials of users. Other methods of EAP create an encrypted tunnel within which is implemented

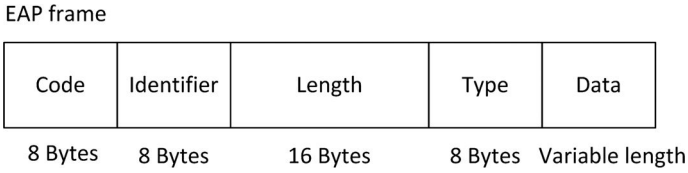


Figure 6.40 Structure of an EAP frame.

the EAP-MD5 validation process. One of the requirements on which EAP-MD5 is based is the sharing of a secret. It is clear that this secret must be shared through a separate communication channel. This secret is then used to encrypt a challenge and to check that the other party is in possession of the same secret.

A schematic diagram illustrating the operation of EAP-MD5 is shown in Figure 6.41.

As can be seen in Figure 6.41, the first step consists of sending, by the client, a login request packet to the access point. Before this, the secret must have been previously shared using another means of communication. Once the access point has received the request, it responds by asking the client to provide data relating to their identity. This identity determines the type of EAP that the two parties should use. The identification field, contrary to what its name would suggest, should not be used to contain the identity of the user, but only for the EAP identity type. As the client requires an MD5 system, the identity type within the EAP frame should contain this information. Once the access point receives the required information, it will forward the information to the authentication server which sends a challenge to the client, represented by a plaintext string. Once the client receives the challenge, it encrypts it using the shared secret key and sends the encrypted result to the authentication server. The authentication server checks the received result and, in the case of a positive outcome, an EAP message is sent of success of the operation. In the event of problems, a message of failure of the request is instead sent along with an explanatory code of this failure that serves to explain to the client the reason for refusal of access.

6.11.8.2 EAP-TLS

EAP-TLS is the acronym for *Extensible Authentication Protocol-Transport Layer Security* created by Microsoft in 1999. TLS was created by the SSL protocol, used to secure navigation on the Internet. The EAP method uses certificates to authenticate users and requests certificates both from the server and from the client. This represents the part of the TLS within EAP that is a good method for the use of certificates. This EAP method represents one of the strongest and most secure methods, even if the same is not able to prevent a possible attacker from entering the network via a computer whose certificate has already been installed.

The EAP-TLS 802.1x support requires specific work. First, the intervention of a Certification Authority (CA) is needed, required for the distribution of the certificates of the client and the server. An AAA-type server is also necessary that supports EAP-TLS. Finally, a client must be able to support this type of EAP. Once all these elements are in place, the same must be configured correctly to make them work properly in synergy.

The EAP-TLS method is similar to the EAP methods. For help in understanding, its operating schema is shown in Figure 6.42.

This begins with a request packet that leaves the client, passes through the authenticator and reaches the authentication server. Once this response has been received, the authentication server sends

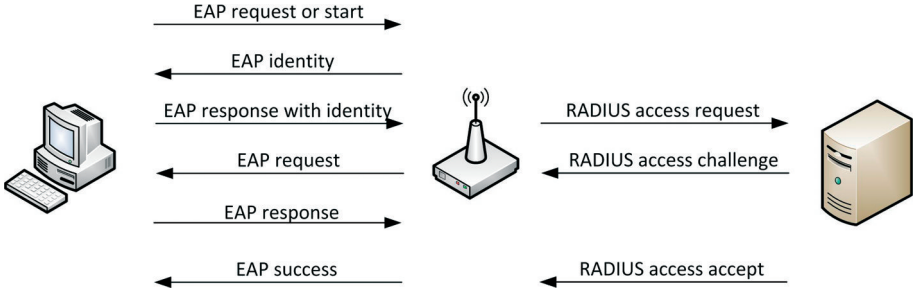


Figure 6.41 Operating schema of EAP-MD5.

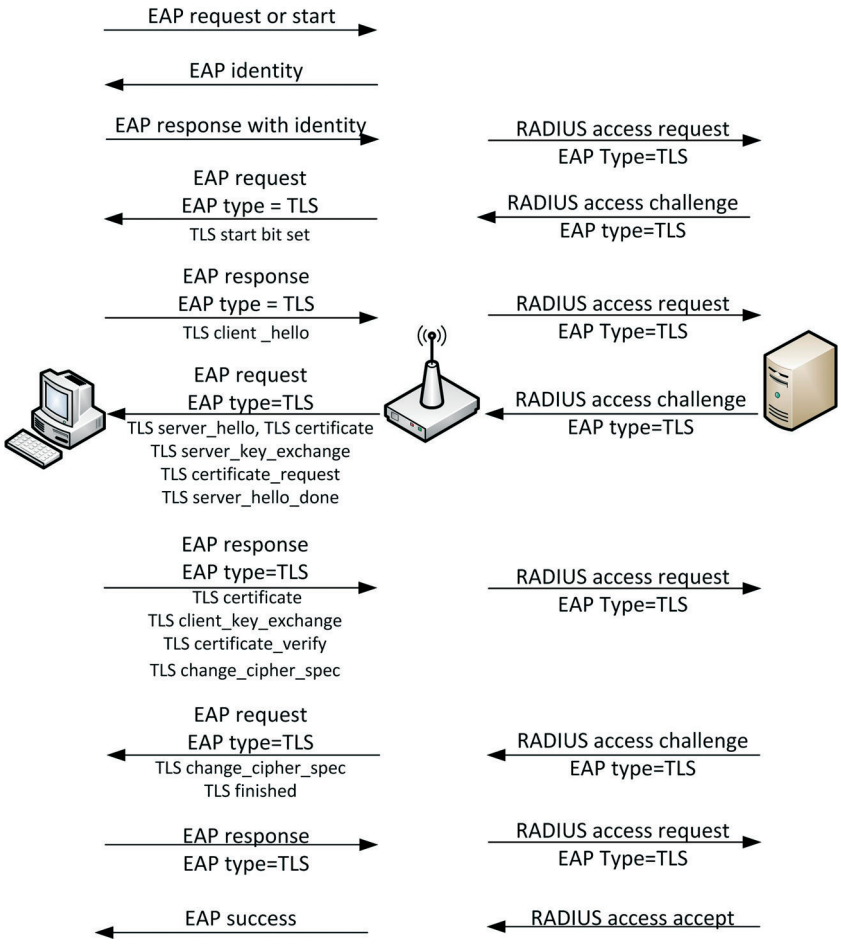


Figure 6.42 Operating schema of EAP-TLS.

back a response to the client to ask for their identity. Subsequently, the client provides its own identity through EAP-TLS. Once such information has been received from the authentication server, it sends the certificate of its public key to the client. Once the client receives the certificate, it responds with its own public key. Once this information has been exchanged, a secure channel can be created. This operating mechanism is very similar to the one used for navigating on the Web, when the same must be done using secure pages even if, in this case, authentication of the client also takes place.

Once both parties are authenticated, there is another EAP process within the EAP-TLS tunnel to allow secure authentication. For example, exchanging EAP-MD5 can take place within the encrypted tunnel allowing weak authentication to become more secure due to the fact that it is occurring within the same tunnel.

6.11.8.3 EAP-TTLS

EAP-TTLS is the acronym for *Extensible Authentication Protocol-Tunnel Transport Layer Security*. It was created to provide support to the older devices that were not able to operate according to the new types of authentication. This method represents an EAP technology that allows secure communication of credentials along with the possibility of using an authentication type provided in the standards.

A schematic diagram illustrating the operation of EAP-TTLS is shown in Figure 6.43.

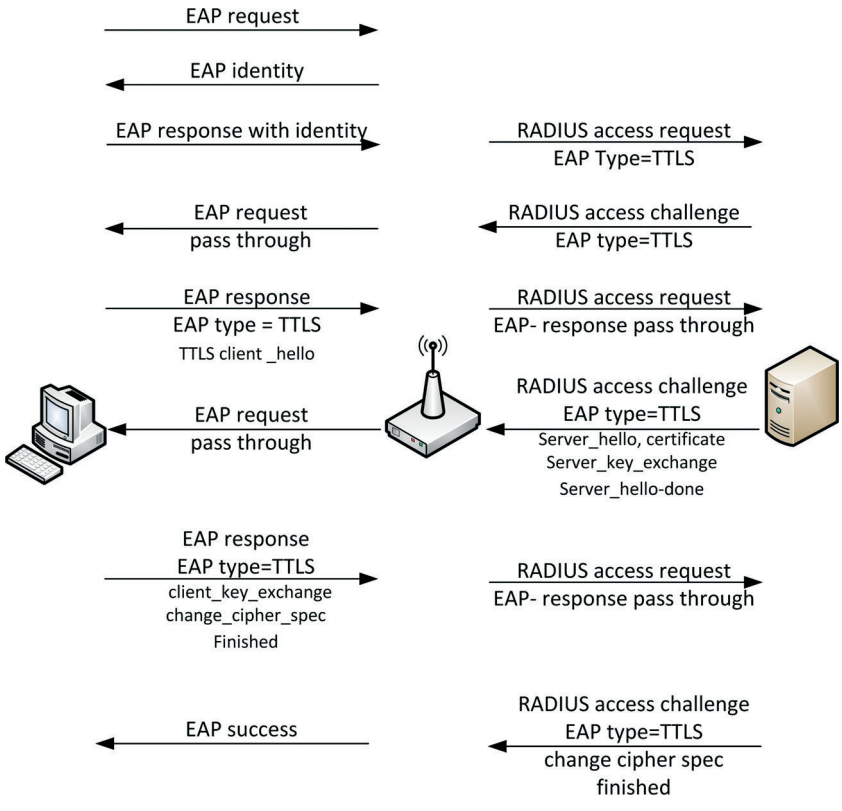


Figure 6.43 Operating schema of EAP-TTLS.

This operates by performing authentication within a secure tunnel and, unlike EAP-TLS, it is able to use different types of authentication for the part relating to the client, and also older parts such as PAP, CHAP, MS-CHAP, MS-CHAP-V2 and others.

This technology works with the usual request that the client sends to the authentication server passing through the authenticator. Once such a request has been received, the authentication server sends back a request for submission of the identity of the client. The client provides their identity requesting the use of the protocol in question. Once such information has been received, the authentication server responds by sending the certificate of its public key to the client. In this case the client does not send its certificate as the EAP-TTLS must be able to guarantee operation of more dated devices that are not able to use technologies of advanced authentication such as those that require the support of a PKI. Once the client receives the certificate from the authentication server, it uses EAP-TTLS to enable a secure channel between it and the authentication server through the authenticator. This action is performed using the TLS and the public key of the authentication server. In this sense, the same operations take place that are performed when a client accesses a Web page protected by TLS or HTTPS. Once this operation has been completed, other types of authentication within the TLS tunnel can be used. This is the difference between EAP-TLS and EAP-TTLS because at this point other technologies, such as MD5, can be used inside an EAP tunnel that already exists. Because of the EAP-TLS, other forms of authentication can be used that would not be possible on the devices that do not support EAP.

6.11.8.4 LEAP

LEAP is the acronym for Lightweight Extensible Authentication Protocol. It works in a different way compared to the other main types of EAP. The process begins with the client request sent to the access point. When the access point receives such a request, it sends a response, asking it to send its own identity. Most EAP methods request from the client the EAP method that it intends to use. LEAP, on the contrary, responds by sending a username and password in the identity field. Once the access point receives this information, it sends it to the authentication server or RADIUS. The server, at this point, sends a challenge to the client via the access point.

The challenge consists of a message of 8 bytes. When the client receives this message, it performs a series of steps to respond to this message. First, it performs an MD4-type hashing that produces a 16-bit password hash. This hash is characterised by having 5 bytes with value equal to 0, by generating a hash of total length equal to 21 bytes. This hash is divided into blocks of 7 bytes that are sent to a Data Encryption Standard (DES) encryption process. Each process uses 7 bytes as the key and the original 8 bytes as plaintext, generating a ciphertext that is 24 bytes long. This text is sent to the authentication server that performs the same operation in reverse, obtaining the original password. Once in possession of the password, it compares it with that of the user that already has it. If the comparison is successful, the authentication server sends a message of acceptance of access to the access point. Contrarily, if the comparison fails, then the authentication server sends a message of denial of access to the access point. Once the access point receives such a message, it sends, as a result, a message to the client.

6.11.8.5 PEAP

PEAP is an acronym for *Protected Extensible Authentication Protocol*. One of its major advantages is the use of a type of very strong EAP that does not require a client certificate, such as EAP-TLS. It operates in a manner similar to EAP-TLS, generating an encrypted tunnel with TLS, which is subsequently used for another EAP method. Unlike EAP-TLS, when PEAP performs this process, it does not validate the client certificate.

A schematic diagram illustrating the operation of PEAP is shown in Figure 6.44.

The process begins with a request packet that is sent to the access point. Once the access point receives the packet, it responds with a response packet that requests the identity of the client. The identity is in EAP format and, in this case, the type field contains the PEAP information. This information alerts the authentication server about the method that is used. The traffic passes through the access point that operates as an authenticator. Once the EAP method has been identified, a TLS tunnel is established using the certificate of the authentication server. Once the TLS tunnel has been generated, a new EAP process to authenticate the client takes place within the same.

6.11.8.6 EAP-FAST

EAP-FAST is the acronym for *Extensible Authentication Protocol-Flexible Authentication via Secure Tunnelling*. This EAP method is particularly fast compared to other EAP methods, and this feature is particularly useful when Wi-Fi phone or applications that are sensitive to time delays are involved.

A schematic diagram illustrating the operation of EAP-FAST is shown in Figure 6.45.

EAP-FAST operates using protected access credential (PAC) that appears to be very similar to a certificate. PAC uses a key that is used to provide an encrypted tunnel between the client and the authentication server. This key is dynamically allocated. After a new PAC connection is established, it can be sent through the encrypted tunnel. To avoid any problem of key management, PAC is characterised by the ability to be periodically changed once a secure connection has been established.

PAC is composed of three parts:

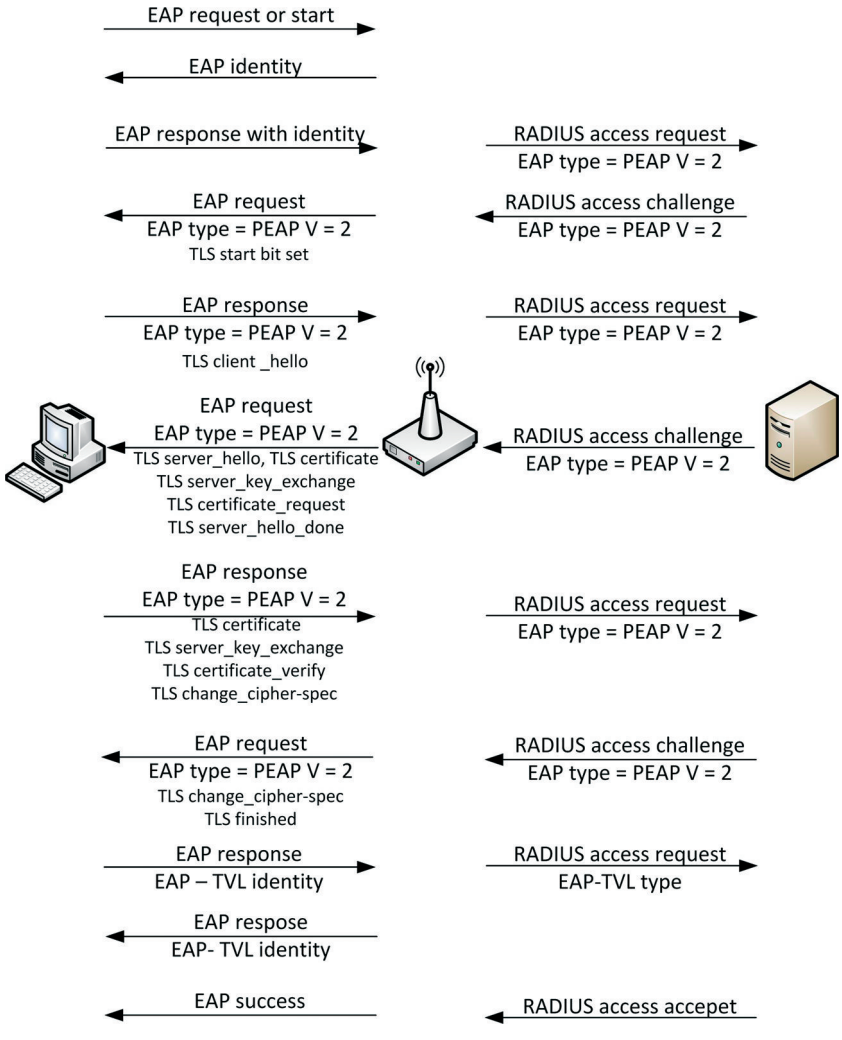


Figure 6.44 Operating schema of PEAP.

1. master key;
2. a component called opaque;
3. PAC information.

These parts are characterised by specific functionality to support EAP-FAST. The master key is randomly generated by the authentication server and this generation is performed to produce a strong entropy. This key is strongly linked to the identification information. After the key has been generated, it is sent to the client. The second part, the opaque component, contains the identification information (ID authority) and the life span of the key encrypted with the key master. The final part is the section of information that contains the identification information and may also contain the time duration of the key. This field is used to support multiple connections and, therefore, sets of multiple credentials.

Once PAC has been generated, the EAP process can begin. It begins, as usual, by means of a request by the client. The access point responds to the request via an identity request. The client sends its identity, which is not the identity of the client, but the identity of the EAP type. In this case, this

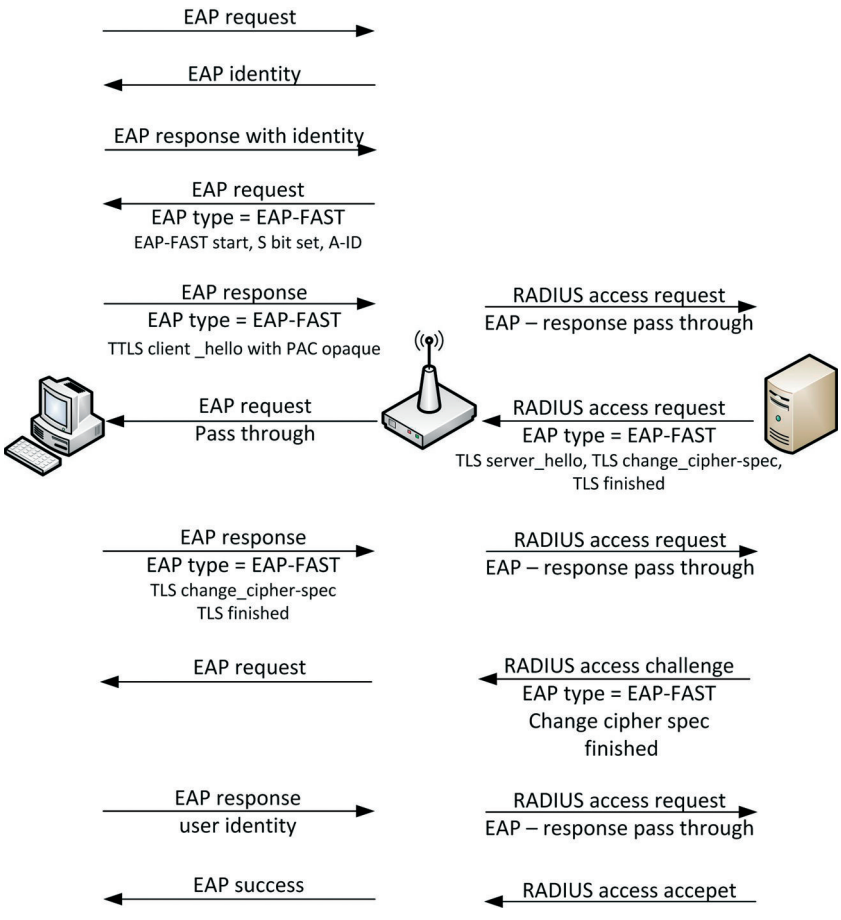


Figure 6.45 Operating schema of EAP-FAST.

identity contains the EAP-FAST type. Once the access point receives this information, it responds with an EAP request to send the PAC of the client. This information is sent to the authentication server. The authentication server sends an authority ID that represents the field contained within the PAC. This field is used to identify the correct PAC, which is taken from a list of PAC. Once the client receives the authority ID and compares it with the PAC, it generates a TLS tunnel based on the master key of this PAC. This tunnel is used to continue with the additional authentication information.

6.11.9 WPA

WPA is the acronym for *Wi-Fi Protected Access*. It originated after an event where WEP had been breached and the industrial world turned again to the IEEE, which announced that it would develop the 802.11i standard. Because realisation of this standard was delayed, the sale of wireless devices began to decrease due to the lack of secure reference standards. This situation prompted manufacturers to request the IEEE and other standardisation bodies to prepare something that could ensure a certain level of security to wireless products. Due to the delay in development of the 802.11i standard, the Wi-Fi Alliance created a subset of 802.11i called WPA. It uses the material already developed by the 802.11i group and formalised it in WPA; this means that the significant changes to the 802.11i

standard had a decisive influence on the future version of WPA, as in reality was the case with WPA2. Currently, having available the full version of 802.11i, the use of WPA has significantly decreased.

The WPA standard is capable of supporting two methods of authentication and key management. The first is an EAP authentication using 802.1x, using this protocol and an authentication server. This method appears to be the most secure of the two and requires a smaller number of management activities by the client. The second method is by using pre-shared keys. This method requires that a key is used by the client and by the access point; this means that anyone in possession of this key can access the network. To prevent a possible attacker in possession of the key from intercepting communications, WPA uses a technology that creates a single session key for each device. This is done using a pre-shared key called group master key (GMK) that is used to generate a pair transient key (PTK). This second method has been added to WPA to provide a simple support for use in homes or in small offices, where it is not always possible to use an authentication server such as RADIUS. A PSK consists of a number of 256 bits or a variable passphrase ranging from 8 to 63 bytes.

WPA is able to support the older Temporal Key Integrity Protocol (TKIP) and message integrity control (MIC). It also uses AES with the difference being that the method is slightly varied with respect to the one defined in 802.11i. WPA is able to negotiate a cipher or an authentication method characterised by those that are defined robust security network information element (RSN IE).

One of the reasons why 802.11i was not published is that some parts of the same had not yet been defined. Given that only the complete parts that were being produced by the 802.11i group were being used, it was inevitably necessary to develop the remaining parts needed and, for this reason, there are differences between the two standards.

The first big difference is that WPA supports TKIP as the default setting, unlike 802.11i that supports AES CCMP by default.

The second difference is that the WPA does not support AES CCMP but supports AES.

The third difference between WPA and 802.11i is RSN IE. This latter element is used to exchange the information related to the ciphers supported between the access point and the client. In 802.11i, this part was not well defined, and in WPA, it was necessary to produce new rules that did not conflict with those already defined by the 802.11i group concerning RSN IE. This led to the creation of a WPA IE that used different values to distinguish them from one another and allowed, once RSN IE was well defined, insertion of the same within WPA.

6.11.10 802.11i

The 802.11i standard was created to enhance the security of networks based on 802.11. In this standard, IEEE defined a secure technology to have access to wireless networks. This standard is intended to reduce the enormous amount of threats that had rendered the wireless networks a serious risk to the security of organisations.

The standardisation process of 802.11i began in 1999 when a strong interest was taken in increasing the security of the 802.11 f MAC layer from the viewpoint of quality of service and privacy. In this sense, in 2000, a group called TGe was founded. The following year, it became clear that the work that this group was to carry forward would be too expensive, and the same was divided into two groups: one relating to the quality of service and one relating to security. This division led to the creation of a group called TGi that gave rise to the 802.11i standard that was finally approved in 2004.

The 802.11i standard uses a certain number of standards, protocols and ciphers, which had already been defined outside of the same standard even if a certain number of standards such as RADIUS, 802.1X, EAP, AES, RSN is TKIP are defined within it.

RSN (robust security standard) is also used for the dynamic negotiation of authentication and encryption. It is used to negotiate the type of encryption that is supported by a client as well as the type of encryption.

Another part of the 802.11i standard describes how to use the EAP. In fact, it was established that the 802.11i standard does not specify a specific method of authentication, but rather a protocol able to support different authentication technologies that represents what is normally EAP as the latter provides for the use of a password, smart card, certificates and other technologies, based on the same procedures of request, acceptance and rejection. To allow EAP to operate correctly within 802.11i, use must be made of 802.1x to facilitate the communication of EAP between trusted and not trusted subjects. The main purpose of 802.1x is to provide support for authentication and the management of strong keys. The 802.1x protocol allows access points to allow only EAP requests to the network. This takes place until the client is authenticated. Once authenticated, it is possible to continue with the procedures of key negotiation and to access the network.

In a way similar to WPA, 802.11i also includes a special option for those environments in which use is not possible, especially for economic reasons, of an authentication server. This authentication server represents a basic requirement of 802.1x. In this manner, to make possible the use of 802.11i both in large organisations and in small offices and homes, it was necessary to resort to another method, the pre-shared key. This method is very similar to the method of pre-shared WPA keys. While using a pre-shared key, each client uses a secret for generating all of the material concerning the same keys. The master key is the same throughout the network, as well as in WEP, although the same is used to generate a session key for each client.

Shown below are all the components of 802.11i that have not been previously illustrated.

6.11.10.1 Robust security network

Robust security network (RSN) was created as a part of the 802.11i standard and this is why it is described in this section. RSN specifies user authentication via 802.1x to the encryption using TKIP (which is discussed in section 6.11.10.2) or through the counter mode of the cipher block chaining-message authentication mode (CBC-MAC), CCMP protocol. RSN is capable of supporting TSN to allow more dated devices to use WEP. RSN uses TKIP and AES as the encryption techniques to protect data confidentiality; TKIP is used to ensure compatibility with older devices, while AES as a long-term encryption technology. AES can be used in many ways and in particular, in the method named CCMP. The RSN protocol uses EAPOL-key messages for key management. RSN works in conjunction with 802.11i in selecting an authentication method and a cipher.

RSN generates the appropriate messages to alert the client and access point of the cipher that they must use and to allow for negotiation. These messages are contained within what is called RSN or RSN IE. An RSN IE is used to communicate to another device the types of supported ciphers. The RSN IE may be sent to the inside of a beacon from an access point or as a response to a client request. After an association request, a response containing an RSN IE is returned that lists which method corresponds with the method supported by the other party.

The standard provides that RSN IE can be contained within beacons; in association request, in re-association request or in probe response.

The structure of an RSN IE frame is shown in Figure 6.46.

As can be seen in Figure 6.46, the RSN IE frame is composed of 11 parts of which, only the first three are required in RSN IE transmissions. After the first three, all the other fields must contain data,

Element ID	Length	Version	Group cipher suite	Pairwise cipher suite	Pairwise cipher suite list	AKM suite count	AKM suite list	RSN capa-bilities	PMKID count	PMKID list
1 octets	1 octets	2 octets	4 octets	2 octets	4-N octets	2 octets	4-N octets	2 octets	2 octets	16 octets

Figure 6.46 RSN IE frame structure.

otherwise they are ignored: this means that in order to be considered, a determined field must have all the previous fields filled with data.

The elements that constitute the RSN IE frame are:

1. ID element: this field is composed of 48 decimals or 30 hexadecimals.
2. Length: this server field identifies the total length of the RSN IE frame. The frame is 255 octets long, and for this reason when there is a high number of ciphers, there may be a situation of only being able to use a limited number due to the limited length of the frame.
3. Version: this field is used to indicate which version of RSN IE is being used. It may contain two octets.
4. Cipher group: this field is used to indicate the ciphers used to protect the broadcast or multi-cast traffic. It can contain up to 2 bytes of information concerning the ciphers supported by multi-cast and broadcast.
5. Number of cipher pairs: this field indicates the number of cipher pairs being used. This field is two octets long.
6. List of cipher pairs: this field indicates all the ciphers that have been selected for the key pair. Each of these ciphers is counted in the previous field. Each cipher is characterised by a corresponding type within the field. Each and every one of them is at most four octets long. Three of the octets contain the so-called organisationally unique identifier, and the remaining octet contains the identity of the cipher.
7. AKM number: AKM is an acronym for authentication and key management that represents a cipher that is used to determine how many different options for the management of keys are available, such as pre-shared keys or those dynamically allocated with 802.1xx. It should be emphasised that in an IBSS, only one AKM can exist. This field can be at most two octets long.
8. List of AKM: this field is used to specify the AKM available. Depending on the content of this field, it can contain multiple sections of four octets, each containing a key management option. In the current state, there are only two versions: pre-shared keys and 802.1x key management.
9. RSN capacity: this field, two octets long, is used to identify which capabilities of RSN are available on the network. It allows the device that receives the RSN IE to understand if it is able or not to support RSN all together; if it is not able to support it, it is always possible to try with transition secure network (TSN). If it is not included, it is assumed that it is not possible to use RSN.
10. Number of PMKID: PMKID is an acronym for *pairwise master key identifier* and the number of PMKID two octets long and is used for re-association. It is used to store keys in such a way that, when a client moves from the coverage of one access point to another, the client need not perform a new authentication process, allowing for maximum operating speed and a reduction of the bandwidth used. This field is used to define the number of credentials that are located within an RSN IE frame.
11. PMKID list: this field is composed of four octets per each of the PMKID identified in the previous field and contains the different PMKID. There are currently three main ones.

RSN, therefore, represents a method to negotiate the type of security that every client and access point can support. Such technologies are identified in terms of ciphers within an RSN IE frame. These ciphers allow the use of, or non-use of, a combination of security technologies. This means that an appropriate security policy can be implemented that denies the use of weak technologies, such as WEP, and encourages the use of much stronger technologies, such as TKIP or AES. This possibility allows the designer of the network to use the desired security policy.

TSN is a part of 802.11i. It is used to ensure compatibility with the more dated devices. It has already been said that RSN allows the use of different technologies of authentication and encryption on the same access point. To be sure that some of the most vulnerable technologies have not been used, TSN has taken into consideration. This makes RSN more secure and allows the deletion of security

flaws of the more dated devices. In RSN, if it is decided to support TSN, WEP is not considered a valid negotiation option between the access point and the wireless client.

6.11.10.2 Temporal Key Integrity Protocol

TKIP is an internal solution developed to address the problem of the re-use of keys by WEP. It has thus become a part of 802.11i and therefore of WAP. This means that different types of TKIP existed until 802.11 was standardised.

TKIP was included in 802.11i to ensure compatibility with more dated devices. 802.11I did not intend to use the RC4 cipher, and for this reason it was decided to use AES. The ability to support TKIP was provided to ensure compatibility with more dated devices. To do this, 802.11i needed to support a protocol that could easily improve WEP into something that was more secure. One of the main reasons for using TKIP rather than WEP is its greater security and not having experienced the high number of attacks that WEP had suffered. In this way, the level of network security is increased and the overall risk of the same is decreased.

The TKIP standard was also supported by the industrial world because the same allowed easy migration from WEP, requiring only a simple firmware upgrade, that was easy to implement and with lower costs compared to replacement of the hardware.

The TKIP encryption component operates in two phases. The first stage generates a session key from a temporal key, the TKIP sequence counter (TSC) and the MAC address of the sender. The temporal key is characterised by a 128-bit value similar to the basic value of the WEP key. TSC is composed of a source address (SA), a destination address, a priority and the actual data. Once this phase is terminated, a parameter named TKIP-mixed transmit address and key (TTAK) is generated. This parameter is used as a WEP session key for use in the second phase.

In the second phase, TTAK and IV are used to generate a key for data encryption. This process is very similar to that of WEP. In WEP, the first 24 bits and IV are added to generate a single key that is used as the encryption data key. Subsequently IV is inserted in the header of the packet. TKIP provides for an extension of IV space, generating a large space that provides for additional 24 bits. In the second phase, the first 24 bits are filled with additional 24 bits of the TTAK while the second 24 bits are filled with the unused part of TSC. This represents a choice characterised by greater security because the key uses different values according to the device being conversed with. In WEP, every client and access point generate the same random value. Some manufacturers use a random value that is incremented by one, making this technology extremely vulnerable to attack.

The basic operating principles of TKIP derive from WEP. In 802.11i, TKIP is considered an encryption system that increases the security of WEP.

Another element used by TKIP is MIC. It provided for different standards before the 802.11i defined it in a single standard. This protocol was created to counter the various attacks aimed at changing messages that represent the majority of attacks against WEP.

MIC is a hash that is calculated on the basis of the single packet. This means that a single MIC hash can be applied to multiple frames, allowing fragmentation. MIC also operates on the basis of a single transmitter – single receiver. This means that in every exchange of information, a different MIC is used for data flows that occur in two opposite directions.

MIC is based on a seed value, on a MAC destination, on a MAC source, on priority and on the payload of data. The seed value is very similar to the IV value of WEP. TKIP and MIC use the same IV space, adding four further octets to the same. This is done to reduce the risk of using the same IV twice in a relatively short space of time.

MIC is also encrypted in the data part, and this means that the value cannot be obtained through actions of *sniffing* on network traffic. To avoid the attacks that change messages, TKIP and MIC introduced appropriate countermeasures. These countermeasures consist of the suppressing of

communications of an access point if there are two problems with MIC in 60 s. When the access point resumes operation, it requests that all clients attempting to re-connect change their keys.

To avoid the noise, inevitably present in the environment, activating the TKIP, the MIC validation process is activated after a certain number of validations. The validations consist of certain controls such as frame check sum (FCS), integrity check sum (ICV) and TKIP or TSC. If the noise interferes with packets, changing them, one of these checks highlights it initially, thus preventing the frame incrementing the MIC countermeasures counter.

6.11.10.3 Advanced Encryption Standard

Advanced Encryption Standard (AES) is a system of encryption that has already been discussed, in detail, in Chapter 2.

802.11i uses it within CCMP that is based on CBC-MAC. CCMP was chosen for the control of data integrity and authentication, where the MAC performs the same functions as MIC within TKIP.

AES operates in two ways:

1. counter mode (CTR), which is used for confidentiality;
2. CBC-MAC, used for integrity.

AES combines CTR and CBC-MAC to create the so-called CMC, which is the acronym for CTR/CBC-MAC, which represents a mode of operation of AES that contains both the confidentiality of CTR and the integrity of CBC-MAC.

6.11.10.4 Features of 802.11i

Once the various parts that make up the 802.11i standard have been illustrated, it is possible to explain this standard in general terms. The standard procedure involves the connection of the client to the access point, their authentication and key negotiation.

The first step is to perform a connection to the access point. This process is carried out through a normal open key authentication, as described above. Contrary to most 802.11 standards, only 802.11i allows open key authentication. This is because of the discovery of the security gap in the process of shared key authentication already described above.

After the initial connection request has taken place, the client waits for the transmission, in broadcast, of an RSN IE or requests the same via a probe. Once the RSN IE frame is received, the client and access point must negotiate a cipher to use. Once a cipher has been negotiated, the EAP process can begin. This begins by sending, from the access point, a request for identity to the client or by the sending, by the client, of an EAPOL start frame. Once the EAP process has started, the identification phase can take place that depends on the particular type of EAP being used. This process, as described previously, ends with receipt by the client of an appropriate confirmation message sent from the access point. During this process, an AAA key will be sent from the authentication server to the wireless client. This key is used as the seed key to generate the process of key generation described below.

The key exchange process uses the EAPOL-key frame of 802.1x, implementing certain changes that allow the use of the WEP-40, WEP-104, TKIP and CCMP ciphers. To perform key exchange, a four-stage process is used. This process generates two main keys and a single group and session key for each client. The group and session keys are generated by the pair of master keys or PMK (pairwise master key) and by the GMK.

In implementing 802.1x in 802.11i, PMK is provided by the authentication server. It is divided into three keys:

1. confirm key EAPOL or Key Confirmation Key (KCK) (EAPOL-key confirmation key) that is used to confirm the originality of the data;

2. encryption key EAPOL or Key Encryption Key (KEK) (EAPOL-key encryption key that is used to ensure confidentiality;
3. temporal key or pairwise temporal key (PTK) that is also used to guarantee confidentiality. To generate PTK, a pseudo-random function is used that employs the MAC address of the access point, the MAC address of the client and a particular number named *nonce* the term *nonce* indicates a number, generally random or pseudo-random, which has a single use. *Nonce* is, in fact, the contraction of the English words *number used once* sent by both parties. This enables a single master key to create multiple session keys without having to exchange again a new master key each time.

Another very important key is the GMK. This key is very similar to PMK with the difference being that it is used for encryption of the beacon and of the management traffic. The same process of hashing of the MAC of the transmitter and of the receiver and the *nonce* is used to generate a GTK (Group Temporal Key) from a group master key.

Once the process of creating and managing of the keys has been illustrated, it is possible to explain the process of four phase exchange.

This process begins with the sending, by the authenticator, of a *nonce* to the applicant. The *nonce* is a random value, which is used to prevent replay-type attacks. This means that old *nonce*s cannot be used. After each party receives a message, before carrying out any other activity, it is necessary to check that the *nonce* has been changed or if the same has been used several times. Once the client receives the first message, it controls the *nonce and* generates a *Snonce*. This *nonce* is used to calculate PTK Once the PTK has been generated, the client sends *Snonce* and other security parameters of the RSN IE frame to the access point. This information represents the second exchange of the four-phase procedure.

All information is encrypted using the KCK key that avoids changes taking place during transmission. Once the access point receives such information, it checks that the *nonce* is not characterised by an old value. Once this is done, it generates PTK and controls KCK to ensure that it has not been altered during transmission. At this point, the third exchange of information of the four-phase procedure takes place. This exchange is to advise the client to install the PTK key that was created and, if used, this message sends a GTK to the client to enable installation of the same. Once the client receives this information, it checks KCK and, if correct, installs the key or keys. The last message is a confirmation that is used to inform the authenticator that the client has successfully installed the keys and that it is ready to communicate using the same. This four-phase procedure is shown schematically in Figure 6.47.

6.11.11 WPA2

When 802.11i was published, the Wi-Fi Alliance intended to continue investing in WPA. This was a significant problem as the 802.11i standard was ready, and the market wanted everything except a new standard that might generate confusion among the various products and consequent problems of interoperability. To maintain WPA, the Wi-Fi Alliance decided to revise the same WPA.

In the creation of WPA2, the Wi-Fi Alliance created this version to ensure Wi-Fi interoperability with 802.11i.

6.11.12 WAPI

WAPI is the acronym for WLAN Authentication and Privacy Infrastructure. It was created when China realised that the 802.11i standard was too late for its own needs and created its own standard. Even now that the 802.11i standard has been formalised, China has decided to continue to preserve its own approach in the industry. This has led to the creation of the WAPI standard that is characterised by many points in common with 802.11i, such as the use of RADIUS and 802.1x. Information about

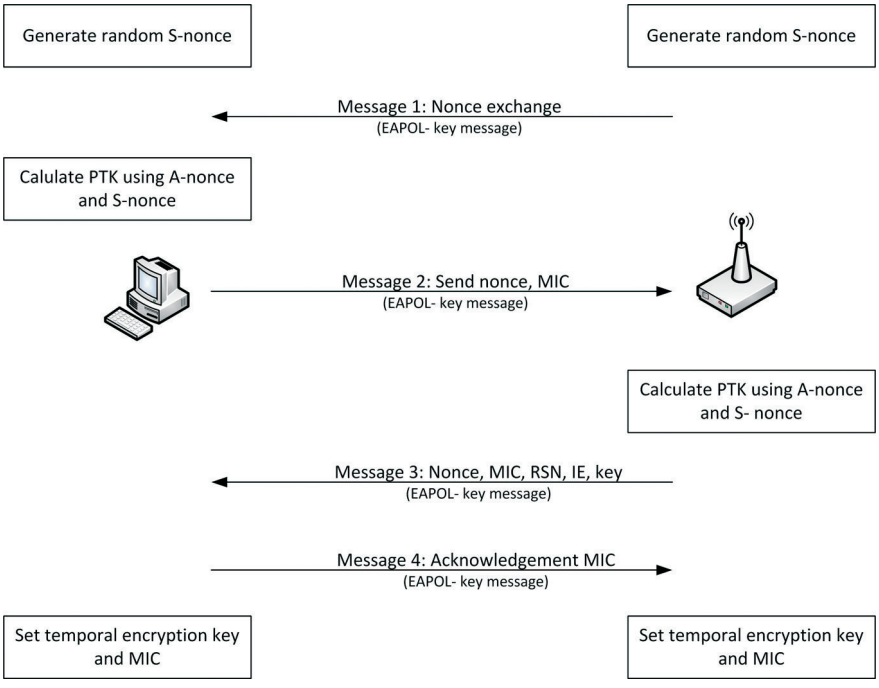


Figure 6.47 Schematic diagram of the four phase process used by 802.11i.

this standard is considered confidential by the Chinese authorities, and the publication of such information constitutes a violation of Chinese national security.

The producers, to sell products, which comply with this standard in China, must have a WAPI licence through an agreement with one of the 24 Chinese companions that are guaranteed rights by the Chinese government. The Chinese government, to allow the sale of these products, requires that producers release the intellectual property related to the development of electronic chips. As this intellectual property represents the strength of producers, many of them have ignored the Chinese market.

6.11.13 Detection of false access points

The detection of false access points represents an activity that has developed its own share of the market. In fact, many producers have developed by offering products aimed at solely identifying fake points of access.

The risks that are run when in the presence of false access points have already been shown above.

There are various detection techniques used by various devices on the market and will not be shown in the following for reasons of space.

6.12 Violation of wireless security

This section explains the main concepts on ways of violating the security of wireless networks and related activities that are carried out, in this sense, by attackers.

The main vulnerability of the networks themselves and countermeasures that can be implemented to defend them will also be illustrated. Addressing the issue from the point of view of attackers allows for proper risk assessment.

6.12.1 The process of attack

The process of attack begins with a typical activity of *war driving* to intercept the wireless network, object of the attack. Then the attack is planned which is then followed by the actual attack. This sequence represents the phases according to which most attacks develop, and that will be illustrated, specifically, below.

6.12.1.1 The acquisition of information

The first step consists of the acquisition of information represented, mainly, by a *war-driving* activity or similar activity. During this phase, the attacker is not usually searching for a specific goal or a specific organisation. He/she is generally searching for a network where the security or free Internet access can be ascertained.

Once the attacker has decided to plan an attack, the phase of acquisition of information takes place. He/she must use as much information as possible and may use the Internet as provider of such information. In this way, the attacker can acquire a considerable amount of information on a specific subject or on a specific company.

During this phase, the attacker tries to acquire information on the network products that the organisation uses with particular reference to the wireless network. Very often information request and clarification emails about wireless products are found that members of the organisation have sent, perhaps, to work groups that operate on the Internet. Information can often be found about the names of the staff that work within the organisation, object of the attack. In some cases, the existence of a commercial relationship between two organisations may also be deduced and therefore the relative close connection between their networks: in this case, the attacker can decide to perpetrate the attack against the subject that possesses a vulnerable network and use it as a service access port and be able to access the main network.

Another source of information is the domain names registry, in a manner analogous to fixed networks. On this site, all the information relating to the organisation is publicly available that can be used to conduct social engineering (as explained above).

6.12.1.2 Numbering

The numbering, identification and scanning represent activities that are executed once all the information has been captured. In the following, these activities are indicated with the cumulative numbering name.

At the numbering phase, the attacker searches for any device with which he/she can connect to the network, attempts to understand what type of products are installed, the producers of the products and the version of software or firmware that runs on these products.

The numbering phase does not necessarily follow the information acquisition phase, and it is also performed, for the most part, during *war driving*.

During the numbering phase, the attacker tries to acquire exhaustively all the information required to be able to move on to the next phase that consists of attempting to enter the network. Once entered, the same must initiate a new numbering phase to catalogue all the devices that are present within that network section. This activity of numbering, breach and new numbering may be repeated several times until the attacker manages to acquire all the information relating to the entire network.

During the numbering phase, it is possible to use a considerable amount of commercially available tools, most of which are scanners. The task of scanning for wireless networks represents a subgroup of activities that a scanner can perform. The most popular type of scanner is known as the port scanner that searches the open Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) ports. After having identified the open ports, the attacker can use a scanner to identify devices by

requesting and examining the information received via these open ports. Most of the communication of the operating systems that occurs through these ports is typical of the same operating systems. A scanner with the ability to examine a network may also examine the response traffic from a device and understand what type of operating system is installed on it. This is possible because some operating systems handle TCP/IP traffic differently and operate with a packet format that allows a return to the same operating systems.

Another type of scanner used is the one that uses SNMP and Internet Control Message Protocol Internet Control Message Protocol (ICMP) to map the network and its related devices. Once all the information are gathered, this scanner allows the structure of the network to be acquired. This type of scanner is usually included in the port scanner and must operate before the port scanner to understand the port of the device to be scanned.

The last type of scanner is the one used for security vulnerabilities. This type of scanner looks for a determined network device to search for vulnerabilities and security flaws. This operation is performed in a relatively short space of time. This represents the last activity of numbering.

Once all the information are gathered, they need to be processed in order to obtain a clear and organic picture of the network and to prepare for the next phase of the attack, that is breach.

6.12.1.3 Breach

Breach is the phase that follows numbering. If an attacker can breach a wireless access point, the same is able to penetrate within the network and carry out its attack.

Breach involves a small number of tools. To breach a device, a vulnerability must be found or created. To find a vulnerability, we need to know where to look. Often these vulnerabilities are available on Web sites not known to the general public. In most cases, once a vulnerability is available to the general public for a very long time, appropriate tools are created that are very easy to use.

6.12.1.4 Acquisition of privileges and accessibility

The acquisition of privileges and accessibility is a very important step in the attack process. This step occurs after the network has been breached. Once a network has been breached, there is not much else to do except go to the next network objective. This step becomes very important in relation to the servers and other network devices, access to which is not completely restricted but is allowed under appropriate conditions. The process of privilege acquisition involves attacking a device whose access is restricted and taking advantage of vulnerabilities to try to change the level of access from restricted to open.

When attempting to exit a network that has been breached, the access doors are often left open to make entry a second time simpler and easier. There are many ways to leave the access ports on wireless devices open. In this sense, it is possible to use the appropriate tools that will allow the attacker to receive the remote shell. Once the attacker receives a remote shell, he/she can install and activate the appropriate remote control software to be able to control the same devices from a distance and be able to access the network, without problems, at any time.

There are various techniques to install an access port.

The first is by breaking into a laptop used by an employee of the organisation that is used both outside and within the organisation and thus that has network connections. If an attacker manages to violate that laptop, he/she is able to change its settings and to use it as a router that guarantees access to the organisation's network passing through it.

6.12.1.5 Removing traces

When an attacker enters a network, he/she inevitably leaves traces that are recorded by the various network devices to which access is attempted such as the log files or in any case by the various security devices, such as firewalls and intrusion detection systems.

If the attacker is not able to perform this operation, the organisation is able to discover, in real time or at a later date, that the attacker entered the network, where he/she went and the activities that were performed.

In the case of wireless networks, an attacker could, to avoid leaving traces, attempt to alter the log file of access points being violated and of the other network devices. For example, the port of a switch will register the MAC address of the attacker even though the same entered through an access point. The same is the case with regard to attempts to access servers. Things get complicated if there is a syslog server which receives all the log files of the various network devices; in this sense, it is also necessary to attack this server to remove any traces of the intrusion.

It must be remembered that when an attacker tries to access a device, the traces of the various attempts remain within the device even if the attempted attack was not successful.

Different techniques are used to alter the log files, and in this sense attackers may have a large number of tools that are readily available on the network.

6.12.2 Breach technologies

This section illustrates the various breach technologies known to attack a wireless network. Once these techniques are known, it is possible to perform a correct assessment of the risks run by the various protocols, methods and security standards used on a particular wireless network.

When an attacker penetrates a network, it is very important to understand how the same succeeded in doing so and to prevent future intrusions by the same or other attackers.

6.12.2.1 WEP

WEP is the wireless protocol that has undergone the greatest activity of analysis and with which all the vulnerabilities were exposed. This was analysed in each of the areas relating to confidentiality, availability and vulnerability. In this sense, in the following, the main types of attack will be illustrated in three different areas.

The first type of attack is the flow cipher. This attack works passively, and it is therefore impossible to detect its activities as the attacker analyses only the traffic that intercepts, without generating any type of traffic within the network that, inevitably, would leave traces.

This type of attack exploits the vulnerabilities of the numbers used to generate IV, which is used in the WEP operating sequence. IV is used to create a different key for each of the packets transmitted. In this sense, approximately 9,000 IV numbers of interest of the 16,777,216 available have been identified. These numbers are considered interesting as there is an "FF" in the middle of the IV sequence, which is defined as three groups of two hexadecimal digits separated by columns. In this sense, an interesting IV number has a form of type XX:FF:XX where the Xs indicate the hexadecimal generic digits. Because this number is encrypted, it is possible to have information both on the plaintext (FF) and on the corresponding encrypted part allowing a cleartext attack which, if conducted properly, may violate the data transmission cipher.

Another type of attack is the known cleartext attack. This can be conducted when the cleartext and the corresponding cipher is known, allowing the encryption key to be derived.

When the original text is encrypted, an XOR operation is performed that blends together the key and the cleartext. This process is reversible and, once applied to the ciphertext, cleartext is returned. It

is evident that if we know the cleartext in binary version and the corresponding ciphertext, in binary version, it is possible to trace back to the encryption key.

The first attack that can be conducted refers to the shared key authentication. We have seen above that in this type of authentication, a packet containing the cleartext is sent, and the ciphertext is expected with the above shared key.

Other types of attack can be conducted depending on the network traffic, which must be identified beforehand. For example, when we turn on a device, the same performs a boot sequence that involves the authentication network; if this traffic can be intercepted, all the information is available to perform an attack of this type. The same can be performed when there is traffic relating to Domain Name Server (DNS), logon, DHCP, etc.

Another type of attack can be conducted by generating, beforehand, appropriate traffic. This can be done by sending an ICMP, telnet or another appropriate request. Such requests may be made only if we are connected to a section of wired network. Another way to acquire information is using email traffic. To be useful, it is important to know the MAC address of the sender and the MAC address of the recipient. Such information can be acquired from the traffic of the access point or captured in many other ways.

The typical sequence in which a shared key authentication attack is followed is shown as follows:

1. Alice tries to connect to the network.
2. The access point sends a cleartext challenge.
3. Alice receives the challenge packet, the cipher with its WEP key and sends it to the access point.
4. Eve intercepts the packets, extracting IV, which is sent in cleartext and attempts to reconstruct the key, by acquiring important elements through an XOR operation between the challenge text and the corresponding ciphertext sent back from Alice.
5. At this point Eve can try to connect to the network.
6. The access point sends Eve the challenge text.
7. Eve, upon identifying the key, encrypts the challenge text received with this key and sends it to the access point.
8. The access point verifies that this string was encrypted correctly and allows connection to the network.

Another type of attack that can be conducted against WEP is dictionary attack that is carried out using the cleartext cryptoanalysis attack already shown above. Once this has been done, each frame characterised by the same IV as that previously breached may also be violated in the same manner. This means that it is possible to conduct this type of attack on multiple frames until every number in the IV space has been violated. This type of attack is not particularly onerous from a computational point of view, and it has been demonstrated that the same may be conducted by resorting to a 25-GB archive. Once the same has been developed, all the frames of the network can be traced to a cleartext until the seed key of WEP is changed on all the client devices and all the points of access.

Another type of attack that can be conducted against WEP is the double encryption attack and exploits the fact that the same key is used both to encrypt and to decrypt. To conduct this type of attack, at least one frame must be captured, as the attack can be conducted on one frame at a time. Once the frame has been identified, the address of the MAC destination and that of another wireless client must be changed. After this, the attacker must wait for IV to put it to one less IV original. At this point, the attacker can send the captured frame into space. When the access point receives the frame with the correct IV, it puts the encryption process in place, in this case deciphering the frame instead of encrypting it. Once the access point has completed the encryption process, it sends the frame in cleartext to the MAC address that was specified by the attacker.

Another type of attack is message change type. In WEP, it has already been said that the IC (integrity check) field is used to verify the integrity of the exchanged messages. This 4 byte value allows the access point to check if a frame has been damaged during transmission. As expected from the

specifications, an access point ignores a frame characterised by an incorrect IC value, and this is done to prevent the high number of errors that occur during wireless transmissions. A feature of IC is that it is independent of the WEP master key and of IV; this independence allows it to be modified without great difficulty.

To perform a message change attack, the attacker must, first, intercept a packet destined for a certain subnet. Once an encrypted packet has been captured, the attacker must change a single bit and try to send it: in most cases, this causes an error detected by IC, and the packet is deleted. A number of attempts are then made to change until they find the one change that does not cause errors in IC even if the packet appears to be nonsense. This process can be repeated as many times as needed, as the access point does not require any login and does not generate any warning message of the activity in progress. Once the packet exceeds the integrity check (IC), it is sent to the router that verifies that an error exists in the packet and sends a reply to the sender. Once a response is emitted into space, the attacker receives a cleartext associated with the encrypted text contained in the packet sent: this allows cryptoanalysis to be performed on the cleartext.

6.12.2.2 DoS attacks

Another attack is DoS. We have seen that networks based on 802.11 standards use a number of management frames. These frames warn clients that they can connect or that they must disconnect. It has also already been seen that there are additionally de-authentication frames that expel a client from an access point. These frames, as is the case with all other management frames, are sent in cleartext every time WEP is used. This means that these frames can be used fraudulently to expel the client from the network; this operation can be performed in different ways.

One possible way to prevent such attacks is to encrypt all the traffic in the second network layer.

Another possible attack is the EAP DoS attack that exploits some of the weaknesses of the EAP architecture. This involves the sending of frames that have been suitably modified according to different techniques.

One technique consists of sending EAP Stat type frames to an access point. If the access point is not able to correctly process these frames, there is the risk that the access point can reboot or crash, allowing the attacker to enter the network precisely when the security level of the access point is at a minimum.

A second technique consists of sending, to the access point, suitably modified EAP messages. Some types of these messages may cause blocking of the access point or RADIUS server.

A third technique consists of sending, to the access point, EAP frames where the identification space is filled, where EAP provides for 255 different types of identification data to keep track of each client's request. In this case, the attacker can overload the access point with a large number of connections that fill this field, making a DoS attack possible.

6.12.2.3 MAC filtering attacks

MAC filtering attacks began once all the WEP vulnerabilities began. This attack relies on the fact that a MAC address can be changed with relative ease using different techniques.

MAC addresses can be viewed using *sniffers*, *and* this possibility is exploited by attackers to find the MAC addresses of devices using the network. Once this has been done, attackers may vary the MAC address by using different techniques. After they have changed their MAC address, they can send a de-authentication frame to the original user or enter the network on which it is found, there being, however, a conflict between MAC addresses.

6.12.2.4 RADIUS vulnerabilities

It has already been seen that RADIUS uses a shared secret to enable communication between a RADIUS server and a RADIUS device. The shared secret is generated with the MD5 hashing algorithm. The shared secret is calculated by the code, the ID, the length, the request of the authenticator and by the attributes of the authenticator response. These values are inserted in MD5 to generate an encrypted string. As MD5 is a one-direction hash function, it cannot be violated, although it is possible to use a brute-force attack, to try every combination of password and compare the result to the original hash; if the result matches, then the original string has been found. This procedure can be put into practice on RADIUS frames as all the above values, except for the password that is located within the hash, may be intercepted by a sniffer.

If an attacker is able to eavesdrop on the authentication process, he/she can take the frames and implement a brute-force attack on them. These frames, by definition, are not authenticated by a RADIUS server, and this allows anyone to begin a process of RADIUS authentication and to intercept the shared key that has undergone the hashing process. It is also very important to be in possession of the correct IP address that is defined in RADIUS, and this can be done through s*poofing*.

Another vulnerability of RADIUS relates to the use of an authentication server for other types of means in which the same can also be used for remote access. In the latter case, administrators use the shared secret for both purposes. If an access protocol is characterised by a vulnerability and an attacker is able to trace a shared secret, the same may use a false access point to intercept attempts to authenticate users.

Another vulnerability of RADIUS is the RADIUS server itself. In fact, very often the RADIUS server is maintained by members of the network and not by the server administrators who are more skilled compared to the former. This is especially true in large organisations, where interaction between the members of the network and server administrators is reduced to a minimum, if not altogether absent. This mode of server maintenance can create vulnerabilities within the server itself, which can be used by external attackers to break into the system.

6.12.2.5 802.1x vulnerabilities

The 802.1x protocol is also characterised by a number of vulnerabilities. This protocol was originally used on wired networks for port-based authentication. In the field of wired networks, it is a good practice to ensure a high level of network security, to keep all network equipment in areas that are enclosed and characterised by controlled access. The only way to connect to the network is via a pin. In the wireless industry, this is not possible because access of the network is through access points that are widely scattered, to allow access to users from all the desired points. This means that false access points can also be used with ease to acquire the credentials of users to then be used to access the network and violate it.

When the 8021.x protocol was created, the vulnerability that could result from its use in the wireless industry was barely considered, being the same created for the use in wired networks. In this sense, the 802.1x standard has undergone changes, together with the EAP standard, to match the needs of wireless network security.

To allow each part of the network to communicate with an authentication server, the way for possible frequent man-in-the-middle attacks is opened. The most famous vulnerability of 802.1x is the use of false access points that allow the very easy acquisition of credentials of authorised clients.

There are different attacks that can be conducted and the most popular is the MS PEAP 802.1x attack given the spread of this standard and the reduction of costs that is obtained by using an MS server. The first step is, as usual, the search for a wireless network that uses the standard and this can be performed through *war driving*. Subsequently, once the network of interest has been found, *sniffing* of the network traffic is commenced to ensure that the same is using the PEAP 802.1x standard. To do

this, keeping a safe distance from the site of interest, use is made of Yagi-type directional antennae to capture wireless traffic. Another way is to enter the entrance hall of the site and to try to stay there for as long as possible, with the most varied reasons such as a request for information or searching for work. During this period, the attacker can use a PDA to accomplish *sniffing* of the network traffic.

The next step consists of finding SSID, which is fairly simple, although 802.1x encrypts the management frames. In this case, it must always be remembered that probe requests and responses always contain a plaintext SSID.

Once SSID has been found, the attacker must install a server and a false access point. The server can run on a normal laptop computer, not forgetting that the same must support features such as DHCP, DNS and Web server. Then the attacker must install the false access point characterised by the same SSID as the one that intercepted. This access point need not include security measures and should emit at the highest power to ensure the maximum coverage area and achievement of the greatest number of clients.

Once the attacker has connected the access point with the server, the same must connect a YAGI aerial to the access point itself and direct it towards a window of the target site of the attack to allow the electromagnetic signal to penetrate within. Once this has been done, users will see this network and its related devices will try to connect. Users who attempt to connect will receive a message advising them that the network they are connecting to is not secure; some users with a minimum of technical expertise will opt for no connection, but the majority of users, lacking in technical skills, will connect to that network. The goal is to have at least one person connect.

Once someone logs in, the attacker's DHCP will provide such person with an IP address, a default gateway and a DNS. At this point, the subject under attack will attempt to use a network service such as email or the Internet. The main purpose is to encourage them to connect to the Internet. Once the same makes such an attempt, a DNS request is sent to the default gateway of the DNS server on the attacker's computer, which, in response, will provide the IP address of its Web server, which is what the user was requesting; in this way, the user has been hijacked on the Web server of the attacker without the same being aware of this.

Once the Web page opens, a Java script opens a pop-up window. This window has a very similar appearance to the authentication window of 802.1x that the user has already seen in previous authentication attempts. Within this window, the user will enter a username and password and will send this information to the attacker, without being aware of it. Once the attacker has acquired such data, the same will turn off the access point and the attacked subject's client will attempt to reconnect to the original network, starting a new authentication process. In this way, the attacker is able to acquire the credentials of an approved person within the organisation without the latter realising anything; the only variation with respect to the normal routine is the double authentication request, which passes unnoticed as, ultimately, the user is connected to the legitimate network. At this point, the attacker has valid credentials without having had to carry out other types of attacks that could have alerted the internal network. The only way to detect a false access point lies in the use of suitable intrusion detection systems.

6.12.2.6 MIC attacks

MIC was created by providing, within the same, a system for preventing change message attacks. It has already been said that MIC turns off the access point if the same reveals two errors in a pre-defined period. When the access point turns off, it does so only for 60 s, after which it re-activates all the procedures to link in all the users that were previously connected. A possible attack consists of sending incorrect traffic to the access point. This traffic could pass the Inizialization Vector Check (IVC) WEP and CRC control, but once it reached TKIP, it would activate the counter-measures. In this way, an attacker could cause the network to malfunction.

6.12.2.7 Attacks on wireless gateways

Most wireless gateways involve authentication forms. As these must support many authentication forms, to support the majority of existing clients, the risk of vulnerability increases because the vulnerabilities of the various authentication forms accumulate. SSL is often used that requires the user, using an appropriate Web page, to authenticate accessing the gateway. This mechanism can be used to create an opportunist attack.

The first step of such an attack is to provide a proxy server to be placed between the gateway and the client. When the client activates the wireless connection, it connects to the proxy and then the client and the proxy establish an SSL connection. Once this has been done, the proxy establishes another SSL connection with the wireless gateway. Once both connections have been created, the user receives a request for authentication from the wireless gateway. This authentication may be easily intercepted as the authentication-encrypted traffic is decrypted by the proxy server and encrypted again to be sent to the gateway.

Another type of attack involves the creation of a false access point and a false network that will lead users to authenticate without hesitation. Such an attack should provide for the use of a Web site that is similar to that of the gateway device in a way that does not arouse the suspicion of the user during the authentication process. When the user connects, this false Web site steals the credentials of the user, which have been provided to access the network.

6.12.2.8 WPA and 802.11i attacks

In 2003, an article was published that explained how it was possible to execute a dictionary attack on WPA and 802.11i. This was possible using the information exchange that takes place during the four-phase process used by WPA and 802.11i to generate session keys.

The process for creating session keys by 802.11i has already been shown above. This process requires the main master key, two *nonces* and the MAC addresses of the sender and receiver; these data represent the algorithm input. When using the pre-shared keys, the master key is generated using a passphrase, SSID and the length of SSID. These fields are supplied to an algorithm called PBKDF2, which performs a hashing function 4,096 times, generating a 256-bit key. If an attacker wants to perform the same operation, the same should possess SSID, the length of SSID and the passphrase. Of all the information just indicated, the only confidential information is the passphrase as the SSID, and its length can be acquired through *sniffing*. Once all the information is obtained, it is possible to carry out a dictionary attack to find the passphrase and be able to breach the network.

In this sense, once SSID has been intercepted via a sniffer, the attacker must observe the four-phase process, which is used for the generation of session keys. In the second phase of the process, it has already been seen that an EAP message is sent containing the two PTK and CEC values that undergo a hashing process using MD5. This hash value allows the testing of passphrase combinations to find the one that is used.

6.12.3 Access point breach techniques

So far attacks on wireless network security mechanisms to access the same networks have been shown. There is, however, a type of attack that can be carried out regarding access points as the majority of network devices are unfortunately characterised by certain vulnerabilities, which depend both on the protocols used and on the specific implementation of the various producers.

Most of the attacks are carried out by exploiting the remote management of access points. In fact, there are different techniques that are used to connect to the access points from a remote location and to manage them. Some of these techniques can have intrinsic vulnerabilities that can be exploited to conduct specific attacks. Knowledge of these vulnerabilities allows decisions to be made regarding

activities that may be conducted on remote access points, reducing the risk of attacks on the same and increasing the overall level of security.

6.12.3.1 Telnet

It has already been seen that Telnet is used for establishing remote terminal sessions on a device. It enables an administrator to properly configure a remote device with an IP address.

Telnet is a protocol that was created in 1960. Owing to its date of creation, it is characterised by a number of security vulnerabilities. One of the biggest vulnerabilities is the non-use of cryptography as all the traffic generated is in plaintext. This means that all the authentication traffic that occurs within a Telnet session can be easily read by network sniffers, and in this way, it is possible to acquire the information relating to usernames and passwords, where the same can be used to enter the access point at a later date, opening a security gap in the network.

6.12.3.2 HTTP

Another technique used to manage an access point is the use of a Web browser and most of the more recent access points include the use of this medium, as it is very easy to use. It has already been seen that http pages are susceptible to automated attacks of password generation attempts, trying the passwords contained in a dictionary, or, more simply, attempting a brute-force attack. Because of the possibility of conducting this type of attack, many manufacturers have removed the option of using such means of access point management.

6.12.3.3 RADIUS

Most organisations have realised that it is not very secure to hold lists of usernames and passwords within the network devices; for these reasons, solutions based on AAA have been developed. These solutions, in most cases, use RADIUS. Due to the vulnerabilities of RADIUS in message management, attackers have developed a series of instruments that are able to capture RADIUS messages and remove the keys. This operation is limited only by the significant amount of time and resources required but in any case can be performed. If we are able to intercept the initial traffic between a user and an access point that uses RADIUS, we can attempt to decipher this traffic at a later time to obtain usernames and passwords.

6.12.3.4 SNMP

SNMP is very similar to Telnet. It is used to control and manage the various network devices remotely. It represents a very powerful tool, if handled by the network administrator, and a very dangerous tool, if managed by an attacker, which could effect any type of change in network devices.

The security mechanisms of SNMP are very simple and are based on two strings: one string allows read-only access, whereas the other string provides write-only access. These strings must be entered into the system and are very similar to passwords meaning that if the same fall into the hands of an attacker, the attacker may use them to access in read or write mode.

If an attacker can access in write mode, the attacker can reboot a device, change the configuration or delete an existing password. An attacker, in order to come into possession of these strings, must perform the appropriate operations, and in this sense there are a series of automated tools to generate a SNMP attack. Once such an attack has been successfully carried out, there are strings that allow access to a certain network device. Often strings are encrypted and a second attack on the cipher used to decipher the same is required.

6.13 Wireless security policies

The security policies of a wireless network are a very important element to ensure their full capabilities and their security.

The first step is represented, as usual, by analysis of the risks. The result of this analysis is all the information concerning the vulnerabilities of the network.

The risk analysis results are used to plan the activities that must be performed to increase the security of the network and the relative priorities of the interventions. This leads to the establishment of security policies.

Many concepts have already been illustrated with regard to the security policies of wired networks, and this section will partially resume and further examine them, whereas many of the concepts are very typical of wireless networks, even if the same can constitute elements that are also useful for wired networks.

6.13.1 Introduction to security policies

When a security policy is to be developed, be it wireless or any other type, a standardised process for the achievement of certain and practical results must be followed, which are shared by the entire organisation within which such a policy must be applied. This policy must be in accordance with the general guidelines of the organisation, and for this reason the development process should also involve management levels within the organisation.

It is very important to have clear definitions along with the purpose of the policy as, very often, documents are written that are halfway between policy, procedures, standards and guidelines. For this reason, the following illustrates these concepts to understand the differences between them.

It is additionally very important to understand why an organisation needs different types of documents to contextualise a single topic, represented, for example, by wireless networks. One reason for this is compatibility as many organisations need to operate according to certain laws and procedures established by the regulations in force. Another requirement is confidentiality, because very often policies contain confidential information that must not be communicated to the clients, which require only standard documents that illustrate how certain objectives are achieved. Another reason is the useful life cycle of each document. In fact, the larger the organisations, the longer the time required to develop a document, because of the larger number of subjects that must be involved. On the other hand, the frequent changes in technology, hardware, software and experience with time require frequent updating of these documents. For these reasons, diversified and efficient documentation must be created, which will be shown in the following.

6.13.1.1 Policies

Policies are high-level documents that are used to indicate the strategic directions of the organisation. They must be very clear, concise and indicate the overall vision of the organisation by its directorate. They do not contain specific elements but instead the goals of the organisation and how leadership intends to achieve them. The implementation and realisation details are contained in another document.

6.13.1.2 Standards

Standards indicate how to achieve the objectives of the policies. The standards contain details on the choice of technology, the software and hardware. This document indicates, in addition, the configuration to be set on each device and the version of the software or firmware that must be installed on the above devices.

In essence, the standards indicate how to reach the objectives where the policies indicate the objectives that must be achieved.

6.13.1.3 Guidelines

Guidelines are very similar to standards in the sense that they indicate how to achieve the objectives stated in the policies. However, guidelines use a different approach in that they do not contain imperatives but instead instructions to be followed. In this sense, guidelines are intended to comply with policies, although there is no obligation to comply with them. Another use of guidelines is to provide general indications, to leave a certain operating margin for the various subjects of the organisation concerned, while the standards are being developed. In many organisations, the directorate uses the technique of leaving the individual groups or departments to implement the initiatives of the organisation and then chooses the best document to become a reference. In this approach, the standard represents a reference to ensure that the organisation complies with certain requirements, whereas guidelines are created by individual groups or departments.

6.13.1.4 Procedures

Procedures explain the key points contained in the standards and guidelines. Procedures apply to a restricted subset of the standards. They are used as reference guides for operating staff. In this sense, a procedure defines the details of a compulsory activity concerning a certain part of the standards or guidelines.

6.13.2 Drafting of security policies

The drafting of security policies is divided into different phases.

The first step consists of risk evaluation, intended to generate a policy of functional security. If the level of risk that the organisation must address is unknown, it becomes impossible to create a security policy to reduce the risk in question.

Once risk assessment has been performed, senior management must prepare the general guidelines concerning the risk priorities to be addressed. The involvement of senior management is very important at this stage, otherwise it is extremely difficult to achieve pre-determined goals as staff will tend not to accept the above-mentioned policies.

Once senior management has identified all risk and has assigned to each of them a priority, a working group must be set up to prepare a draft of the security policies. This draft is generated by examining each risk according to the relative priority and the list of measures that must be implemented to reduce each specific risk. In some cases, certain risks may be transferred to third parties, in this case insurance companies, through the entering into of specific policies.

Once all of the controls have been generated, the next step involves division and approval. During this phase, other subjects that were not initially involved in the activity of drafting, such as departments, individual groups or operating units, are invited to read the prepared draft and to express their comments. This is necessary because a draft may contain goals that are not achievable by those directly involved.

Once all the relevant parties have viewed the draft, expressing any comments, senior management can incorporate the same, change the draft and approve the same as a final document.

Once the document has been approved, an appropriate working group should be established in such a manner as to put into practice the content of the document within the organisation.

6.13.3 Risk assessment

It has already been said that the first task that must be carried out to develop a correct security policy consists of risk analysis or evaluation. Correct risk analysis identifies all the risks currently present and their controls, or counter-measures, which must be implemented to reduce the risks identified. Such an assessment could have a broad spectrum, analysing all of the risks that the organisation must address, or be more specific, analysing specific risks such as those that concern the use of wireless networks. Once such information has been deducted, an overview of the controls and actions that must be implemented within the security policy to reduce the risks identified can be obtained. This evaluation provides the finalised group with a draft of the security policies of the greatest risks and risks considered acceptable. This prevents the working group from wasting time in the development or improvement of control that may already be considered good quality.

The terms used and the strategies identified depend greatly on the subjects that develop them. There are various approaches that can be used by organisations to develop risk analysis, and in that sense insurance companies have well-established methodologies as risk analysis is at the heart of their activities, and their survival is directly linked to their ability to correctly quantify the risks in the various areas in which they are active. When an organisation takes out an insurance policy with an insurance company, the same transfers the specific risk contained in the policy to the insurance company, in return for payment of a fee assessed by the insurance company by means of appropriate risk analysis.

This risk analysis is based on three fundamental elements:

1. identification of the various risks;
2. quantification of the impact that this risk has on the organisation (magnitude);
3. time frame of the risk (probability of occurrence or, very often, frequency of occurrence).

Once all the risks have been properly evaluated in terms of magnitude and probability of occurrence, it is possible to use this information to perform an assessment. In this sense, reference parameters that numerically quantify identified risks are used. These parameters are the exposure factor (EF), the annualized rate of occurrence (ARO), the single loss expectancy (SLE) and the annualised loss expectancy (ALE), and are discussed in the following sections.

6.13.3.1 Exposure factor

EF is used to measure the impact of the value on assets. It is expressed in percentage terms. This percentage expresses the amount of loss that characterises an asset. If an asset is an industrial secret, its loss is 100% if a competitor manages to obtain it. This value is used to calculate the single loss expectancy or SLE. This also helps in calculating the annualised loss expectancy or ALE. These quantities are defined below.

6.13.3.2 Annualized rate of occurrence

ARO is used to measure the frequency of occurrence of a risk on an annual basis. This magnitude is used to calculate another quantity, ARO. ARO can be calculated even if the frequency of occurrence exceeds the one year, and, in this case, it assumes values less than one.

6.13.3.3 Single loss expectancy

SLE is a currency magnitude that is used to measure the economic impact that occurs when a risk is manifested. It is calculated as:

$$\mathrm{SLE} = \mathrm{ARO} \times \mathrm{V} \tag{6.1}$$

where V is the economic value of the asset expressed in local currency.

6.13.3.4 Annualised loss expectancy

ALE is a currency magnitude that is used to measure the economic impact that a given risk can cause in a year. It is calculated as:

$$\mathrm{ALE} = \mathrm{SLE} \times \mathrm{ARO} \tag{6.2}$$

This parameter allows risks to be classified according to this parameter in descending order.

When risk is quantified from an economic point of view, it is said that a quantitative risk assessment has been performed.

When risk is quantified, ordering the same according to a scale of decreasing impact, then it is said that there a qualitative risk assessment has been performed.

6.13.4 Impact analysis

Impact analysis is used for the impact that a possible change may cause. When a hardware platform or a network needs to be replaced, there is always a risk and potential loss that this may cause. In this sense, impact analysis is essentially risk analysis, although it is aimed at analysing a specific event that could result in an impact, including economic, on the organisation in which the change occurs.

This analysis allows for management to have all the information required to better manage the change, avoiding, as far as possible, the occurrence of great losses.

6.13.5 The areas of wireless security policies

Wireless security policies are characterised by well-defined areas that must be understood to ensure the correct development of the same security policies. Such policies should go beyond the normal policies of information systems and should include all the wireless technologies that are used within an organisation. Given that the boundaries between mobile phones, portable computers, PDA, etc., are becoming increasingly adaptable, it is essential to develop a proper policy of wireless security to extend its life cycle as far as possible, encompassing the entire possible evolution of the wireless industry.

Wireless network security policies should be present in every organisation, even those that do not involve the use of this technology. In fact, having this security policy in an organisation in which the use of wireless devices is not provided for reinforces the fact that these technologies are not permitted, explicitly stating the penalties for those employees who use them despite what is provided at the organisational level.

A wireless policy is not easy to develop, also by those companies that have a deep culture of security policies. In fact, wireless policies often overlap other security policies that must necessarily transpose them and therefore be changed.

A possible strategy is to divide the wireless policy into subgroups that can affect specific areas of information security policies of the organisation, leaving the other policies untouched. If the area wireless policy does not impact the other policies, the process of developing them is very easy, as is their updating and maintenance.

Once the risk assessment process has taken place and senior management has expressed its own priorities in this regard, it is possible to start drafting the area security policies. In the following sections, we illustrate some specific areas of the wireless industry.

6.13.5.1 Password management policies

Passwords are a very important element to ensure the security of information, both when it is located within information systems and when it is transmitted through the wired or wireless networks. In this

sense, passwords are used in many ways, and there are many suggested criteria to generate strong passwords. It is always necessary to find the right compromise between the strength of a password and the ease of remembering it without having to necessarily write it down somewhere.

Passwords, *per se*, involve specific risks. In fact, there is a specific software that is able to try a large number of passwords, according to suitable criteria, until the right one is found, and to be able to enter the system or device, thus breaching it. In many cases, passwords are taken from a specific dictionary, giving rise to what is called a dictionary attack. In any case, if a password is not present within a dictionary, it is always possible to try all the permitted combinations, giving rise to what is called a brute-force attack. This attack requires a large number of resources and if the latter are not available in sufficient number, the attack will not be successful within a reasonable amount of time.

A password is considered a single-factor authentication mechanism. This factor may fall into three categories:

1. something we know, such as a password;
2. something we have, such as a badge and a key.
3. something that is part of us, such as a biometric element (fingerprint, face, retina, iris, etc.).

If multiple factors are used at the same time, the overall risk of breach of the system decreases. It must always be remembered that, among all the factors discussed above, passwords are the most economical to implement.

When a password policy is involved, the issue of complexity of the same must also be addressed. The complexity must represent a fair compromise between their strength and their ease of being remembered. If the criteria contained in the security policy are used, a strong password is ensured because its creation reduces the risks of violations of systems and devices.

A fundamental requirement for the creation of a strong password consists of the use of letters, numbers and special characters and of the length of the total string that contains them. If these criteria are being used, we can be sure that the password created will not be found via a dictionary attack but may, perhaps, be found via a long and exhaustive brute-force attack. It is very important to remember that an excessive increase in the complexity passwords can be counter-productive because the same can be difficult to remember and are often written on a sheet that is generally left hidden in the vicinity of the workstation, representing an easy prey for any fraudulent use. From this point of view, a weak password is preferable to a strong one that is written somewhere.

6.13.5.2 Access policies

Access policies serve to define who can access the wireless network and which device can be accessed. This part is very important for the entire security policy.

In the initial phase, it is very important to understand how risk analysis impacts different levels of access. It is also very important to understand the level of risk that can occur when devices that are not standard, and not covered by the policy of the organisation, connect to the wireless network within the same organisation.

Once the risks associated with logins have been identified, it is possible to define the risks associated with the presence of any false access points, which can be installed by external attackers or by disloyal employees of the organisation. In this sense, it is necessary to provide all employees with the information concerning the procedures to be followed in the event that such false access points are discovered. It is also important to define the consequences for any employees who have carried out the installation of false access points, as this can represent a strong deterrent against this type of action.

The access policies must also clearly indicate the devices that may, or may not, access the wireless network. In fact, certain devices, such as PDA, which have an already integrated wireless communication system, can represent a serious threat to the organisation's network because it may interfere with sections of network that manage mobile production devices such as industrial robots. In

fact, these devices can fail in their authentication operation at an access point, thus blocking it, as is also the case with certain security procedures such as those used by MIC; in this case, significant damage would be caused to the productive activities as the communications between the access point and the manufacturing equipment would fall. This aspect might have even more dangerous consequences if a dedicated access point is blocked against activities of health care within hospitals. In this sense, the security policy should explicitly indicate the devices that may connect to the wireless network and the areas where that connection is possible. That section should also indicate the penalties for those who do not comply with these policies.

6.13.5.3 Public access management policies

Public access represents a relatively recent aspect to be addressed in security policies, as it is only relatively recently that access to wireless networks in different places by the public has been allowed.

Currently, there is a widespread of the so-called hot spots which are those specific areas in which wireless coverage is provided that can be free or for a fee. These hot spots are often used to encourage people to come to places where these hot spots are present such as in meeting places, for example bars. Hot spots can also be provided in dedicated spaces within organisations to ensure Internet connectivity to any external visitors. However, these hot spots are not without risk as, however, they allow connection to the organisation's network. In this sense, appropriate security policies in these areas must be provided for in order to reduce, as far as possible, the associated risks.

If companies intend to provide the hot spots internally, they must absolutely separate these networks from the organisation's network and adopt security devices that do not allow access to the internal network via hot spots. The security policy could, in this sense, provide for the non-physical connection between the two networks, and the use of an external Internet service devoted only to the hot spot service. This solution is immediate and affordable because it does not require the use of network devices that separate the two networks, that is that of the organisation and that which is public. Another way to increase security is not allowing employees of the organisation to use the public network. As this can generate ill-feeling among the same, it is always possible to provide a public network dedicated to the employees and a public network dedicated to the visitors. In this sense, the network dedicated to employees should use VPN to allow connections to resources of the organisation. This would allow employees to use hot spots also outside the spaces of the organisation. Regardless of the security measures that are implemented, it should always be borne in mind that hot spots always represent a real risk.

A part of the policy must also address wireless devices. In fact, a large share of the risk actually stems from these devices and not from the network. The use of email services or remote management has, in fact, vulnerabilities that can be exploited by possible external attackers to penetrate the network. These risks are reduced if the connection is via a wired network in order to access the Internet. In fact, in the case of wired networks, even if the traffic occurs on a very high number of devices that are located along the path of established connections, the greatest risk of traffic interception is due to internal staff of the Internet service provider. In the case of wireless, since the signal propagates anywhere within the coverage area, it will be easier for an external attacker in possession of the right technical skills to intercept this traffic and in this sense specific and dedicated security measures must be provided for.

Another part of the security policies should be directed at the theft of information from the organisation from wireless devices. The easiest way to do this consists of the physical theft of the device itself; this topic concerns physical security and is addressed in the following section. If information is considered as separate from the device, there are many ways to steal the information without stealing the device itself, leaving traces in the latter case. The easiest way to protect such information involves the use of strong passwords for access to the end device.

6.13.5.4 Physical security

Physical security has its own characteristics, one of which has to do directly with wireless networks. In fact, very often, access points are installed in areas that are easily accessible, which allow them to be removed without great difficulty, and for this reason a correct security policy should take into account this possibility and provide for proper counter-measures such as those to install access points out of the reach of persons.

Another aspect is the end devices, which, being characterised by size and extremely low weight, can be easily forgotten anywhere or stolen, representing a serious risk with regard to the loss of confidential data of the organisation. In this sense, the use of strong passwords is recommended, and even the cancellation of all the data contained in them when a certain number of invalid password attempts has been exceeded. These devices can also include a GPS system, which allows the exact location of the device to be remotely found.

6.14 Wireless security architectures

This section discusses how it is possible to implement different security technologies to minimise the risks of a wireless network, explaining its merits and defects.

There are basically four types of technologies: static WEP, VPN, wireless firewall or gateway devices and 802.1x.

For operational reasons, networks are divided into three groups according to the number of users:

1. small networks: a single site that serves from 1 to 10 users;
2. medium-sized networks: two sites that serve from 10 to 1,000 users;
3. wide networks: multiple sites that serve more than 1,000 users.

By using this division, it is possible to adapt the different network architectures to the various dimensions provided.

6.14.1 Static WEP

WEP is the first static architecture. It was widely used until some time ago but is being replaced just about everywhere because of its low security. The great advantage of WEP is its speed and its standardisation. Many more older devices are capable of supporting WEP only, which is included in many 802.11 standards.

In a typical architecture that uses WEP, access points are used as an extension of the wired network, and each of them is connected to the network in a manner similar to a hub on a wired network, to extend the number of ports, as shown in Figure 6.48.

The main drawbacks of the use of WEP are security and manageability. In fact, it has already been said that the number of known attacks that have been carried out against WEP is very high. Furthermore, WEP requires considerable management activities, due to, for example, the need to deploy, in a secure manner, all the static keys to allow the various network devices to connect to the same.

However, it is possible to create a solution based on WEP and, even if the same does not appear to be very secure, it is surely the most economic because it requires the lowest number of apparatuses on the landline. WEP is one of the most economical solutions because it is already included in most of the 802.11 standards. However, it must be noted that if, on the one hand, WEP is very economical because it requires a small amount of network equipment. On the other hand, it is very expensive from the point of view of management required. This aspect is often forgotten when cost/benefit

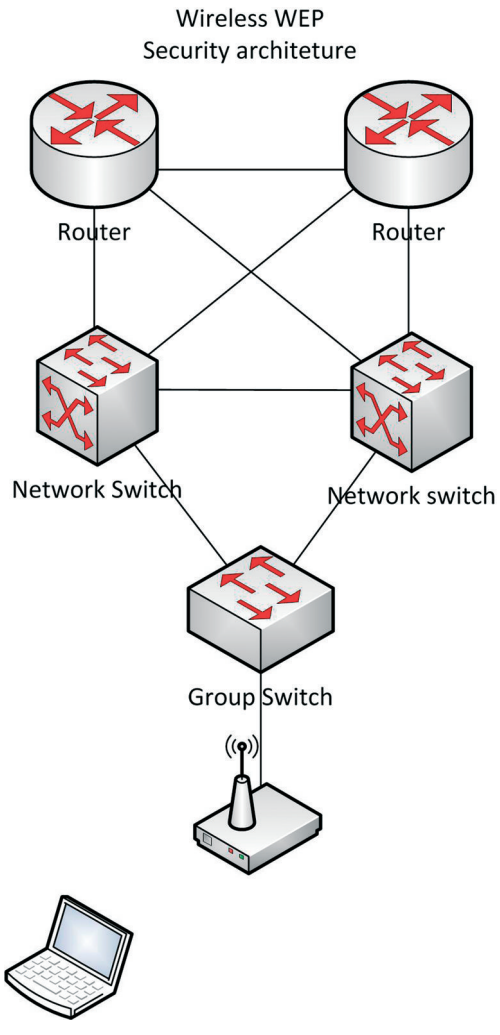


Figure 6.48 WEP wireless architecture.

evaluations are performed when selecting the wireless standard to install. In this sense, it is very important to provide for all the costs of design, installation and support when performing this type of analysis.

The use of WEP for small- and medium-sized organisations is very economical in the short time but costly in the long term. The higher cost is due to the support for the management of keys.

Unfortunately, the great problem with WEP is its low level of security. In this sense, software for breaching the same without great difficulty is also available on the Internet.

Another big problem is the keys. In fact, their replacement cannot be done gradually but must be done all at once for the various wireless devices, meaning it is a very costly operation if the number of devices that make up the network or that use it is very high.

Another problem is that WEP is a computer-based authentication mechanism: this means that if someone is able to steal a laptop that has access to a given network, the same can enter the network at any time without difficulty.

6.14.2 VPN

VPN, which has already been mentioned in Chapter 5, can also be used in the wireless industry. In this case, VPN considers the wireless segment as a danger zone and considers it to be similar to a public network such as the Internet. This is done both at the level of policy and at the technical level, by suitably segmenting the wireless network and positioning it outside the internal trusted network. In this way, all wireless clients access the network as if it were a remote access. As such, wireless clients must establish a VPN session with a VPN firewall, gateway or a VPN concentrator that is located within the network. For this reason, a number of concepts used in VPN will be included in the following along with an explanation on how to use VPN in the wireless industry.

6.14.2.1 VPN technologies

We have already mentioned that VPN is used to create encrypted tunnels within public networks, connecting two private networks to each other. This procedure can be expanded by connecting several private networks, generating an encrypted extranet. An example on a large scale of this concept is the network called ANX, which is composed of different VPN gateways connected to each other to generate a single, large, secure and protected network. VPN technology allows a high level of confidentiality and integrity through the use of cryptography.

In this sense, different concepts are used, such as IPSec, ISAKMP, IKE, AH and ESP, which are shown below.

IPSec is an acronym for IP Security protocol, which is an open-type framework offering the various producers the option of creating secure tunnels. This is done using a set of rules that defines the mode of communication of the devices and the processes of negotiation, creation and closure of encrypted tunnels between each other. IPSec allows different producers to establish encrypted tunnels between their products without having to deal with a prior standardisation process; if their products are in accordance with IPSec, then they are able to operate according to this standard. IPSec is responsible for creating and exchanging keys that are used for the creation of secure and encrypted tunnels within a public network. Once these tunnels have been created, it operates in two main ways:

1. authentication header (AH);
2. encapsulating secure payload (ESP).

ISAKMP is an acronym for Internet Security Association and Key Management Protocol and represents a framework provided for IP services to manage the security association (SA). In this case, the IP service is IPSec. The purpose of ISAKMP is for management of the authentication keys and SA. One of its main purposes is the use of a management protocol that is independent of the exchange of keys: this means that while ISAKMP creates, deletes and manages the security of the keys, it is not able to exchange keys with others.

IKE is an acronym for Internet Key Exchange and is a method to perform an exchange of keys on VPN because VPN can support different ciphers characterised by keys with different lengths. Once these keys have been generated with ISAKMP, they are exchanged by IKE through a VPN tunnel. The keys are therefore generated outside of IKE, which is only concerned with their exchange. This represents a target of IPSec designers that aimed to use one protocol for the creation of keys and another for exchange of the same.

AH represents that part of IPSec that deals with integrity without connection and authentication of the origin of the data in IP datagrams. It can be selected to provide additional protection to IPSec transmissions. AH works by encrypting the IP header and by generating a hash cipher that is used for the control of integrity. It must be remembered that not all the information in the IP header is used for calculation of the hash because some of it may vary during transmission over the network. AH can be

used in conjunction with ESP (which is illustrated below) to increase the security of packets that are protected by IPSec.

ESP is an acronym for *Encapsulating Security Payload* and represents the way in which IPSec operates. It guarantees confidentiality, authentication of the origin of data, integrity, without connection. This is done by encrypting the data part relating to conversation and placing the same in the same dedicated part of the IP packet. This process can perform the encryption on TCP, IP, UDP and ICMP. To make it known that a packet contains an encrypted ESP packet, an ESP header is inserted immediately after the IP header.

ESP can operate in two distinct modes:

1. tunnel,
2. transport.

The difference between these two modes is the encryption of the IP header that provides further confidentiality.

In tunnel mode, the client IP address and the IP address of the recipient are encrypted and placed in the data part of the packet. To ensure that the communication is carried out correctly, the two IP addresses of the IPSec gateways are used, which exchange packets with each other and address them correctly, once received, deciphering the part concerning the addresses. This is not the case in wireless communications, because in this case, communication takes place between the wireless client and the gateway and not between gateway and gateway.

In transport mode, the IP information contained in the header is not encrypted or used with ESP.

6.14.2.2 VPN wireless architectures

When a VPN wireless architecture is created, the wireless part must be considered as a network that is not secure: this means that the same must be separated from the internal network. If the wireless network is outside the internal network, a possible attacker that managed to enter it would not have many opportunities to find the necessary information or carry out illegal activities. One possible architecture is shown in Figure 6.49.

As can be seen in Figure 6.49, the wireless network is located outside the internal network, in the demilitarised zone (DMZ). The internal network is protected, with respect to the wireless zone, by a firewall behind which there is a VPN concentrator. The VPN concentrator is the apparatus where the end wireless devices terminate their encrypted IPSec tunnels. When a client wishes to connect to a network, it first connects to wireless without security measures or with limited security measures. Once connection to the wireless network has been established, the client sends a VPN request to the VPN concentrator that is located within the protected network. This request can pass through the firewall and reach the VPN concentrator. Once the request reaches the VPN concentrator, an encrypted tunnel is created that allows the wireless client to communicate securely with the internal network. This mechanism protects the internal network from attacks from the wireless section and considers the risk of communication with the wireless client similar to the risk of communication with any subject that is located on the Internet.

VPN concentrators can be located locally or at a remote site. If they are located locally, a VPN concentrator must be provided for at each site where there is a wireless network. If these are located in a remote site, only one VPN concentrator should be placed in the centre of each site or in any case at a point marked by high-speed communication. This VPN concentrator serves all the wireless networks of the site. Each of these approaches provides advantages and disadvantages.

With regard to the local installation, all encrypted wireless traffic terminates locally on the local VPN concentrator, allowing it to be decrypted and sent on the internal network. This solution ensures a short response time and greater availability than the centralised solution and, in case of a malfunction of a WAN, this solution is still able to operate properly. In addition, this solution is more secure

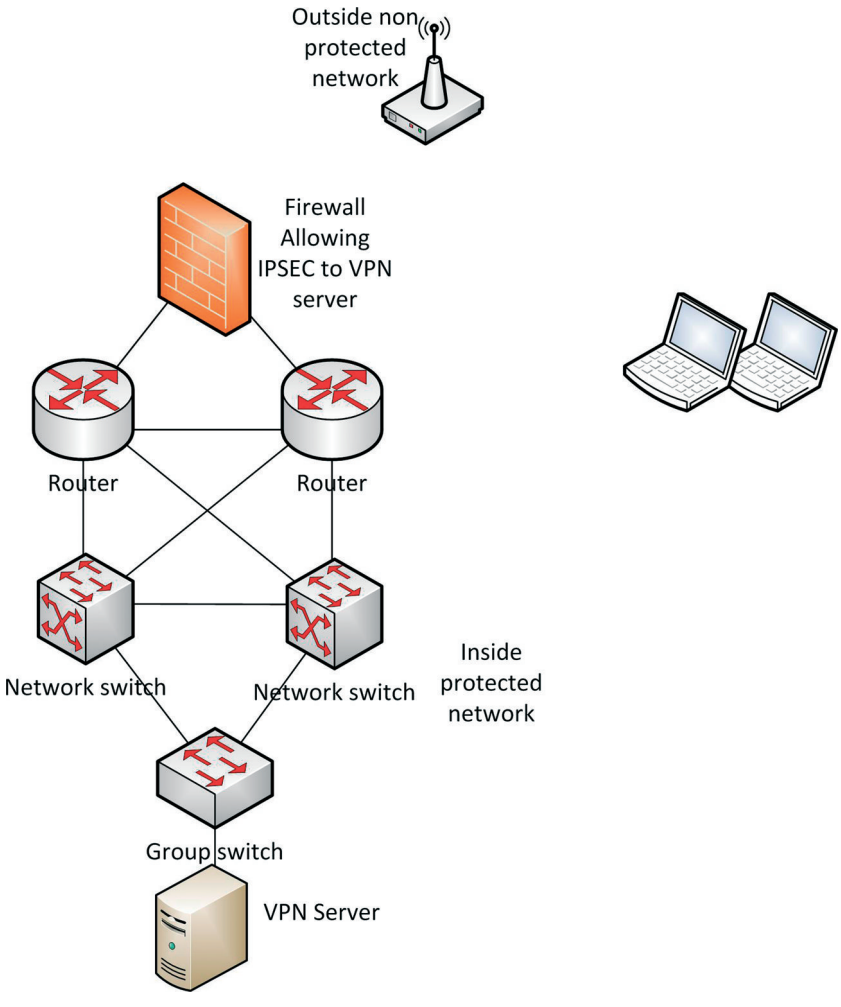


Figure 6.49 Example of VPN wireless architectures.

because the traffic need not enter the internal network for the processes of authentication and encryption. The disadvantages are the higher cost due to the greater number of devices that must be installed and managed.

With regard to remote installation, this involves the use of a remote VPN concentrator. In such a situation, when a wireless client wishes to connect to an internal network, the request must traverse the entire WAN as far as the site where the VPN concentrator is located, increasing the risk because the data is transmitted along with other confidential and private data. Another major risk is that, as already stated, if there is a WAN failure, requests can no longer reach the concentrator, blocking, in fact, the use of wireless networks at all sites. The great advantage of this solution is that costs are extremely low because there is only one central device that must be purchased, installed and maintained.

VPN wireless architecture, as a whole, is very secure because has been tested, for a very long time, for secure communications via the Internet, which is an environment characterised by a low level of security compared to wireless networks. In addition, VPN architecture can support different authentication methods. One of its major drawbacks is the higher cost of equipment compared to other solutions. Another drawback is that the access points are very vulnerable, because they are installed in the DMZ, and therefore lack any protection that prevents attacks being carried out on

them. In this sense, it is possible to carry out very common attacks on them such as DoS attacks. An attacker could also use a false access point to acquire the VPN authentication information of any users and then use it later to enter the network. In addition, given the visibility of the wireless network, it is subject to the activity of *war driving*.

6.14.2.3 VPN policies

The use of VPN together with wireless networks requires appropriate security policies. As the wireless network is located outside the local network and is not connected directly to the Internet, it must be treated separately and properly in security policies. Many of the security policies used to protect networks from threats from the Internet can be used in this case.

A high-level policy should consider wireless as characterised by the same level of risk as the Internet. In reality, even if the wireless network is a very vulnerable network, due to the fact that the same emits electromagnetic waves that can be easily intercepted within the coverage area, the same is however subject to a smaller number of attacks because of the small number of attackers compared to the Internet.

Continuing in the comparison of risk between wireless networks and the Internet, some measures, such as dual factor authentication used for the Internet VPN, can be used securely even in wireless VPN.

In addition to this, there are many other security concepts applied to the reduction of risk from the Internet that can be applied to the reduction of risk originating from wireless.

6.14.3 Wireless gateway

Wireless gateways are another type of architecture that can be used. An example of this architecture is shown in Figure 6.50.

As can be seen in Figure 6.50, the wireless network is segmented and separated from the wired network in a manner very similar to the VPN architecture. This means that the wireless access points and the associated wireless clients are located in the DMZ outside the firewall. In this architecture, the firewall or gateway allows or denies access to wireless clients. This access may be adjusted differently in respect of the various devices that are located within the wired network. Some of these devices are able to select groups of users based on their credentials and to allow or deny access to different parts of the internal network.

The use of wireless gateway is characterised by merits and defects.

One of the biggest advantages is that the user authentication is practically supported by all the gateways and firewalls for wireless use as such devices are capable of managing lists of users to allow access to devices and network segments. They are also characterised by a high degree of flexibility in the management of security.

Some devices on the market allow the encryption of communications at the data link layer or at the MAC level, offering superior performance with respect to VPN as the latter requires the MAC or IP layer to send or receive packets. A VPN device may mask the IP level using only a communication between the gateways. The gateways are capable of encrypting the whole packet, leaving visible only the MAC address, as shown in Figure 6.51.

Certain gateway devices are able to operate authentication on three levels:

1. Login ID (access ID) that represents a password, which is located within the software of each client. This password is known only to the network administrator that configures the client and not the user. This password allows encryption to permit authentication together with the other two levels.
2. Device ID that is the key, which is loaded into the client device. This ID enables the administrator to avoid a client entering the network, when the same has been lost or stolen, rejecting its ID. This

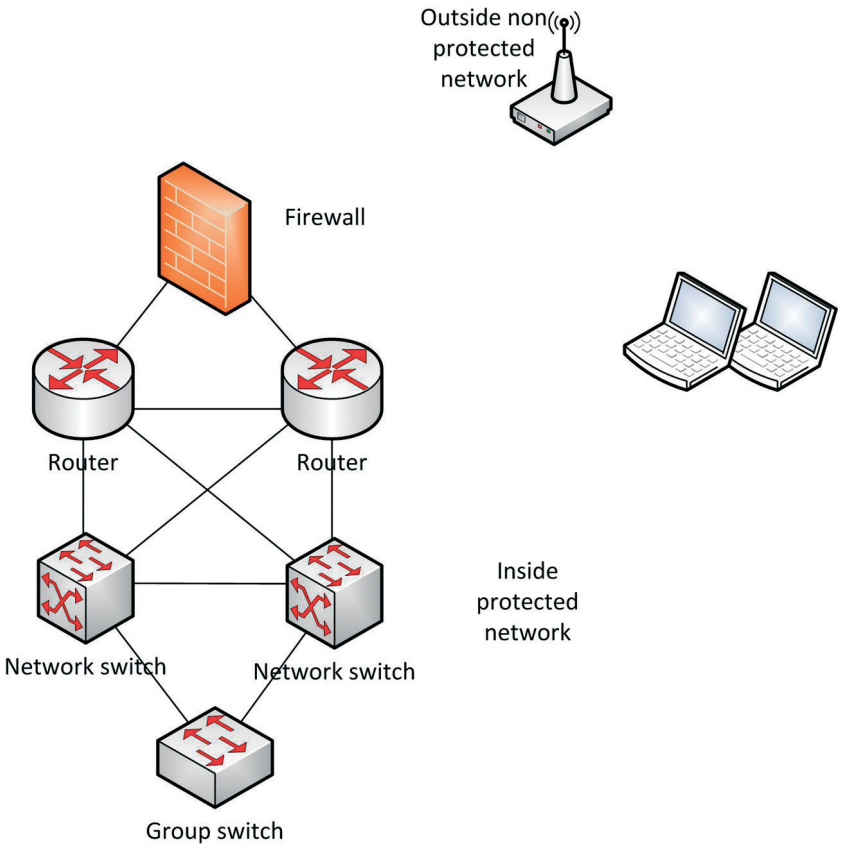


Figure 6.50 Example of gateway/firewall wireless architecture

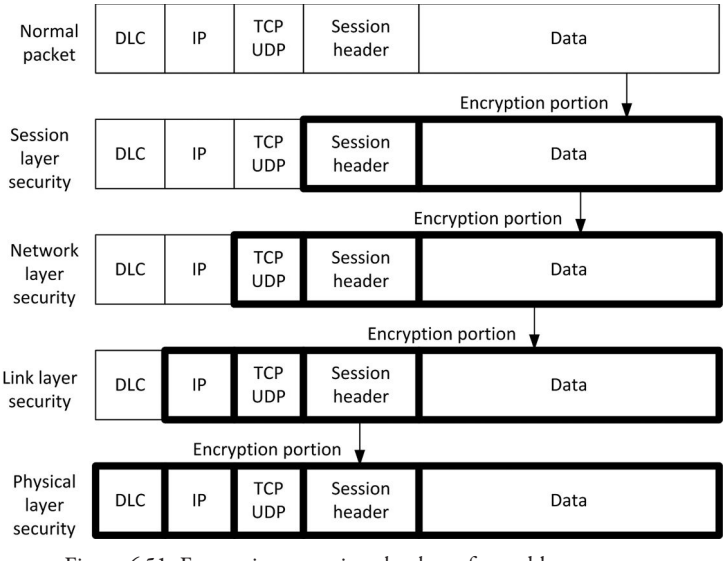


Figure 6.51 Encryption at various levels performed by a gateway.

mode is very useful for low-level devices that do not use authentication such as handheld scanners.
3. User ID that represents a password, which is required by the system to identify the user of a device. This ID supports identification systems such as RADIUS, and many others. This ID can be used with or without the ID of the device. When using the ID of the user together with the ID of the device, a high level of control can be reached because the administrator is able to limit the user's connection to the individual devices or allow a user to be able to use different devices.

Some gateway devices are characterised by a high degree of flexibility in the management of environments where there is a high variation of devices and users over time. These gateways use SSL for authentication of users and such authentication occurs on a Web page generated by the same gateway. This solution is relatively inexpensive to manage even if the same is characterised by increased vulnerability.

Some gateways are equipped with sensors that can communicate with a server to indicate the presence of false access points or of someone attempting to attack the wireless network.

Some gateways are able to ensure roaming from one subnet to another without any loss of communication and without requiring users to perform the authentication process again. This feature is very useful for the future, as multi-technology devices able to move from wireless technology to another without losing connection will certainly be marketed.

Some gateways are characterised by having an integrated anti-virus system, which greatly simplifies the management of security policy, having more features integrated within a single device.

One of the great disadvantages of such devices is that they are not able to protect wireless transmissions that occur in space; this means that the transmissions are exposed to risk until the same reach the gateway. The only security measure is the use of cryptography.

Another disadvantage of these solutions, which is thus the same as VPN solutions, is that the acquisition and management of devices with a considerable financial commitment is required.

6.14.4 802.1x

802.1x wireless architecture requires a different approach with respect to those in place for the previous architectures. The architecture in question refers to that provided by IEEE 802.11i as many parts of it refer to 802.1x. It has already been said that the 802.1x architecture ensures the security of the wireless network using an encrypted EAP channel to exchange the authentication traffic. Once the client is authenticated, the same can access the network; until authentication has occurred, the only traffic that is allowed with the network is authentication traffic.

This architecture is shown in Figure 6.52.

From Figure 6.52, it can be seen how the wireless network is not considered to be a subject of high risk, and for this reason it is not kept outside the main wired network. When using 802.1x, the only traffic that can pass without authentication within the network is authentication traffic. To support this architecture, three elements are necessary that serve to ensure the correct operation of 802.1x. They are:

1. applicant, represented by the wireless client;
2. authenticator, represented by the access point;
3. the authentication server, represented, in most cases, by a RADIUS server.

Once the key elements have been identified, the description of the mode of operation can be addressed.

When a client connects to the network, the access point asks him/her to authenticate using EAP. Depending on the EAP type used, different modes of authentication may take place. Regardless of the EAP type used, the successful authentication is declared to the access point with an appropriate

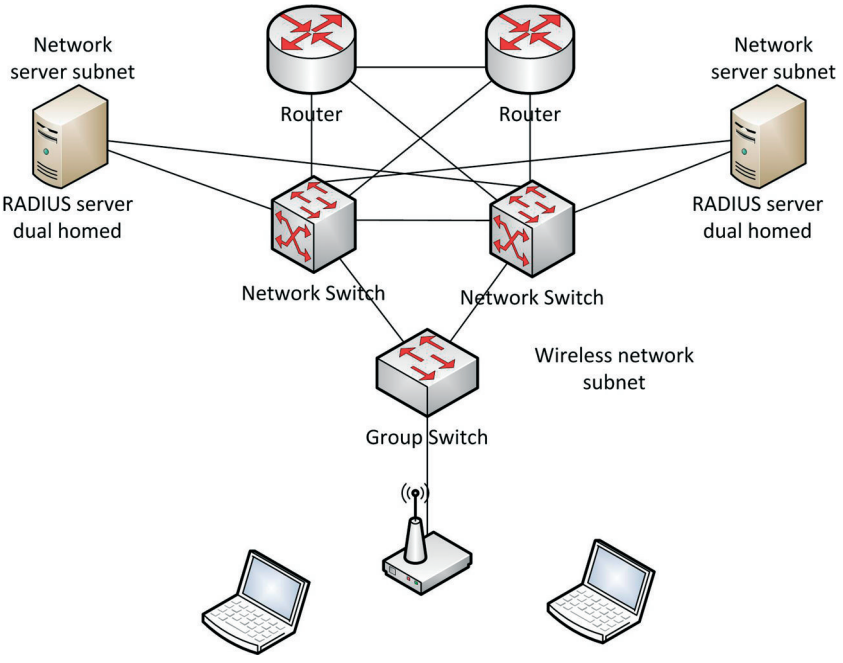


Figure 6.52 Example of 802.1x wireless architecture.

message. Once the authentication server sends this message to the wireless access point, the latter allows the wireless client access to the network.

One of the major benefits of using the solution in question is that having together the wireless network and the wired network, reducing the cost of purchasing various network devices and the consequent maintenance. Another great advantage is the ease in upgrading the network to be able to work with the 802.11i standard, an operation that would require a huge financial and time commitment where it was applied to one of the other architectures shown above. In this case, it is only necessary to change the configuration of access points and possibly the support on the clients in such a way as to allow the use of AES.

However, there are also disadvantages that are very similar to those of other architectures. In the first place, the network is visible in any of the activities of *war driving* even if to a lesser extent, because in the case of the other networks, the traffic can be seen in plaintext, whereas in the case of 802.1x, only management traffic is in plaintext. Given that there is traffic in plaintext, a man-in-the-middle or DoS-type attack is always possible. There is also the possibility of carrying out attacks against 802.1x, most of which were discussed earlier. A certain type of attack can occur if older RADIUS servers are used. In fact, when starting to use 802.1x and RADIUS in wireless networks, a number of questions concerning the dynamic creation of keys and the relevant exchange had not yet been defined. This meant that a number of older RADIUS servers used the same WEP key for all, which is not the case with the more recent RADIUS servers.

6.14.5 Comparison between the different wireless architectures

This section compares the wireless architectures just illustrated. Performance evaluation is a necessary activity for every organisation before being able to decide which architecture is more attuned to the company's needs and requirements.

Comparison will be made at a high level, highlighting the major critical issues, the associated risks and any hidden costs. Particular attention is paid to the security aspects, which are a fundamental requirement for wireless networks specifically, for the networks in general and for all communication systems.

6.14.5.1 WEP architecture

WEP architecture is very convenient, in terms of cost when used in relatively small networks. WEP is the most economical if the subsequent costs necessary to maintain it are disregarded, even if the activity of maintenance of this network is not particularly onerous. If the number of access points and the number of clients to be managed is not particularly high, then this network can become more convenient than other networks characterised by low management charges. The main source of risk is its vulnerability, which has already been mentioned above. If this network is the only one that operates in a controlled space, the risk is relatively low.

If this architecture is extended to medium-sized networks, then the overall cost is considerable, and the cost of the devices necessary to ensure a certain level of security is almost negligible compared to the overall cost of the network. In a medium-sized network, the number of access and client points can be relatively high, thus increasing the management costs to which must be added the costs necessary to ensure a certain level of security.

If such an architecture is extended to large networks, the relative cost of management becomes predominant, far surpassing the costs necessary to maintain a certain level of security. In this sense, WEP does not represent the optimal solution for large networks, apart from considerations relating to the security aspects.

6.14.5.2 VPN architecture

By repeating the analysis made for WEP for small, medium-sized and large networks, it can be said that VPN is not convenient for a small network, because the cost of a single VPN apparatus is far more than the cost of a few access point, which represents the wireless part of the entire network, making the security component economically predominant in relation to the entire cost of the wireless network. The maintenance that is required for this type of network can be greatly reduced with respect to WEP, even if each apparatus and each client requires initial implementation work that represents a relatively low-cost component.

From the point of view of security, depending on the type of encryption used, VPN is recommended when working in an environment in which it is necessary to ensure a high level of security.

In medium-sized networks, characterised by multiple sites, the main choice that must be made concerns the use of local VPN concentrators or a single remote concentrator, whose merits and defects have already been described. Once it has been determined how to position the VPN concentrator, the cost of support can be analysed, to understand the cost necessary to train all users in the use of VPN technology and the cost needed to create a help desk to which users can turn in case of difficulty or to ask for general information. These costs can be relatively low if the organisation already uses VPN technologies, perhaps on a wired network, but may be relatively high if the organisation faces the installation of a VPN technology for the first time.

One of the greatest risks is manifested when the network is installed and operational. In this phase, many users may be tempted to circumvent VPN to avoid all the operations of access necessary to ensure a certain level of security, perhaps by installing other unauthorised access points. In fact, it must be remembered that in VPN, the access points must be installed in the DMZ, failure to do so possibly compromising the entire security of the system. In this sense, there are additional costs required to verify that the network always remains as it was established at the initial stage.

In large networks, the total cost of the security component is negligible in comparison with the total cost of the wireless network, making VPN a very advantageous and secure choice. In this case, the use of VPN allows the use of different authentication technologies (such as badges, certificates and passwords) that permit an increase in the overall level of security of the wireless network.

6.14.5.3 Firewall architecture or wireless gateway

With regard to this type of architecture, many of the considerations made for VPN apply.

If this solution is being used in small networks, the cost of administration may be less than architectures based on WEP or VPN. If a higher level of security is required, and thus the additional software on clients, then the cost starts to resemble that of VPN and approaching that of WEP solutions. To assess the applicability of this solution in small networks, it is necessary to take into consideration two major factors: cost and security. It should be remembered that the cost of the hardware is still an ever-present element unlike the cost of security that can be optional. The cost of security directly depends on the level of security to be ensured: a greater level of security of course means a higher cost.

If using this solution in medium-sized or large networks, the costs are inevitably increased because of the greater number of apparatuses that must be installed. Another cost is the support need to continually check that the access points are installed within the DMZ; in large networks, scattered across multiple sites, this activity can become very onerous and therefore expensive.

6.14.5.4 802.1X Architecture

The architecture in question can be used, without any major problems, in a small network if there is a RADIUS server and a Windows-based network. If these elements are already present, then the installation of 802.1x requires only a change of configuration on the server and access points. If this is not the case, then the installation of a RADIUS server can represent a significant cost. However, it is always possible to use only one centralised RADIUS server to reduce costs compared with the cost needed to purchase and install other devices and to ensure a support service.

In the case of large and medium-sized networks, it is possible to check to see if an MS RADIUS solution is well adapted to these situations. In most large- and medium-sized networks, there is already a presence of MS products along with domain controllers that process the network authentication. These controllers can be used to perform the small work load required by RADIUS, a load that is very small compared to other services such as DHCP or DNS. If there are already controllers that can be used, then this solution is the cheapest of all those seen so far.

6.15 Wireless tools

We have seen that wireless networks are accompanied by significant risks for those using them as the electromagnetic signal propagates into space and can be easily intercepted by an attacker in possession of all the necessary technical skills as opposed to wired networks, whose signal interception that passes within is more difficult, this activity requiring direct physical access to network cables. In wireless networks, on the contrary, such interception may also occur outside of the site of interest through the use of appropriate directional antennae capable of increasing the two-way radio capacity of access points along certain preferential directions.

Electromagnetic signals, in wireless networks, are within the reach of anyone has produced a proliferation of a certain number of tools with positive features, such as the network management or troubleshooting tools, and tools with negative features, such as tools of interception and network

attack. However, even those tools with negative connotation can be widely used to find and analyse the vulnerabilities of networks.

These instruments are characterised by a small series of features such as scanning, sniffing, cracking and the possibility of generating DoS attacks.

In the following, these tools are discussed.

6.15.1 Scanning tools

Scanning tools are easy to use sniffers that intercept network traffic and provide on-screen all the information that they are able to intercept. They allow the type of network and the relevant network configuration to be shown, features that are very interesting for the activities of *war driving*. The main information that such scanners are able to provide are as follows:

1. channels used;
2. type of security;
3. MAC address;
4. access point manufacturer;
5. transmission speed;
6. signal intensity;
7. noise level;
8. GPS localisation.

These scanners are used for the identification of wireless networks and are used by *war drivers* to find networks and then proceed with an activity of *sniffing*.

6.15.2 Sniffing tools

Sniffing tools allow users to view data packets of wireless networks and therefore network traffic.

Sniffers are of great help in troubleshooting network issues but can be extremely dangerous if used with negative purposes. In fact, since they can provide information about the wireless network, the SSID, the channel, the power and many other points of information concerning the type of security used, a possible attacker may use such information to attack the network itself. In addition, *sniffers* also provide information related to IP traffic, allowing indirect mapping of the network, permitting the locating of servers and other devices that contain confidential information. In many cases, a sniffer allows the acquisition of valuable information to conduct a man-in-the-middle attack and to circumvent the security measures implemented in the network.

Furthermore, network *sniffers* allow information concerning the security methods used to be obtained, to be able to find older and therefore weaker ones.

Very often there are protocols that use plaintext authentication such as Telnet, which has already been discussed above. If an attacker is able to use a sniffer, the same is able to intercept all traffic relating to authentication such as username and password.

Another critical aspect of security is the use of email as POP3 uses plaintext authentication. In this sense, particular attention must be given while using email outside the corporate site, as authentication credentials of our email account can be easily intercepted and used fraudulently.

6.15.3 Hybrid tools

There are also tools that can behave like scanners, *sniffers* and *crackers* (which will be illustrated in the following) at the same time. These tools are called hybrids, precisely because of their ability to perform several functions within the same tool.

6.15.4 DoS tools

DoS tools are used to conduct a denial-of-service attack against wireless networks. The greatest risk for wireless networks is jamming, which directly interferes with the electromagnetic signal, but, in such networks, it is also possible to carry out an attack against management logics, generating a DoS-type attack.

Some of the tools, to conduct a Dos attack, send messages of disassociation to force the wireless clients to exit the network and to re-enter, repeating the authentication operation. During re-authentication, the attackers have a greater chance of intercepting the authentication traffic or of producing a man-in-the-middle attack.

6.15.5 Cracking tools

Cracking tools are used to violate the type of cipher used in wireless networks. Most of the standards for wireless networks use strong ciphers. In many cases, however, the way in which these ciphers are used has vulnerabilities that can be exploited by attackers to breach the security of the network.

6.15.6 Access points attack tools

The tools in this category are intended to directly attack the access points. They exploit the vulnerability of the protocols used from access points such as SNMP, HTTP and Telnet.

6.15.7 Security tools

Security tools do not belong to any of the categories of tools outlined above. They are able, for example, to circumvent the filters based on MAC or gateways based on SSL.

This page intentionally left blank

CHAPTER 7

VOICE SECURITY

7.1 Introduction

Voice communications take place, for example, when speaking normally on the phone. The structure of the system of communication by telephone has already been addressed in Chapter 1.

It represents a system that is extremely vulnerable to interception, especially the section of the so-called "last mile", in which the signal normally travels in analogue and clear form over a simple twisted pair whose path is, in most cases, easily accessible and can allow easy interception of signals that travel along it.

In this sense, if we want to communicate vocally in a secure manner, in order to guard against interception, recourse to appropriate techniques that make communication secure is required.

In most cases, use is made of encryption techniques, both analogue and digital, which make the signal incomprehensible to a possible interceptor that, if not in possession of the right encryption/decryption apparatuses, such as those used by interlocutors, is not able to comprehend anything.

The same techniques can be used for transmissions via radio, which, by their nature, propagate everywhere and are easily receivable and can be intercepted.

7.2 Characteristics of the spoken language

Before addressing the techniques that make voice communications secure, it is necessary to know about the characteristics of the spoken language that represents the object to be made secure.

The human voice can be represented by a series of sounds called phonemes. A phoneme is the smallest part of speech that differentiates words spoken vocally.

These phonemes are emitted by the so-called voice organs that are shown in Figure 7.1.

All the various sounds of the voice derive their strength from the respiratory system that pushes air out due to the lungs. This flow of forced air passes through the vocal cords that are located within the throat. The vocal cords are characterised by an opening which can be opened or closed as desired to vary the sound that they generate. The part included between the vocal cords and the lips is called the vocal tract. The shape of the vocal tract can be varied as required to produce the different sounds. The

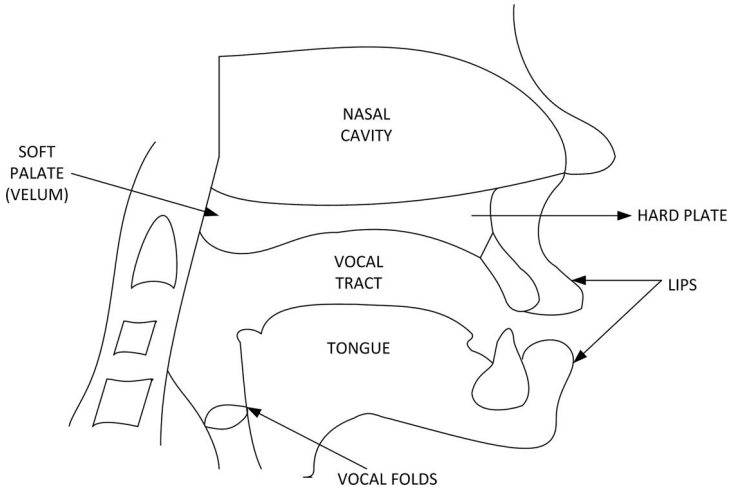


Figure 7.1 Sectional view of the voice organs.

vocal tract is composed of a series of elements, such as the tongue, the lips and jaw, which are called articulators.

The vocal tract and the nasal cavity, characterised by uneven sections, act as tubes resonating at various frequencies, which resonate, possibly, at the frequencies produced by the vocal cords. Resonance frequencies are called formant frequency or simply formants. The formant frequencies of each particular phoneme vary from person to person as they depend on the shape and size of the vocal tract.

The basic generation apparatus of the voice is able to produce different sounds according to the shape of the vocal tract, which can be changed appropriately during the action of speech, and by the excitation source (as will become clearer later).

Each spoken language is characterised by a certain number of phonemes capable of producing the entire language. In the case of the English language, there are 40 different phonemes that can be divided into three groups:

1. voiced;
2. unvoiced;
3. plosive.

When we breathe normally, the vocal cords remain open, to allow the normal passage of air. When we close the vocal cords and exert a certain amount of pressure on them through the air pushed from the lungs, the cords are opened, pressure decreases, the cords are closed and so on, generating a cyclic and vibratory movement. This vibratory cycle persists as long as there is still pressure exerted by the air emitted from the lungs and this makes it possible to produce characteristic sounds (such as the "a" and the "e") that resonate in the vocal tract. These noises are referred to as voiced, and the vibration frequency of the vocal cords is called the pitch frequency. If the length, thickness and tension of the vocal cords are varied, the relative frequency of oscillation also varies. In general, pitch frequency is higher in females and children as compared to adult males: the frequency range of the latter is 100 to 400 Hz while in women and in children it can reach 3,000 Hz and higher.

The unvoiced sounds are generated by loosening the tension of the vocal cords in such a way that air flows freely, preventing oscillation, and partially obstructing the air flow with the upper articulators. In this way, it is possible to make sounds such as "ss".

The plosive sounds are similar to voiced sounds but in this case, the air flow is temporarily closed using the articulators and subsequently all are opened together as in the sounds "p", "v", etc.

7.2.1 The structure of language

In order to acquire all the relevant information pertaining to the science of the voice, there follows a brief history of language. Anthropologists, from studying languages, have found many similarities between each of them and have discovered that they originate from certain principle languages.

Until 1950, language scholars believed that the study thereof should be approached using the following steps. The first step is phonology, which is the study of the sounds contained in a language. Phones are linguistic sounds that have a very specific meaning in each language. They were recorded using the International Phonetic Alphabet, represented by a series of symbols that are used to describe the various sounds in different languages. The second step is morphology, which is the study of the ways in which language is combined to generate the morphemes-words and their constituent meaning. A morpheme is the smallest unit of speech that has a meaning. Language agglutination combines several morphemes. Inflection languages change the shape of words to emphasise certain grammatical differences such as gender. Indo-European languages are highly inflectional. The next step is syntax, which is the study of the organisation and sequence of words in sentences.

7.2.2 Phonemes and phones

No language in the world contains all the sounds in the International Phonetic Alphabet. Phonemes have no meaning individually, but assume a meaning when combined together.

Standard English comprises 35 phonemes including 11 vowels and 24 consonants. The number of phonemes varies from language to language. There are between 15 and 60 with an average of 30 to 40 phonemes per language. The number of phonemes varies with dialect. Phonetics studies the contrast of sounds in languages. In every language, a given phoneme extends over a so-called phonetic range.

7.3 Voice configuration

Before moving on to illustrating the techniques for secure voice communication, it is necessary to illustrate how it is possible to model the human voice.

7.3.1 The classic source–filter model

The classic source–filter model is widely used for voice generation. This model, shown in Figure 7.2, configures the vocal tract like a filter, generally linear, with variable characteristics over time.

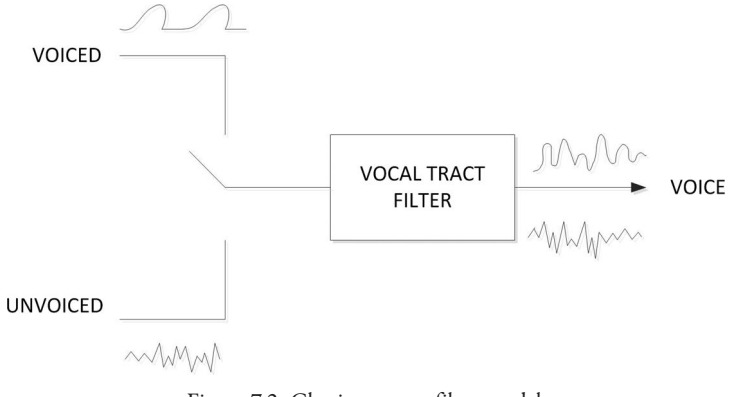


Figure 7.2 Classic source–filter model.

The energy source for such a filter is the excitation signal. The source–filter model regards the excitation and the vocal tract as two separate entities in voice generation. The excitation is produced at a point in the vocal tract and subsequently, the same is treated spectrally by the rest of the vocal tract.

The throat, nose, tongue and mouth form resonant cavities filled with air that affect decisively the sound produced by the human voice system. It has already been stated that the resonant frequencies of these cavities are called formant frequencies. Different configurations of the vocal tract lead to different formant frequencies. Formant frequencies are one of the two main factors that determine which phoneme the vocal tract must generate. The other key factor is the excitation of the vocal tract.

In terms of voiced sound, a periodic wave form generates the excitation of the vocal tract. The periodic wave is generated by the rapid opening and closing of the vocal cords.

Unvoiced speech is generated by using white noise that is random and has a flat spectral form, with energy evenly distributed over all frequencies. White noise is generated by means of restricted air flow.

There are certain sounds that are generated via excitation of the vocal tract using periodic excitation and by forcing air through a narrowing in the vocal tract. This mode is called mixed excitation. One of the major problems with voice encoding is the way in which to represent voiced, unvoiced and mixed sounds.

7.3.2 The general source–filter model

In addition to the classic source filter model, there is a general source–filter model, which is illustrated in Figure 7.3.

As can be seen from Figure 7.3, the pitch data are usually represented by the value of the pitch period that varies over time. Depending on the pitch period, the periodic excitation block generates a pulse wave form that represents the impulse generated by the vocal cords. The noise excitation block generates a loud sequence with a flat spectral response. The two excitations are introduced into a mixed-resolution block. The variable time data for the sound of the voice are the other input in the mix-decision block. Depending on the sound level of the original voice, the mix-decision block combines the periodic excitations and loud excitations in an appropriate manner to produce the excitation signal.

The classic version of the filter consists of two states in which there are two types of excitations and the decision mix mixing is taken from the switch that decides if the voice is voiced or unvoiced, depending on the classification. The data relative to the vocal tract are supplied to the vocal tract block to generate a vocal tract filter. The filter shapes the excitation spectrum to make it resemble the original voice. In practice, the vocal tract data can be represented by numerous methods, including the so-called linear predictors or the Fourier transforms. The purpose of the vocal tract model is to shape the

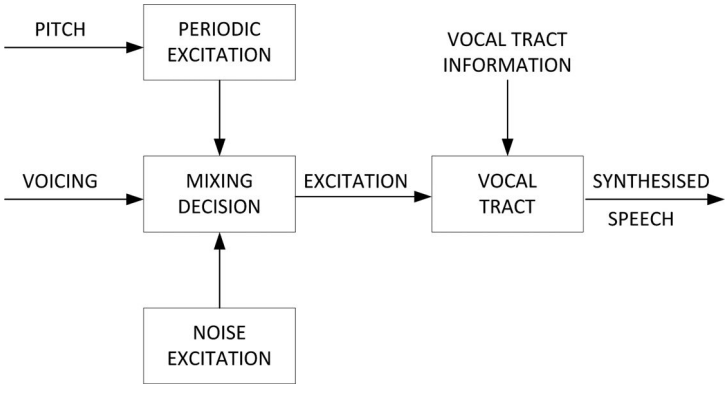


Figure 7.3 General source–filter model.

temporal flow of the signal produced, to make it sound as close as possible to the sound of the original human voice.

The excitation source for the voice sound is characterised by a combined spectrum that decreases to 12 dB/octave (one-octave is the interval between one frequency and another, the frequency of which is double). It is sent to the vocal tract filter that has an almost flat frequency response, generating a voice output response of −12 dB/octave. The voice then passes through the lips that function as a differentiator characterised by 6 dB/octave, generating an overall speech spectrum assembly that decreases to 6dB/octave. Since voice production generates a mechanical movement of the articulators in a specific period, the voice signal changes relatively slowly with time. Thus, the voice is defined as a quasi-periodic phenomenon and is almost static, being periodic for only a short period of time and random over longer periods of time.

The voice signal may be illustrated by using the so-called spectrograph, capable of generating a spectrogram that represents a three-dimensional representation of the voice, where time is the horizontal axis, frequency the vertical axis and the intensity of the resulting image proportional to the energy of the signal. The spectrograph is a very important tool for the understanding and analysis of the voice. Figure 7.4 shows an example of a spectrogram.

7.3.3 Linear prediction modelling

Linear prediction modelling (LPM) is widely used to show the vocal tract frequency behaviour of the source–filter model.

Linear prediction (LP) analysis is used in voice encoding and serves to characterise the shape of a spectrum of a short-time segment of the voice, using a small number of encoding parameters. Linear Predictive Coding (LPC) predicts a voice sample in the time domain based on a weighted linear combination of previous samples. LP analysis is a method to eliminate redundancy in a short-term correlation of close samples. The formulation of LP takes place by using differential equations that define a tube without loss that models the vocal tract.

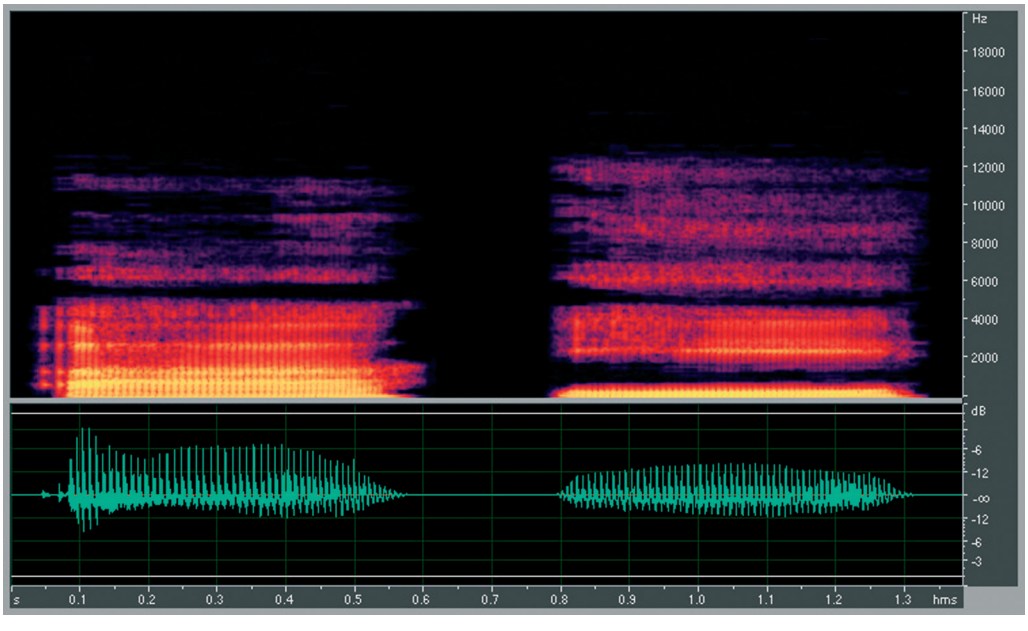


Figure 7.4 Spectrograms of the spoken sounds “a” and “i” pronounced by a native Italian speaker (top) and the related wave forms (bottom).

Sound waves are variations in pressure that propagate in the air through oscillations of the particles that make up air. The modelling of these waves and their propagation in the vocal tract provides a tool for illustrating how the vocal tract shapes the frequencies contained in the excitation signal. A model is widely used to treat the vocal tract like a uniform tube without loss with constant section. The variation in pressure, the speed of volume over time and the position is described by the following pair of differential equations:

$$-\frac{\partial p}{\partial x} = \frac{\rho \partial v}{A \partial t} \tag{7.1}$$

$$-\frac{\partial v}{\partial x} = \frac{A}{\rho c^2}\frac{\partial p}{\partial t} \tag{7.2}$$

X being the position of the tube, t the time, $p(x,t)$ the sound pressure at position x at time t, $v(x,t)$ the speed of the volume at position x at time t, ρ the air density within the tube, c the speed of sound and A (x,T) the area of the section of the pipe at position x at time t.

A simple model of the vocal tract has the same characteristics as a simple electric circuit. In fact, if we compare the wave equations of a tube without losses and those of the current $I(x,t)$ and voltage $V(x,t)$ in a uniform transmission line and without losses that are explained below:

$$-\frac{\partial V}{\partial x} = L\frac{\partial I}{\partial t} \tag{7.3}$$

$$-\frac{\partial I}{\partial x} = C\frac{\partial V}{\partial t} \tag{7.4}$$

where L represents the inductance and C the electric capacity, we can see how the four equations are similar in pairs to the correspondence of the quantities and of the acoustic and electric parameters.

7.4 The transmission of voice signals

Voice signals are currently always processed, quantised, digitised and codified to be transmitted and encrypted, over various types of channels, both in analogue and digital form.

With analogue transmissions, the message varies continuously, as has already been seen, while with digital transmissions, the signal can vary only over discrete values.

Regardless of the fact that a signal is analogue, there is, in the transmission system, a process for converting analogue to digital (A/D) and for converting digital to analogue (D/A). The two systems for transmitting analogue and digital are shown in Figure 7.5.

One of the most important elements for a successful encrypted voice transmission is synchronisation. A receiver is able to decipher the voice stream accurately only if it is properly synchronised. Subsequently, the system must provide for the use of an appropriate synchronised signal so that the receiver can synchronise with the transmitter with accuracy, and to be able to decipher the encrypted voice stream. Given its importance for the functioning of the encryption/decryption system, it is very important that the synchronised signal is not easy to identify.

There are two types of synchronisation schemes:

1. initial synchronisation;
2. continuous synchronisation.

With initial synchronisation, data relating to the synchronisation are transmitted at the beginning of the transmission after which they are used as a suitable timer to maintain synchronisation. The main disadvantage of this system is the fact that if the receiver is not able to pick up the initial synchronised signal, the entire message is lost, since it is no longer possible to synchronise the receiver with the

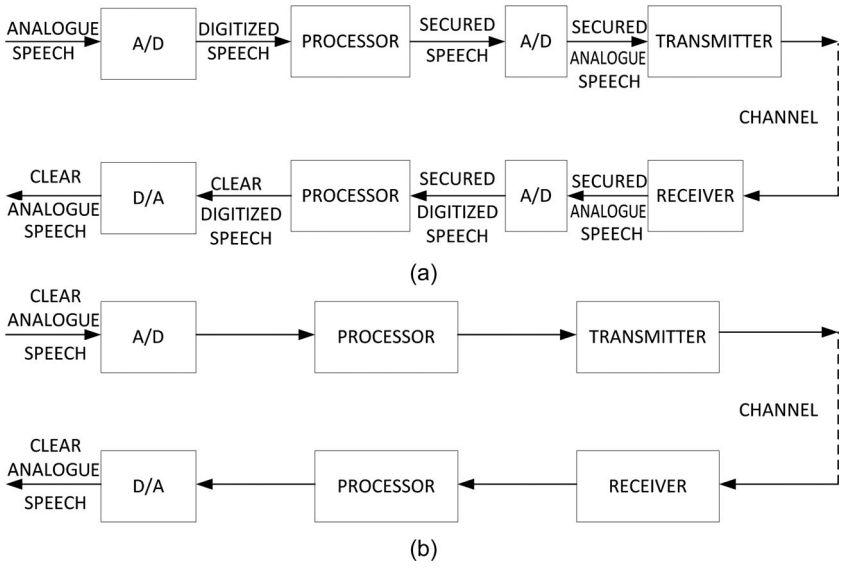


Figure 7.5 Analogue encryption systems, with (a) digital processing and (b) fully digital encryption systems.

transmitter. Therefore, the synchronised signal must be very robust with respect to the signal that is used for continuous synchronisation, as the receiver can pick up this signal, without uncertainty, only at the initial phase of the communication. For this reason, the above system must be used exclusively on channels with particular standards.

With continuous synchronisation, the encrypted signal is constantly interrupted to transmit the synchronised data. Every synchronised signal contains a new message key that, usually, is a function of the previous key. In this way, if the receiver fails to correctly receive the synchronised signal, it can always generate the anticipated message key and continue with the deciphering. The advantage of continuous synchronisation is the fact that if the receiver loses certain parts of the message, it can start the decryption as soon as it receives the synchronised signal.

The synchronised signal can be simply made up of two tones at different frequencies that switch with each other to gather the temporal data. Alternatively, the temporal data can be obtained by transmitting predefined sequences of data at the time of the update, in a way that the receiver can recognise and acquire the temporal data. Obviously, the data sequences must select both tones in a way that they are not present in the voice stream, to avoid the receiver picking them up and causing synchronisation errors.

The synchronised signal can be transmitted in many different ways. If we are using initial synchronisation, obviously the synchronised bits are transmitted at the beginning of the message. The full band can be used to transmit the synchronised bits using digital modulation techniques such as frequency shift keying (FSK) and phase shift keying (PSK). We have already seen that with these modulation techniques the different frequencies, or the different phases, are used for the binary values 1 and 0.

Continuous synchronisation systems can send both the synchronised data on a separate channel, increasing the required band, and on the same channel as the encrypted voice, using multiplexing techniques in frequency or time. The part of the spectrum where the energy of the voice is generally lower is in the 1,800 to 2,000 Hz band: in the 100 to 200 Hz range, the synchronised data are sent using digital techniques previously illustrated. To do this, we must use very selective filters, generally the notch type, so as to be able to delete the voice from the band, appropriate to the integration of the

synchronised signals. Unfortunately, this mode of communication is susceptible to disturbances, voluntary or involuntary, which may occur by using high-powered signals in this band.

In many cases, multiplex techniques are used in the time domain, in which the voice is suppressed for a relatively short time interval, within which the synchronised signals are being sent. This removal must be conducted over a relatively short period to make the voice comprehensible, but it is an excellent system against determined hacker attacks.

7.5 Voice signal encryption

There are numerous techniques that can be used to encrypt voice signals. As already said, the human voice is a predictable signal and temporally slow to vary. Specific sound configurations of the various phonemes are responsible for the different sounds that we hear. To make voice secure and therefore incomprehensible to interceptors, its structure should be as unpredictable as possible, rendering it unintelligible. At least, to achieve the highest level of security, voice should be transformed so that the so-called residual intelligibility, that is the level of understanding of the encrypted signal, is equal to zero. At the same time, the listener should be able to decipher the encrypted voice signal by simply applying inverse transformation. The encryption/decryption system should be very robust to sound, which is an inevitable factor and always present, at various levels, in all processes, channels and methods of communication. This system should also be free from distortion and should provide an output signal whose band is preferably the same as the voice signal band in plain text, so as to be able to use the same communication channel. Besides, this system should not result in excessive processing delays, which would make very difficult, if not impossible, the reciprocal concurrent communication process, typical of a vocal conversation between two people.

A system or a device used for voice encryption is also called, in the English language, *scrambler*. Voice encryption techniques may be divided into analogue techniques and digital techniques. Analogue techniques are illustrated for cultural and historic completeness only, as they are practically no longer in use.

In relation to analogue techniques, they can be further divided into techniques:

1. in the frequency domain;
2. by transformation;
3. in the time domain;
4. in the time and frequency domain.

Analogue *scramblers*, in general, do not obscure or remove any data from the speech signal: they simply rework the data contained in the signal to create a one-to-one correspondence with the voice signal in plain text. As such, these *scramblers* are very similar in operation to transposition ciphers where blocks of text, or individual characters, are permuted. Analogue *scramblers* are also classified as narrow band systems because they can be used over standard telephone channels.

Digital *scramblers* are used to encrypt voice using pseudo-random sequence, as normally occurs in cryptographic processes, as illustrated previously. In most cases, stream ciphers are used, as the voice signal can be a continuous stream of digital data if it has been previously quantised and digitised. Depending on the digitisation technique used, digital encryption systems can be broad band, if we encode the shape of the wave, or narrow band, if we encode the source, as will be shown below, using modelling concepts for voice sources, as illustrated previously.

The various analogue and digital techniques are illustrated in the following sections, subdividing them appropriately according to the above method, for the sake of clarity.

7.5.1 Voice signal analogue encryption

It has already been shown that the encryption of analogue voice signals uses different techniques that can be divided into:

1. encryption in the frequency domain;
2. encryption by transformation;
3. encryption in the time domain;
4. encryption in the time and frequency domain.

These techniques will be illustrated in the following.

7.5.1.1 Encryption in the frequency domain

With regard to encryption in the frequency domain, there are different techniques that can be used, which are explained below.

The simplest basic technique is the reversal of the band that is, simply, reversing the signal band so that the higher frequencies are moved to the lower frequency bands and vice versa. With this technique, since most of the voice signal is included in a band not exceeding 3,000 Hz, a higher frequency than this band is considered, for example 3,400 Hz, and a carrier is modulated in amplitude at this frequency in such a way as to have a lateral higher band, whose spectrum is equal to that of the original voice signal, and a lateral lower band, whose spectrum is inverted compared to the original signal. If we pass this signal through a low filter with a cutoff frequency of less than or equal to that of the modulated signal (3,400 Hz), the carrier in the upper lateral band is deleted, obtaining as a result a voice signal with the spectral flow reversed with respect to the original voice signal, resulting in what is called band inversion. This situation is illustrated in Figure 7.6.

To ensure that only a negligible part of the voice signal is outside of the signal band, the inverted frequency varies randomly within a small range of 3 kHz. According to the rate of change of the carrier frequency, there may be a slow random inversion (less than 5 hops per second) or a fast variation (more than 5 hops per second). With this method, it is possible that during the various hops, a part of the voice signal spectrum exceeding 3 kHz is deleted, but this does result in a significant deterioration of the voice signal, always leaving it quite comprehensible. The great disadvantage of this technique is the fact that within the inverted signal, it is always possible to recognise a similar rhythm to that of speech,

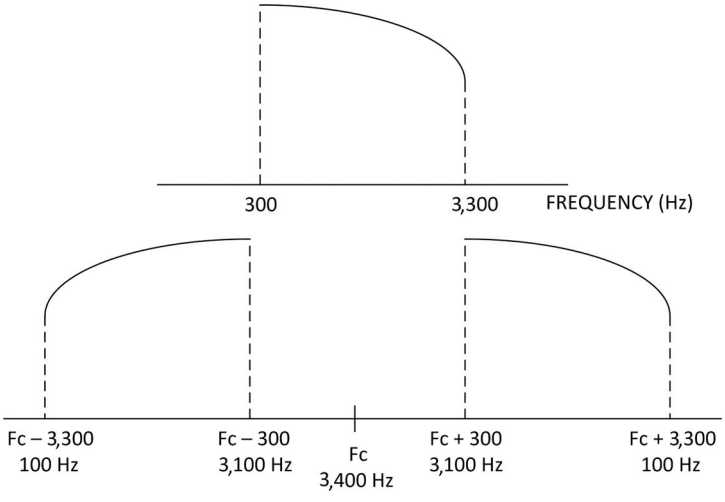


Figure 7.6 An example of band inversion: (a) original signal; (b) modulated carrier and lateral bands.

leaving a high residual intelligibility, further reinforced by the fact that the central band frequencies (1.5 kHz) remain virtually unchanged after the inverted band operation.

Another technique that is widely used is the division of the band into smaller bands that are in turn moved to other positions, inverting any individual bands, or combining the switching operation of the lower band and the inversion of the same. The aim is to disturb as much as possible the spectral structure of the signal and to create discontinuities.

The simplest method is by splitting the spectrum of the original signal into two sub-bands, choosing a suitable frequency at which to operate the division. This situation is illustrated in Figure 7.7.

A triangle is used to arrange the voice signal band as it has higher energy at lower frequencies and a lower energy at higher frequencies. This spectrum is divided around a central frequency, after which it is possible to select three different options:

1. to exchange with each other the sub-bands (band shift);
2. to invert one or both of the bands (band inversion);
3. to combine together the shift and the inversion (band shift – inversion).

Band shift involves a transfer of the low-frequency band in place of the high-frequency band, and vice versa, as shown in Figure 7.7(b).

With regard to band inversion, one or both of the bands may be inverted, as shown in Figure 7.7(c). This approach will not be particularly effective in the suppression of residual intelligibility. As such, it is possible to use the frequency hopping band division.

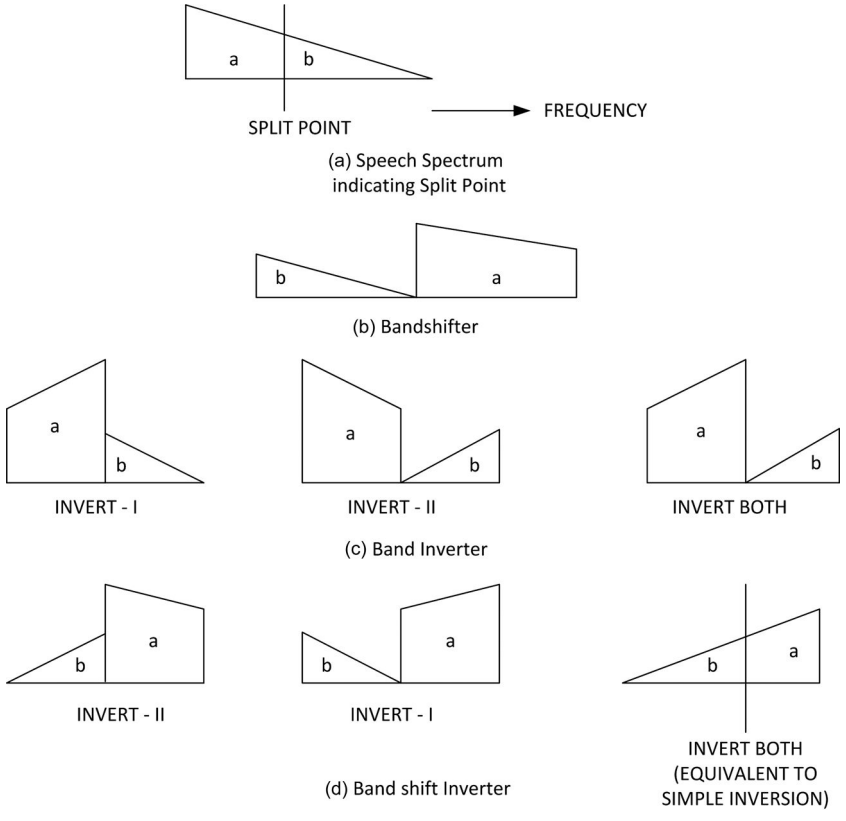


Figure 7.7 Different methods of band shifting and inversion. (a) Speech Spectrum. (b) Band shift. (c) Various modes of band inversion. (d) Various modes of inversion and band shifting.

In terms of band shift–inversion, this combines both techniques together, as illustrated in Figure 7.7(d).

In all the cases referred to above, frequency hopping is used with band division/inversion, trying to jump in frequency at least every 10 to 20 ms. A faster hop speed is not very useful in the reduction of residual intelligibility as happens in the human brain, above these speeds, to compensate for and to try to make sense of the encrypted voice signal. In addition, a very fast hop speed leads to excessive distortion after decryption, generating deterioration in the quality of the voice signal.

The division of two bands can be generalised by dividing the original voice signal band into multiple sub-bands, whose number is greater than 2, and then by combining these to obtain the encrypted voice signal. A diagram of this system, which shows a division into four sub-bands, is illustrated in Figure 7.8.

As can be seen from the diagram in Figure 7.8, each sub-band is modulated in the desired position according to the encryption code. The most complex part of this system is the filter bank that is required for dividing the spectrum into sub-bands. Sub-bands can be selected to be the same extension, or a different extension, overlapping or non-overlapping. For the non-overlapping band, very clear filters must be used. In some cases, when different bands must be used, narrow bands are selected for low frequencies and broad bands for high frequencies in a way that will disturb more the low frequencies where the data content is greater.

In many cases, the voice signal is divided into relatively short time frames, and the same is subject to, frame by frame, the process of division into sub-bands and appropriate mixing.

If, for example, four sub-bands are as in the diagram in Figure 7.8, then the number of possible permutations that can be used is $4! = 24$, according to the variable sequences, to the frames that make up the voice signal. It is clear that some permutations will be better than others from the point of view

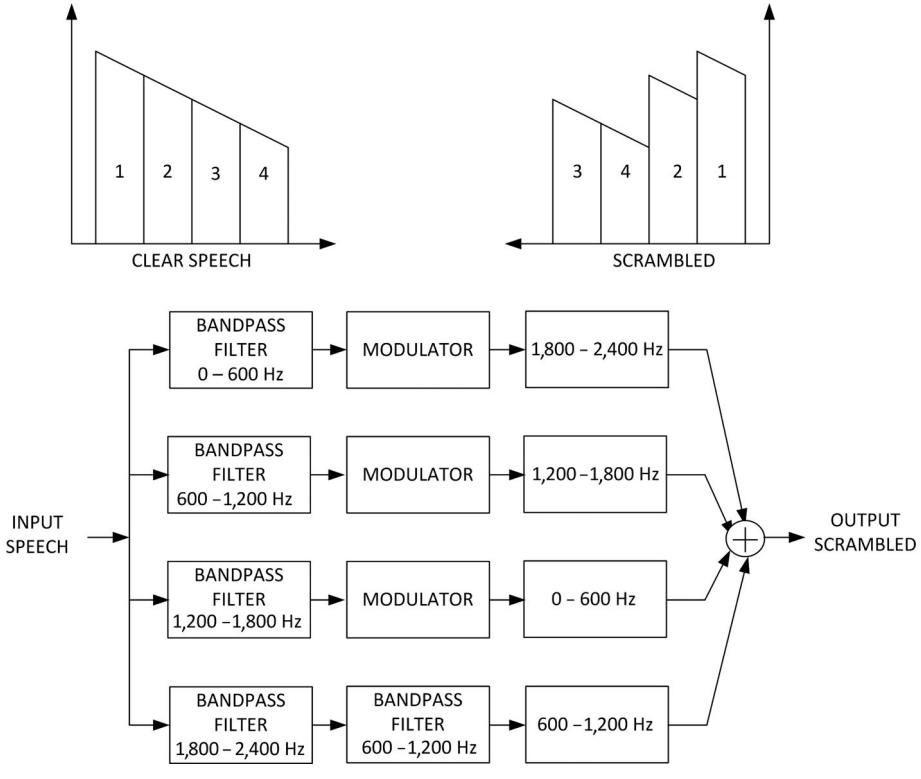


Figure 7.8 A system diagram of division into four sub-bands.

of reduced residual intelligibility. In particular, we have to remember that the lateral bands left nearby after the encryption process improve residual intelligibility, and that the bands moved away as far as possible from the encryption process reduce residual intelligibility.

Therefore, the shift factor (SF) is defined as the average difference in absolute value between the original location of each band and the relative position after the change. To obtain the best values in terms of reduced residual intelligibility, this value should be as high as possible. For example, consider the initial sequence (1 2 3 4) and the altered sequence (4 1 2 3). The SF is equal to $(|1-4|+|2-1|+|3-2|+|4-3|)/4 = 1.5$.

The properties of the human ear and brain use the verbosity of the voice to take advantage and give meaning to an incomplete conversation, combined with the fact that 40% of the energy of the voice is in the 0 to 1,000 Hz range, making the choice of permutations very important. A good permutation for a division of bands should have a high SF because ideally the shifted bands should be moved as far as possible from the original location where they do not remain adjacent, if they were in the original signal.

This ensures the best permutations are recorded onto the device memory from which they are retrieved, depending on the periodic pseudo-random sequences. These permutations should be changed as quickly as possible but if the permutation sequence is changed too quickly, inevitable distortions are introduced. Furthermore, synchronisation between the transmitter and receiver is very important and becomes particularly critical with the increase in the rate of change of the code and as such, this speed must not exceed five changes per second.

Another possible variation of this system is the use of a large number of sub-bands: the greater the number of sub-bands, the greater the margins of permutation, although the greater the complexity of the filter banks, the greater the need to divide the voice signal into various sub-bands. Subsequently, the group velocity also becomes an important factor, that is the different speeds with which the various frequencies propagate over the channel.

In many cases, the band divider becomes more complex when randomly inverting some of the sub-bands, making the choice of the permutation codes more difficult. Therefore, a code with which certain bands do not have to be moved far from their original position, is better if some of the sub-bands are also inverted randomly. Similarly, if two bands are adjacent to both the original signal and the modified signal, if applied to one of the two inverted bands, residual intelligibility is considerably reduced.

The N band dividers were widely used until the advent of digital technology, even if found to be quite expensive.

7.5.1.2 Encryption by transformation

Encryption by transformation or transform-based scrambling (TBS) is based on linear transformations that are performed on the voice signal. In this situation, frames consisting of N samples are transformed into a vector with N components. Encryption, *or scrambling*, is applied by changing these elements before applying the inverse transformation to obtain the encrypted signal in the time domain. With the receiver, inverse transformation is first applied, as was applied to the transmitter, and subsequently inverse permutation is applied, as was also applied to the transmitter, to restore the original order of the voice samples.

Consider the vector V composed of N voice samples and a transformation matrix T with a size $N \times N$ in a way that the vector C of the transformation of N components is:

$$C = TV \tag{7.5}$$

If permutation P is applied, whose matrix is of the size $N \times N$, C is obtained by a permutation of the coefficients:

$$D = PC \tag{7.6}$$

By applying the inverse transformation, the encrypted voice V′ is obtained in the time domain as:

$$V' = T^{-1}D \tag{7.7}$$

On the receiver side, the original voice V can be obtained by:

$$V = T^{-1}P^{-1}V' \tag{7.8}$$

The only condition required for transformation is that V is composed of real values. Strictly, this technique cannot be viewed as a method in the frequency domain.

There are a number of them that can be used for this purpose, which should meet the following requirements:

1. De-correlate the data in the transformation domain, reducing, in this way, the redundancy contained in the voice.
2. Restrict the band of the encrypted voice to reduce distortion.
3. Easy and practical to implement.

The transformations that can be used are discrete Fourier transform (DFT), discrete cosine transform (DCT) and discrete Walsh Hadamard. DFT and DCT are very suitable for this type of use, as they produce a reduced residual intelligibility of the treated voice and a high deciphered voice quality. In addition, both can be easily implemented using standard hardware for processing the signals.

The delay produced by this technique strictly depends on the number of N samples, as a large number of N require a large number of permutations and processing. An equitable number to balance the security requirements and to reduce the processing time is 256 samples. With this technique, the band does not remain the same due to the discontinuities caused by the permutations. This increase in the band is compensated by setting the coefficients that are outside the speech band and the remaining M samples for permutation to zero. In theory, we can run M! permutations even though not all of them have a high degree of security, as illustrated above. These permutations must be selected with precision and should be changed for each frame to make the system more secure.

The big problem with these techniques is their sensitivity to the channel properties. In fact, due to channel distortions, it is very difficult to maintain the integrity of the individual samples, that is the samples received are not exactly the same as those transmitted, which distorts the overall quality of the samples received. These systems are implemented, for all practical purposes, with specialised hardware, which have all the features for signal processing.

Channel equalisation is therefore a very important technique to improve the quality of the received signal. It is well known that the telephone channel introduces attenuation and delay in the signal flowing through it, and in a different way, depending on the frequencies. Therefore, it is necessary to use an equaliser with attenuation and delay, which are opposite with respect to those present in the telephone channel.

7.5.1.3 Encryption in the time domain

Encryption in the time domain or time domain scrambler (TDS) seeks to destroy the continuity typical of the voice signal which, as said previously, is a signal that is slowly variable over time. To do this, the voice signal is split into a number of time segments that are processed individually, or into blocks. The major considerations for these systems relate to the size of the segment and the frame, the deterioration of acceptable sound quality and the delay generated by this process.

The segment size is very critical in these systems because this length determines how much data are contained in each segment. Of course, it is advisable to have the smallest possible amount of data in each segment, but this would result in a high number of discontinuities in a limited duration of the voice signal, generating a large number of high-frequency components that may not be able to pass through a limited transmission channel band, leading to receiving a degraded signal, which, once subjected to the inverse process, may no longer be of good quality, making it difficult to understand. As discontinuities only occur at the beginning and at the end of the time segment, time segments of reduced duration result in a greater number of discontinuities per unit of time and therefore higher frequencies and consequently greater distortion, caused by limited channel bandwidth indicated above. The second effect, which is very important, concerning the size of the segment, is the group delay. The various components at different frequencies are delayed differently due to discontinuities and therefore the size of the segment should be large enough to reduce this effect. Subsequently, the size of the segment should be larger than the difference between the maximum and minimum delay. The optimum size of the segment, which maintains a reduced degradation and an acceptable delay, is between 16 and 60 ms. The number of segments per frame, that is the length of the frame, determines the delay between the moment in which the signal enters the system and the time when the voice is received in plain text. If the size of the segment is T seconds and there are N segments per frame, the overall delay is equal to $2NT$ seconds, as the signal undergoes the same delay of NT seconds both in the transmitter and in the receiver. Obviously, it is recommended to keep this delay to a minimum, taking into account the fact that a large number of segments also create the possibility of generating a bigger number of permutations, a factor that is very important to improve system security. In addition, relatively large frames ensure that the voice can vary within the frame: otherwise, the permutation would have no effect on the intelligibility of the voice signal. A number recommended by scientific literature is 8 to 10 segments per frame.

A technique widely used is the *scrambling* of temporal elements. With this technique, the voice is divided into frames, divided in turn into segments, which are permuted in a pseudo-random manner. This technique is also called time segment permutation or time division multiplexing. The choice of the size of the segment and the frame stems from a compromise between the acceptable delay, which increases with the increase in the size of the segment, and the distortion, which decreases with the increase in the size of the segment. A good compromise is a variable segment within the range 16 to 60 ms and 8 to 10 segments per frame. In addition to the size of the segment and the frame, another very important factor is the choice of the permutation, which can be fixed for the entire message, or variable, using a suitable changeable key from frame to frame.

Even if in a frame with eight segments, there are $8! = 40{,}320$ permutations possible, not all are useful for security purposes, for the reasons discussed above. A valid test for choosing permutations that are more suitable for security purposes consists of listening to the processed signal to assess the residual intelligibility. In general, even with eight segments with 40,320 permutations available, ultimately, effectively between 512 and 1,024 of those are actually valid. The permutations are evaluated, as has been said, on the basis of the level of residual intelligibility: different systems have different levels of security and therefore use a different number of permutations.

There are different ways to implement temporal element scramblers. The principal two are:

1. window hopping;
2. sliding window.

Window hopping is also known as time block scramblers. The voice is digitised and stored in the memory, frame by frame. The segments are selected according to the permutations taken from a preselected set and stored in a non-volatile memory, converted to analogue and transmitted. With the window hopping technique, the memory is filled sequentially and the frame segments are selected only when the entire frame is stored in the memory.

With sliding window, also known as sequential temporal elements, in contrast to window hopping, the new segment is used to fill the place that has just been vacated by another element. This means that the segments in a particular frame can be received anywhere in the message rather than being confined to a single frame. It can also happen that certain segments remain in memory for a long time before being transmitted, creating problems, as these delayed segments generate a very long delay and require a rather large memory on the receiver side. To avoid this, the maximum delay time may be defined by which a segment must be chosen and transmitted. Usually, this delay is fixed to be equal to the maximum of two times the size of the frame: meaning that all segments are transmitted within the duration of two frames. If a segment exceeds a predetermined threshold, it is transmitted regardless of the sending code produced by the pseudo-random generator. With this scheme, there is no need to have frames as the signal is processed continuously, even if the frame concept is very useful to maintain control over the permutations.

The general plan that can be attributed both to the window hopping system and the sliding window system is shown in Figure 7.9.

7.5.1.4 Encryption in the time and frequency domain

We have seen that one-dimensional *scramblers*, which only operate in frequency or time domains, leave a certain residual intelligibility that decreases the security level of the encrypted voice signal. To decrease the level of residual intelligibility and increase the overall level of security, it is possible to combine techniques that work in the frequency domain with techniques that work in the time domain. Subsequently, if on the one hand, time domain techniques can help eliminate the rhythm of the speech, which tends to remain in the frequency domain techniques, on the other hand the latter tend to eliminate the spectral characteristics of the phonemes that remain unchanged in the time domain techniques. There are several combined techniques that can be used. They are principally:

1. time element scrambler with random frequency inversion;
2. band dividers with time element scrambling;
3. multi-band dividers with time element scrambling;
4. time element scrambling with variable segment size;
5. dynamic time reverberation with frequency inversion and cyclical band displacement;
6. delayed sub-bands with time element scrambling.

With regard to time element s*cramblers* with random frequency inversion, the frequency of the *scrambler* may be single carrier or multiple carrier. In the first case, a 3 kHz carrier can be selected and the segments can be inverted or left as such, depending on the code. The code varies in a random manner and consequently the segments that need to be inverted in each frame change. In the second case, by using different frequencies and depending on the key, the different segments of a frame are

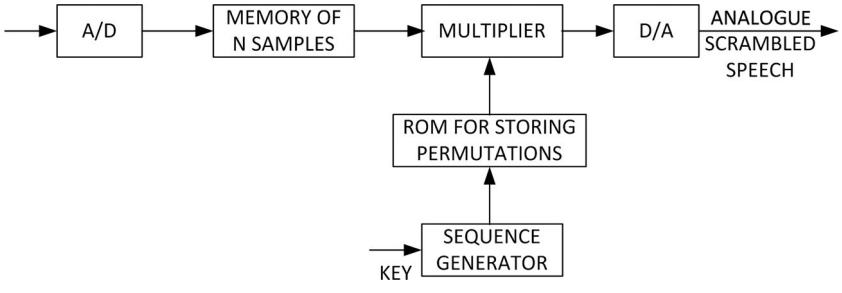


Figure 7.9 General plan for the window hopping and sliding window systems.

inverted around the various carriers. The quality of the signal can fall, in every way, due to band limitations.

For band dividers with time element scrambling, frequency techniques are combined with time element *scrambling*, of the window hopping or sliding window type, to increase the security level. In this case, *scrambling* in the time domain deteriorates the time continuity and the two-band division deteriorates the frequency ratio. The division points are randomly chosen, segment by segment. *Scrambling* is non-commutative because the codes that perform the inverse operation change, depending on the order of operations.

For multi-band dividers with time element *scrambling*, the band division combines with time element *scrambling*. Usually, a certain number of sub-bands and temporal segments per frame are selected. This system ensures a high level of security even if it appears to be quite onerous from the point of view of memory usage, which must be large enough. Disturbances caused by the channel can be considerable due to the high number of discontinuities.

For time element *scrambling* with variable segment size, the different segments of each frame are composed of a variable number of samples that increase the difficulty in finding the right size of segments. The number of samples over a given period of time also varies.

For dynamic time reverberation with frequency inversion and cyclical band displacement, multiple echoes are generated that are added to the signal. The voice signal is first inverted, then moved to the band and subsequently subjected to reverberation to achieve the desired effect.

With regard to delayed sub-bands with *time element scrambling*, voice sub-bands are delayed with different techniques prior to the application of division into bands.

All two-dimensional *scrambler*s have a residual intelligibility that is lower than the product of the residual intelligibility of one-dimensional systems that form them (time and frequency domains).

However, it is possible to combine several techniques in the time domain and several techniques in the frequency domain to generate what are called multidimensional scramblers which, rather than increasing the level of security, increase the quality of the signal.

7.5.2 Digital encryption of voice signals

Analogue *scramblers* have a certain intrinsic vulnerability due to the fact that traces of the voice parameters remain within the processed signal, which appears to have a flow similar to white noise.

Digital encryption techniques, in this sense, solve this problem, due to the use of robust ciphers, and ensure a higher level of security due to the enhanced security features that modern ciphers currently use. In digital systems, voice signal is digitised due to the use of suitable analogue/digital (A/D) converters, generating an output bit stream that can be encrypted using a flow-type robust cipher. The outflow, given its intrinsic randomness, has the same pattern as white noise in a way that masks the presence or absence, within the same, of the voice signal. The decryption system accepts as input the encrypted flow and deciphers it in a plain text flow that is sent to a digital/analogue (D/A) converter to regenerate the original voice stream.

The only drawback in using digital encryption systems is that the bandwidth must send the digital stream. In fact, if the standard telephone channel is used, whose bandwidth is only 3 kHz, there may be problems in sending the digital stream, which have relatively high speeds. Subsequently, it is possible to use wider channels of communication or the speed of the digital stream can be reduced so that it can be transmitted, without limitations, on a standard telephone channel. In the latter case, voice source encoding techniques must be used, which are illustrated in the following section.

7.6 Voice source encoding

It has already been said that one way to reduce the bit stream without the loss of data is by voice source encoding. These types of systems are called vocoders. Encoding of the voice source is based on the source–filter model, where the vocal tract has a linear filter with variable characteristics over time, which is stimulated by a periodic source to produce the resonant sounds from a source of random noise for the production of unvoiced sounds. The parameters required for the synthesis of vocal sound are sound/unvoiced choice for the appropriate selection of the source, pitch for specifying the voice source and filter coefficients for the spectral characteristics.

In most voice sources, these parameters vary step by step with respect to speech and there is no need to transmit the voice samples at the same rate. The number of parameters, the bits required to quantise each parameter and the rate at which the data are transmitted are regulated in such a way as to obtain the level of quality and intelligibility required for the voice. The disadvantage in using these systems is the fact that the sounds produced may seem artificial and synthetic due to the artificial excitation pulses that are used to simulate the behaviour of the vocal cords.

7.6.1 The formant vocoder

The formant vocoder parameterises the voice in the form of frequency formats, pitch and gain. It uses a combination of three to four filters whose central frequencies track the resonant formants of the acoustic model. This vocoder is able to save the band as it covers the full spectrum by using three to four formant frequencies. Formant frequencies can be obtained from the voice and used for their synthesis. Formant vocoders may be in parallel or in cascade depending on whether the formant frequencies used for the synthesis are connected in parallel, each excited by an appropriate source, or in series to form a chain of filters, in which the output of each filter is the input of the next filter. Usually, it takes from 3 to 4 bits for each formant to generate an intelligible voice: for this reason, the transmission speed of these systems is currently around 30 bps. The performance of these systems depends on the accuracy by which the formant frequencies are determined. Because of the precision required to determine the formant frequencies, these types of vocoders are not commonly used.

7.6.2 The channel vocoder

The vocoder channel is based on the fact that voice perception depends on the conservation amplitude of the voice itself for a short time interval. It is composed of a bank of filters whose central frequencies and the corresponding bandwidth are chosen according to the perceptive characteristics of the human ear. In Figures 7.10 and 7.11, the block diagrams of the voice encoder and decoder are illustrated, respectively.

The bank of filters is able to cover the entire band of the human voice, which extends from 300 to 3,400 Hz. The energy of the voice within each band is measured and transmitted together with the pitch and the selected voice/invoiced voice. If the system is designed properly, between 36 to 40 bits for each frame are required to generate an intelligible voice. The quality of the channel vocoder is closely related to the errors that the system makes in determining the resonant voiced/unvoiced sounds and the pitch estimation. Performance depends on the number of filters, their spacing and their bandwidth: the greater the number of filters, the better the quality of the voice produced. The number of channels in frequency is a design parameter that stems from a compromise between the bit stream and the quality of the synthesised voice. The number of channels is fixed for a given implementation. Channel and bandwidth spacing is usually non-linear in some bands, as it is used for better coverage for low frequencies that, as has already been said, are perceptually more significant.

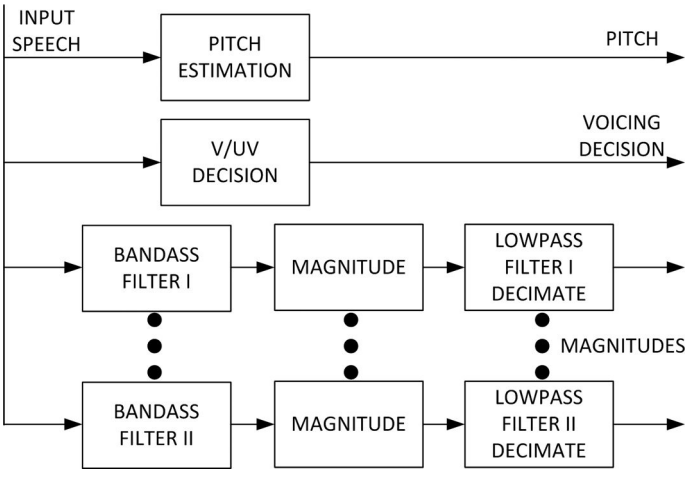


Figure 7.10 A block diagram of a channel vocoder encoder.

7.6.3 The vocoder based on linear prediction

The vocoder based on LP is based on the analysis of LP of the voice. It is a technique in the time domain based on the fact that the current value of the amplitude of the voice depends on a number of previous samples and therefore can be expressed as a linear combination of these values such as:

$$x(n) = a_1 x(n-1) + a_2 x(n-2) + \cdots + a_{n-p-1} x(n-p-1) + a_{n-p} x(n-p) \qquad (7.9)$$

Where $x(n)$ is the predicted value using the first ps past values of $x(n)$.

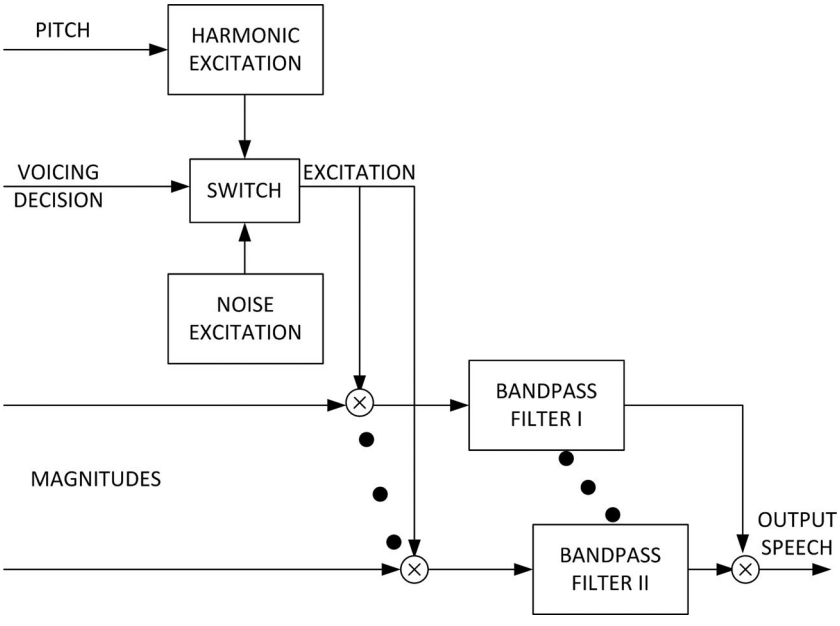


Figure 7.11 A block diagram of a channel vocoder decoder.

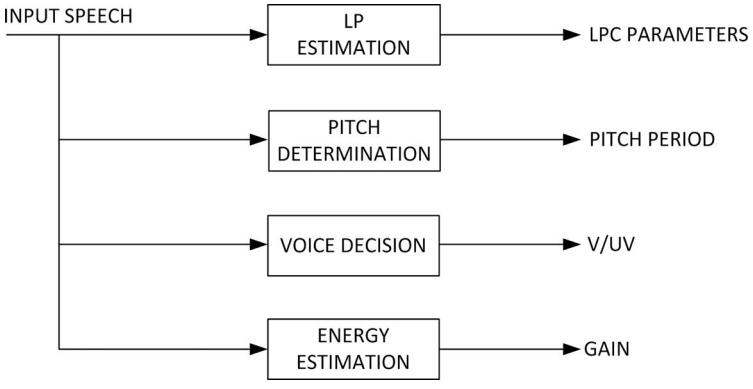


Figure 7.12 Block diagram of the LPC encoder.

The prediction error $e(n)$ can be written as:

$$e(n) = x(n) - \sum_{k=1}^{p} a_k x(n-k) \tag{7.10}$$

The coefficients a_k are called linear prediction coefficient (LPC). They are selected to minimise the prediction error E, which is given by:

$$E = \left[x(n) - \sum_{k=1}^{p} a_k x(n-k) \right]^2 \tag{7.11}$$

A number of samples should be selected so as to correspond to a frame size of 10 to 25 ms for the voice, to obtain good voice quality. A higher number of samples would lead to a crude presentation of the speech spectrum. The error E tends to be a train of pulses, spaced according to the pitch interval for the resonant sounds where it tends to be similar to white noise for the unexpressed sounds. The block diagram of the LPC encoder is shown in Figure 7.12 while the diagram of the LPC decoder is shown in Figure 7.13.

An important factor to take into consideration is that the coefficient of prediction pth depends on all the coefficients of a lower order to p and that all the prediction coefficients are not completely independent. This means that the parameter values depend on the order of the analysis. Another factor to take into consideration is the fact that small variations in the filter coefficients can result in significant changes, in a negative sense, in the synthesised voice, making these coefficients unsuitable for transmission, where the noise inevitably tends to overlap with the voice. To avoid this, it is always possible to transform these coefficients to another set of numbers, solving the effects problem due to

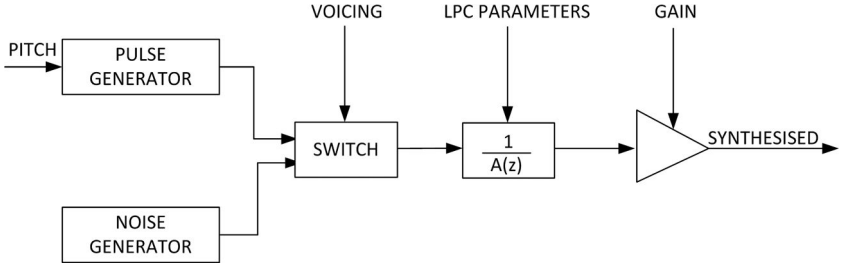


Figure 7.13 Block diagram of the LPC decoder.

their minor alterations. To do this, the reflection coefficients and the log area ratio coefficients are used.

For the reflection coefficients, the intermediate amounts k_i are referred to by this term. The term is borrowed from the theory of transmission lines, where the reflection coefficients indicate the amount of power that is reflected from the interface between two different impedances. The vocal tract behaves like a variable section transmission line from the point of view of acoustic waves. As the reflection coefficients are contained in the range -1 to $+1$, it is very simple to quantise and transmit them. The reflection coefficients, together with the gain, provide a good indicator for estimating the spectral properties of the voice.

Log area ratio coefficients can be derived from the reflection coefficients. Subsequently, the vocal tract can be considered as a lossless acoustic tube, formed by cylindrical joint sections of equal length. These coefficients are suitable for quantisation.

There are many variations to the basic LP technique. In the standard design, the pitch, the gain and the voiced/unvoiced data are directly encoded for each frame. In some cases, if voice variances are considered, a significant number of bits can be saved. The coefficients and the pitch are only updated when their variation exceeds a certain predefined threshold. In addition, there are many schemes that are designed to prevent transmission of the pitch data.

With the technique called residual-excited linear prediction (RELP), the residual error that occurs after the prediction of the voice sample using the LP coefficients is transmitted to improve the data related to the pitch, the gain and the sound/unvoiced, which are obtained from the receiver. The RELP encoder can operate in the presence of sound, at a speed of 9.6 kbps.

Another technique is code-excited linear prediction (CELP) that is a very efficient way to reduce the transmission rate to 4 kbps. With CELP encoders, each voice frame, from which both the LPC parameters and the pitch values are extracted, is divided into smaller segments and each segment is searched for an excitation code within an archive that contains a large number of data entry vectors, providing the minimum mean squared error between the actual and the synthesised speech. Therefore, only calculated vector indices and LPC values are encoded and transmitted. CELP has certain complexities but can guarantee good voice quality and transmission rates lower than 4.8 kbps. It can also improve voice quality when it operates at higher speeds.

The parameters of each vocoder referred to above are quantised and encoded to obtain bit sequences that are permuted and encrypted to achieve a high level of security. The permutation of the bits within each frame can be a way to ensure a certain level of security, even if the permutation may not be variable, as other bits cannot be added for synchronisation. The method to permute LPC parameters is only partially useful to obtain voice distortion.

For example, to encrypt the bit sequence obtained after digitising, module 2 is added to a pseudo-random sequence produced by the same kind of special generator such as those illustrated in Chapter 2. If the spectrogram produced after the encryption process is viewed, it would produce a graph similar to that of white noise.

The most important aspect of this system is the encryption algorithm that is used. Initially, retro-reaction encryption generators or auto-synchronising generators are used, in which the bits used for the encryption depend on a number of previous bits, eliminating the need to send a signal of the synchronisation generators. There is however a disadvantage in that the current bit depends on the previous n bit, due to the fact that if the receiver loses just one single bit, or makes an error due to the noise that was added along the channel, the receiver itself loses the generation sequence and decrypts the encrypted flow incorrectly. For the same reason, even in encrypted blocks, in which the output is dependent on virtually all the input bits, it produces totally incorrect results even with an error of just one single bit. The encryption techniques most commonly used in these contexts are flow ciphers, which do not propagate errors, allowing us to use the same noisy channel used for communication in plain text, even for encrypted communications.

7.6.4 The sinusoidal model

The sinusoidal model uses the source–filter model, where the source is modelled as a sum of sine waves. The encoding/decoding diagram of this model is shown in Figure 7.14.

If located within a suitable range, both the resonant sounds and the unvoiced sounds can be modelled conveniently with this technique. Using this modelling, the resonant sounds can be modelled as the sum of the harmonic sinusoidal wave spaced by a frequency equal to the base frequency, with a phase that depends on the base itself, while the unvoiced sounds can be a sum of sinusoids with random phases. In this way, the vocal wave form v can be written as:

$$v(n) = \sum_{k=1}^{C} A_k \cos(\omega_k + \varphi_k) \tag{7.12}$$

Where A_k, ω_k, φ_k represents, respectively, the amplitude, the frequency and the phase of each of the component sinusoidal C waves.

As can be seen from the diagram in Figure 7.14, spectral analysis is also carried out using the cepstrum that consists of calculating the Fourier transform of the spectrum logarithm. Its name derives from the inversion of the first four letters of the word *spectrum*. Using the same procedure for the other variable characteristics of signals, a complex cepstrum, a real cepstrum, a power cepstrum and a phase cepstrum can be calculated. This analysis is limited to the amplitudes, while the relative frequency variables are analysed to determine the pitch harmonics and the voice data.

7.6.5 Standards

Many of the vocoders illustrated previously were standardised for use in various telecommunication systems such as the Global System for Mobile communications (GSM) and many others.

7.7 Voice cryptanalysis

Cryptanalysis of voice signals involves the search for a code or a private encryption key that is used for voice signal decryption. With vocoders, the cryptanalysis tries to determine the parameters necessary to reproduce the encrypted voice.

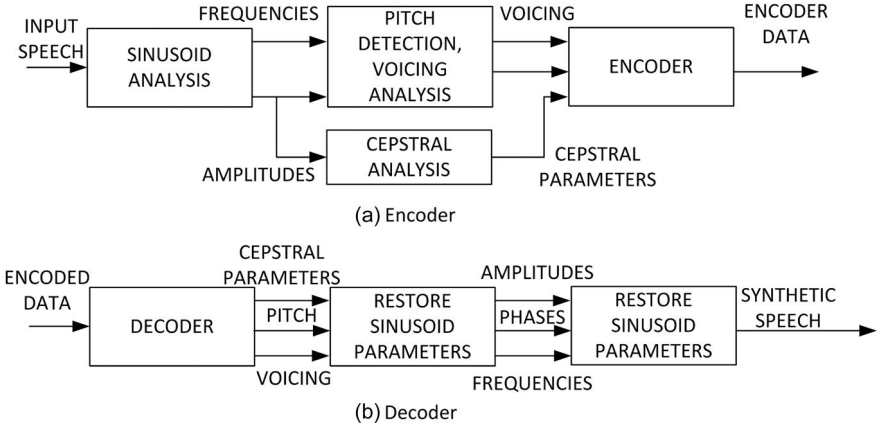


Figure 7.14 Block diagram of the sinusoidal model (a) encoder and (b) decoder.

In contrast to texts, where if the exact cipher and the encryption key are unknown, it is impossible to obtain the plain text; for voice, it may be sufficient to obtain a similar key to be able to decipher the signal in order to obtain a relatively intelligible voice signal.

It has already been seen that analogue *scramblers* are not very secure due to the residual intelligibility of the treated voice signal. They are however a good compromise between security and voice synthesis.

Cryptanalysis is carried out in two steps:

1. identification of the encryption technique;
2. decryption using various voice parameters to obtain a relatively intelligible voice signal.

Legibility can even be achieved with a not particularly good quality signal, because of the characteristics of the human ear and brain.

7.7.1 Tools and parameters for voice cryptanalysis

It has already been stated that the human ear and brain are the basic tools for voice cryptanalysis. Despite the many efforts of cryptographers, it is very difficult to completely eliminate the data contained in the human voice that the ear and the brain locate with great ease. In fact, the human ear is able to tolerate high levels of interference in the form of noise, distortions in frequency and lost parts of the signal. The brain, combined with the human ear, is able to use the redundancy contained in the speech signal to decode its meaning, even if the voice signal is not heard perfectly. Subsequently, even if the voice signal passes through a low-pass filter with a cutoff frequency of just 800 Hz, it is still possible to decode it with cryptanalysis.

The human brain is also able to focus on a particular conversation of interest, even if it occurs in the presence of a large number of other conversations. This ability is called the cocktail party effect. This can lead to a certain degree of intelligibility in some types of *scramblers* if all segments of a frame are heard at the same time. Subsequently, the combined effect of the ear and the human brain can achieve a certain intelligibility of the encrypted voice signal, even if decryption is not performed perfectly. To perform decryption, even if incorrect, the use of tools such as the spectrograph is necessary, which has already been discussed previously.

7.7.2 Using the spectrograph for cryptanalysis

The three-dimensional spectrogram of the voice in plain text can provide all the data relating to itself, such as the pitch, the sound/invoiced, the formant frequencies and the energy, in easy to read graphic form. Any distortion or alteration caused by a *scrambler* can be detected by carefully studying the spectrogram. The human ear is able to recognise the presence of any gaps in the time domain as well as the variations in the frequency domain but it is not able to identify the nature of the scrambler. If the spectrograph is used, it is possible to clearly identify the bands that are disturbed by the scrambler, which appear as horizontal gaps. If the gap also appears on the horizontal axis, it signifies the presence of a coded signal via a two-dimensional *scrambler*. Subsequently, in order to attempt a decryption operation, the various areas of discontinuity can be cut and recombined to match their various lines, eliminating the discontinuity.

In addition to facilitating the identification of the *scrambling* technique used, the spectrogram facilitates the identification of some of the parameters of the same *scrambler*. The interval between two discontinuities along the frequency axis corresponds to the frequency band. It can be measured directly by the spectrograph and can be useful in the identification of the *scrambler*.

Once the technique used by the scrambler has been identified, using the spectrogram, the decryption of the voice signal involves the reorganisation of the various segments, by matching

certain voice parameters, or the relative synthesis, if the vocoder is used. These parameters can be obtained by suitably processing the voice signal. Depending on the representation in the time or in the frequency used for the processing, these are called parameters in the time domain or parameters in the frequency domain. The basic assumption made in most analysis techniques is that the signal is processed very slowly over time, a valid assumption for voice signal. A sufficient amount of data ensures the calculation of the required parameters without error. Subsequently, the relatively short segments from a time perspective, in which the human voice has constant characteristics, can be identified and processed. There are different parameters that can be extracted from these segments, which can be used for cryptanalysis. Assuming that the signal is almost stationary, all parameters can be calculated for time durations of the order of 10 to 30 ms. The main parameters in the time domain are a short-term energy function, a short-term auto-correlation function and a short-term average amplitude function. The main parameters in the frequency domain are fast Fourier transform (FFT) spectrum valuation, a banks filter system, homophonic voice processing, modelling techniques, LP spectral valuation and maximum entropy spectral valuation. Furthermore, the following decryption methods in the time domain can be used: frequency inverters, random frequency inverters and selective inversion. All parameters listed above are the source characteristics of the human voice vocal tract. In particular, energy and pitch are specific characteristics of the excitation source of the vocal cords. The characteristics of the vocal tract depend on the voice. The movements of the articulators are responsible for changes in the shape of the vocal tract and require a finite time to be followed, due to their mechanical inertia, and this generates a voice signal slowly varying in time.

7.7.3 Analogue methods

There are basically three ways to decipher the voice. They are as follows:

1. An estimation of small segments of the voice parameters and use of the same characteristics to match adjacent segments. Decryption may require the comparison of individual parameters and may require the comparison of multiple parameters.
2. Voice synthesis by extracting the voice parameters from an encrypted version.
3. Identification of phonemes by simple comparison/adaptation techniques, using a set of codes. This technique can be used as the voice can have a finite number of characteristic vectors.

The basic sequence of deciphering the voice usually requires the use of a computer equipped with systems for converting analogue/digital (A/D) or digital/analogue (D/A) and with the aid of spectrographs. A typical system is illustrated in Figure 7.15.

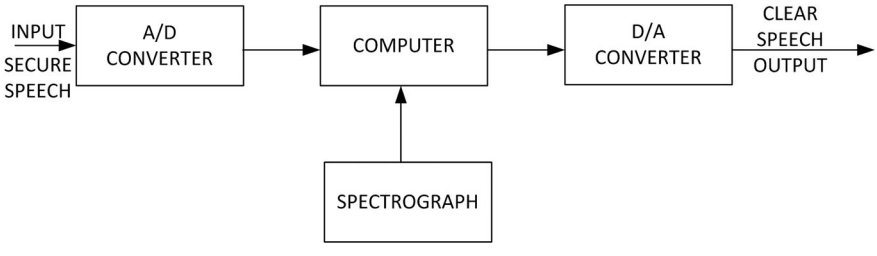


Figure 7.15 Diagram of a typical cryptanalysis system.

7.7.4 Cryptanalysis of digital ciphers

It has already been said that analogue encryption systems still leave fundamental traces within the encrypted voice signal that can be used to decipher it. In addition, continuous listening at well-defined time intervals together with the use of the spectrograph are valuable instruments for deciphering.

In the case of strong digital ciphers, the encrypted signal is very similar to white noise. In this case, listening to and the use of a spectrograph are not valid solutions to decipher the voice signal, features that make digital encryption much more difficult to decipher compared to analogue encryption.

As for digital cryptanalysis techniques, these are similar to those illustrated in Chapter 2, to which reference should be made.

7.7.5 Linear prediction vocoder cryptanalysis

In systems with LP, it can be seen that the voice depends on its characteristic parameters which are:

1. pitch;
2. voiced/unvoiced data;
3. gain;
4. reflection coefficients.

In LPC systems, these parameters are quantised, encoded and ciphered using pseudo-random sequences and transmitted. With these systems, most of the voice redundancy is removed and subsequently, one element of vulnerability having been removed, which can be used for cryptanalysis, although their security is less than that of digital systems, is considered safe, for the reasons already discussed.

The basic properties of the voice, and the pseudo-random sequences may however be analysed, when using these systems, in an attempt to obtain useful data on the encrypted message. Subsequently, it is possible to consider the following factors:

1. Each coefficient, after the linear prediction analysis, has a predetermined number of bits and the relative position of the coefficients in the frame remains the same.
2. A sequence of zeros in pseudo-random sequence leaves some of the coefficients unchanged, while a sequence of one, in this sequence, reverses some of the coefficients.
3. The human voice can be fully portrayed by a finite number of samples.

A possible approach that may be employed, using the factors outlined above, is to form samples of the reflection coefficients of the voice in plain text and to try to match the reflection coefficients of the encrypted voice with the reflection coefficients of these samples. The sample that has the maximum number of concurrences and reversals is the most suitable for the encrypted voice.

7.8 VoIP systems security

Voice over IP (VoIP) has already been discussed in Chapter 1 and will not be discussed again in the following, except with regard to its security aspects.

These systems utilise both fixed networks and wireless networks, and as they are used for communicating the human voice, relevant aspects of security are illustrated in this chapter.

VoIP systems transmit voice packets via networks, such as the Internet and facilitate large cost savings in voice communications. Unfortunately, the Internet is not designed to ensure the quality of service required for voice communications and as such, quite often, VoIP technology does not guarantee high-quality communications, which can often be interrupted during transmission. As such,

the Internet is less reliable than dedicated telephone networks. Other problems using VoIP systems derive from the fact that during transmission over the Internet, packets are subject to loss, fragmentation, delay, latency and jitter, which inevitably results in the degradation of voice communications that, in order to remain intelligible, must have good transmission channel quality. These factors, although not constituting a threat, can influence the confidentiality and the reliability of VoIP systems. Latency can be caused by a number of routers that packets must traverse on the network: if they must execute too many passages over the router to reach the final destination, the latency itself can reach such a high value as to make communication extremely difficult, if not impossible. Jitter occurs when packets follow the same path on the network, but arrive at different times, causing delays. Finally, as VoIP systems are not able to use all the features of the dedicated telephone networks, such as reliability, flow control, error detection and correction, the relative IP data packets may arrive at the destination synchronised or fragmented, or may not even arrive at all. There are different standards and products on the market, each having its own merits and defects.

It has already been said that Transfer Control Protocol/Internet Protocol (TCP/IP) protocol was developed for applications that are not sensitive to the time factor or the band factor such as email, file transfer and remote access. The main purpose was to connect multiple systems via a robust network. As such, if data transmission systems are able to use it fully, voice and video transmission systems, being subject both to delays and available band, can suffer as a result and cannot guarantee services of acceptable quality. Table 7.1 illustrates the different types of traffic and their features.

In public switch telephone network (PSTN), the cost of calls is calculated on the time used on the dedicated line. In packet switching networks, the fee is calculated based on the amount of band used. This means that, while with PSTN, moments of silence that occur during a conversation between two subjects are included in the cost, this is not the case for switched networks, leading to considerable cost savings.

The different vulnerabilities with this type of system can be divided into:

1. physical;
2. natural;
3. hardware;
4. software;
5. communications;
6. human.

Table 7.1 Different types of traffic and related features.

Type of traffic	Bandwidth	Data exchange modes	Sensitivity to latency	Sensitivity to jitter	Sensitivity to loss
Transfer of data blocks	10 to 100 Mbps	Period between two subjects	Low	None	Low
Data transaction	<1 Mbps	Burst between two subjects	Average	None	None
Voice and fax	8 to 64 kbps	Variable between two subjects or between multiple subjects	High	High	Low
Multimedia (voice + images)	Up to 384 kbps for video	Variable between two subjects or between multiple subjects	High	Average	Low
Video on demand or by streaming	28.8 kbps to 1.5 Mbps	Variable between multiple subjects	Low	Low	Low

These vulnerabilities are not specific to VoIP systems but are common to all networks that transmit voice data, regardless of the transmission method.

With regard to physical vulnerabilities, these occur due to the poor physical security that is offered to local hosting routers, switches, gateways, servers, etc., which do not often have access control systems. This leaves these systems vulnerable to theft, tampering and destruction.

With regard to natural vulnerabilities, such as violent storms, these can seriously compromise the functionality of the network, thus interrupting voice communication services.

With regard to hardware vulnerabilities, the malfunction of one or more network devices can lead to the interruption of voice communication services or to a denial-of-service (DoS) situation.

With regard to software vulnerabilities, if these have internal security flaws, these may be used to hack the systems or to install *trap doors* to be able to access a system easily.

Communication vulnerabilities can lead to the interception or disruption of voice communication channels, with serious security consequences.

With regard to human vulnerabilities, these can result in the impairment of the system due to problems caused by poorly trained staff that perform incorrect actions.

VoIP systems operate both on fixed networks and on wireless networks, always sending voice data in the form of digital packets in TCP/IP format. Voice packets are sent from a point of origin to a point of destination indicated in the packet header. Subsequently, these systems rely on Real-Time Protocol (RTP) that is used to ensure time support when sending packets that contain voice data. In a manner similar to TCP that operates at the transport level for the transmission of data and guarantees connection-oriented services, RTP provides a certain level of reliability in the transmission of voice packets, operating above User Datagram Protocol (UDP). It should always be remembered that TCP is not the best solution for VoIP systems due to its reduced reliability in terms of delivery.

Because of the potential loss of traffic due to congestion, VoIP systems operate more efficiently in virtual private network (VPN). With these VPN systems, voice packs are condensed and sent from a private network to the Internet, where they are sent in encrypted form, increasing the level of security of communication and then arrive at the private network destination, where they are decrypted.

With wireless systems that use VoIP solutions, it is necessary to pay attention to the same vulnerabilities, already illustrated in Chapter 6. Subsequently, wireless networks that use Wired Equivalent Privacy (WEP) are particularly vulnerable, given the known vulnerabilities of this security protocol.

In VoIP systems, it is very important to consider the three key aspects of communications security, which are:

1. confidentiality;
2. integrity;
3. availability.

It should be briefly remembered that:

1. confidentiality means that the data contained, transformed or transported by a communication system cannot be read by unauthorised parties;
2. integrity means that the information contained, transformed or transported by a communication system cannot be modified by unauthorised parties;
3. availability means that the information contained, transformed or transported by a communication system is always available, upon request, to authorised individuals.

Confidentiality guarantees a certain reliability due to the fact that the data remain confidential as far as possible. Subsequently, it is possible to use cryptography to improve the level of confidentiality of a system or network. With regard to wireless networks, there is a major problem as these are subject to interception. As such, it is necessary to increase the strength of the encryption algorithms used in such a way that, even if the transmitted signals are intercepted, they cannot be breached.

The same problem arises in the wireless area with regard to integrity because the signals can be captured during transit and replaced with other data. In addition, the latency or delay in the transmission of a VoIP packet can contribute to corruption of the integrity of the data if the packet is lost.

VoIP uses IP datagram packets of the type containing voice rather than data. As such, it is subject to the same vulnerabilities of IP networks, including *spoofing*. In fact, it has already been seen that when launching a *spoofing* attack, a destination host is attacked, possibly through a trusted host. In this situation, the attacker usually disables the trusted host to check the packets that are exchanged between the two legitimate systems to impersonate the trusted system. The trusted system is impersonated by changing the source address in the IP header. Since IP is a connectionless method for transmitting packets, there is no error control system, because this task is left to the upper levels of the International Organization for Standardization / Open Systems Interconnection (ISO/OSI) stack. Once the IP pack has been modified, if the destination host accepts authentication based on the address, it is deceived as it assumes that the pack comes from a trusted host. In the case of VoIP, confidential conversations can be eavesdropped and listened to by an illegitimate third party.

To reduce the risk associated with this type of attack on VoIP systems, it is necessary to use an authentication mechanism that is stronger than that based on the basic address authentication. As such, hosts should only establish communication connections with trusted parties. In addition, as in the majority of cases, in order to strengthen these security systems, it is essential to use strong encryption systems. To increase the security level of VoIP systems, many manufacturers have created dedicated firewalls that ensure security and authentication. VoIP firewalls are not just simple firewalls used for authentication.

It has already been seen that problems increase if VoIP systems that use wireless networks are used, due to the ease with which electromagnetic signals can be intercepted by using dedicated scanners and operating on suitable frequencies, or using dedicated *sniffers* and positioned appropriately. As such, *sniffer*s can monitor network traffic and identify VoIP packets, being able to read their contents if they are not encrypted. The situation becomes more complicated if the authentication mechanism transmits data in plain text, allowing a potential attacker in order to read all the authentication credentials to access at a later time. In most VoIP systems, the authentication and transmission of voice packets is performed by encrypting all communications, using proven secure algorithms such as the triple Data Encryption Standard (DES) or Advanced Encryption Standard (AES).

Another type of attack that can be conducted against VoIP systems is DoS that is capable of compromising these systems from the point of view of confidentiality, integrity and availability. An attack of this type cannot just compromise communication services but may constitute a danger to an organisation in terms both of image and financially. A system that is not particularly secure from the point of view of software and hardware can be an easy target for a DoS attack.

With regard to wireless networks, this can be accomplished by simply resorting to disturbance (jamming), emitting electromagnetic signals of considerable power in the relative band in such a way as to disturb normal communications that occur in this band. One way to avoid these attacks is by using special equipment that reveal or neutralise hostile signals. In relation to wired networks, this can happen in many ways, as illustrated in Chapter 5. One way to avoid this is to use intrusion control systems that recognise packets characterised by typical attack data and ignore them, preventing their systems from beginning to respond to a large number of these attacks, overloading and blocking them. With regard to VoIP systems that use both wireless and wired networks, it is possible to make use of dedicated firewalls, antivirus software, VPN, encryption and specific security plans. Physical and technical measures can be taken into account, directed at gateways, routers, servers and various apparatuses used in order to reduce the risk of specific attacks such as *spoofing*, eavesdropping or DoS against VoIP systems.

This page intentionally left blank

CHAPTER 8

PROTECTION FROM BUGGING

8.1 Introduction

The evolution of technology has led to the proliferation of a multitude of environmental bugging devices which are now available almost everywhere, from specialist shops to national and international Internet sites, and at a relatively low cost.

It is very important to remember that the task of environmental bugging is restricted to law enforcement for investigative purposes and is a serious crime if perpetrated by a private citizen.

This consequence does not deter bugging operations due to the availability of a large number of devices on the market at a low cost.

Bugging devices and miniature cameras are now so small that they can be installed, for example, in:

1. drawers;
2. lamps;
3. ceiling lights;
4. bookcases;
5. curtains;
6. telephones and telephone lines;
7. electrical junction boxes, electrical sockets, plugs, adaptors, extension cables, etc.;
8. furniture;
9. false ceilings;
10. household appliances and electronic devices (alarm clocks, portable radios, portable audio systems, portable desktop calculators, etc.);
11. motor vehicles.

For this reason, to ensure secure communications for your business or personal environments, it is essential to be familiar with bugging devices, their limitations, and tools for the detection of bugging devices and procedures and actions to implement if you suspect you are under surveillance.

8.2 Devices for environmental bugging

Bugging devices are available in the following:

1. bugging devices and miniature cameras;
2. directional microphones;
3. tracers based on cellular technology and global positioning system (GPS);
4. various devices.

This list is illustrative and by no means exhaustive, given the ongoing production of new devices due to technological advances.

8.2.1 Bugging devices and miniature cameras

Bugging devices and miniature cameras are small in size and can be concealed virtually anywhere, both within the environment to be spied on and directly on persons.

They can operate over radio frequency and microwave bands, via cellular technology, through standard electric cables or via power line technology, over the infrared band, with ultrasound, directly connected to mobile or fixed telephone lines, emitting a signal detected in the radio frequency or microwave band.

8.2.1.1 Radio frequency and microwave bugging devices

In this section, the term "bugging device" refers to devices capable of transmitting voice, video and data signals from a distance, within the range of radio frequencies and microwaves.

Every electronic device emits electromagnetic waves characterised by a frequency that is typical of each apparatus. This emission is due to the oscillation of the electrical signals within them, which produces an inevitable emission of electromagnetic waves that may be deliberate, as in remote transmission devices, or inadvertent, resulting in serious problems in cases where these signals may be intercepted remotely.

Bugging devices are designed to detect environmental sounds, images or data from electronic devices (computer, fax, data connections, etc.) transmitting remotely through electromagnetic waves.

For bugging devices, the following general features apply:

1. The maximum antenna emitting power of radio pulses is usually between 5 and 50 mW.
2. With these emission power levels, the range is usually not greater than 100 m.
3. If transmission is rapid, individual pulses usually have a duration ranging 20 and 100 microseconds.
4. The bugging range is usually the same as a normal sized room of 20 to 30 m^2.

The data mentioned above indicate that there may be bugging devices with a greater emission power, which are capable of achieving higher flow rates but require a hidden power supply, and consequently a larger battery, which reduces the possibility of concealing these devices unless they are connected to the mains supply.

Table 8.1 illustrates a summary of the current scenario concerning bugging devices using radio frequencies or microwaves.

Table 8.1 Summary of the current range of bugging devices using radiofrequencies or microwaves.

Bugging devices can be powered by either batteries or mains supply.

If battery powered, even if the consumption is relatively low, it is necessary to replace them frequently, for which the person who installed the bugging device must have regular access to replace the battery, making the bugging operation all the more difficult, as this may arouse suspicion and exposure. Bugging devices that use batteries have low energy consumption and consequently a lower

emission power, and subsequently a reduced flow, for which an individual bugging device must be located nearby, possibly arousing suspicion. However, these bugging devices, characterised by their small size and small batteries, can be easily hidden almost anywhere.

If powered by mains supply, the bugging device must be connected directly to the grid and can enjoy unlimited power (as long as there is power supply) and it is not necessary to have frequent access to the bugging device to replace batteries as these are not required. Typical installation places are electrical junction boxes, where the bugging device can be easily concealed and connected to the grid as well as to plug adaptors, power supplies, wall sockets, etc.

The range of bugging devices, especially if battery powered, is not usually more than tens of metres in normal atmospheric conditions, reaching a maximum of 100 m; the distance decreases if correctly installed indoors, where walls and ceilings can result in weakened electromagnetic waves.

For this reason, it is necessary that within this range there is a receiver (with its listener) and any recording device, which remain hidden.

Therefore, it is important to check for any changes or suspicious behaviour that may occur within 100 m of the room in question.

Regarding the transmission mode, the following general guidelines apply:

1. The signal can be transmitted in plain text and then be heard with a standard receiver or scanner.
2. The signal can be encrypted and thus can only be heard with a special receiver but detectable with a spectrum analyser. Encryption can be:
    a) analogue: band inverted signal or encrypted using one of the techniques discussed in Chapter 7;
    b) digital: signal encrypted with DES algorithm or other algorithms described in Chapter 2.
3. The signal can be scrambled and thus audible only with a special receiver and not detectable by a spectrum analyser. It can be:
    a) direct expansion spread spectrum;
    b) frequency hopping spread spectrum.
4. The signal can be emitted at predetermined intervals, or upon request, remotely, in plain text mode, encrypted mode and spread spectrum mode. In this case, the bugging device is equipped with a digital recorder that records all audio signals, video or data during the bugging operation and then compresses and transmits in a single sequence (burst) at well-determined time intervals or by remote command sent by radio frequency. These types of bugging devices, especially those using spread spectrum mode, are the most difficult to detect because the emissions occur in an unpredictable manner and for very short periods of time, being compressed signals. Additionally, the use of spread spectrum techniques further complicates the counter-spying activity. These bugging devices are still relatively expensive and are not within the price range of everyone. However, these bugging devices can be detected even if switched off or inactive due to non-transmission using detection systems such as non-linear seam detectors that will be discussed below.

Bugging devices currently available commercially use wireless fidelity (Wi-Fi) technology and operate within the industrial, scientific and medical band (ISM) band. These bugging devices use the same Wi-Fi component access points and are thus available at low cost, Wi-Fi being a developed and well-established technology. Since the mentioned bugging devices use the same spread spectrum techniques, they are difficult to detect because of the technique used and because they operate within the ISM band which is very cluttered with all the devices that use the same technology, now widely available in all workplaces and homes. These bugging devices are easily able to transmit voice, video and data over the same range as normal Wi-Fi access points, being capable of reaching maximum distances in the order of 100 m, in the absence of barriers. These distances may be increased, from the receiving side, by using special directional antennae that can be pointed in the direction of the interior of the site itself while being conveniently situated outside. In figures Figures 8.1 to 8.5 some pictures of bugging devices are shown.

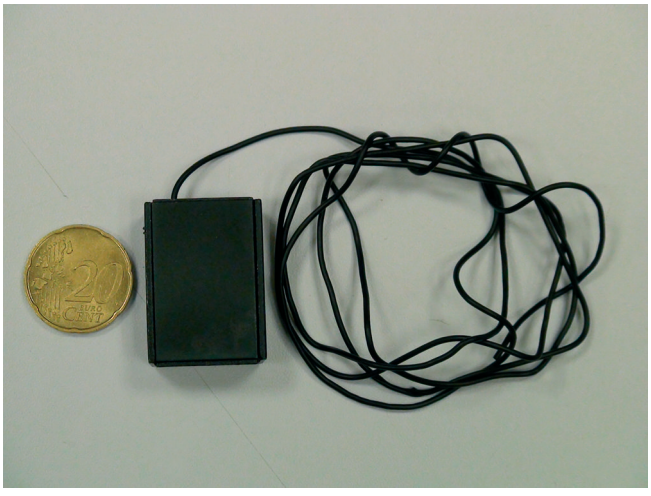


Figure 8.1 A bugging device.

8.2.1.2 Bugging devices based on cellular technology

Bugging devices incorporating cellular technology, mainly the Global System for Mobile communications (GSM) type used only for voice bugging or Universal Mobile Telecommunications System (UMTS) for video bugging, are actual mobile phones that can transmit environmental sounds from any distance using cell technology. They must be equipped with a subscriber identity module (SIM) card in order to have access to the cellular network. Their size is very small in order to be easily hidden.

They are concealed within the environment to be surveyed and work on external call, sending the sounds picked up in the environment to the bugging device that calls externally and that has an activation code for bugging to prevent inadvertent calls activating it. Certain types of bugging devices are activated when they pick up sounds within the range to be monitored (vox functionality).

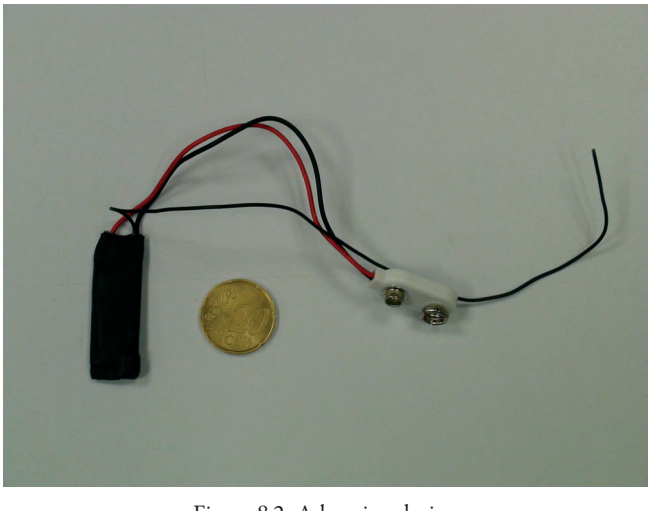


Figure 8.2 A bugging device.

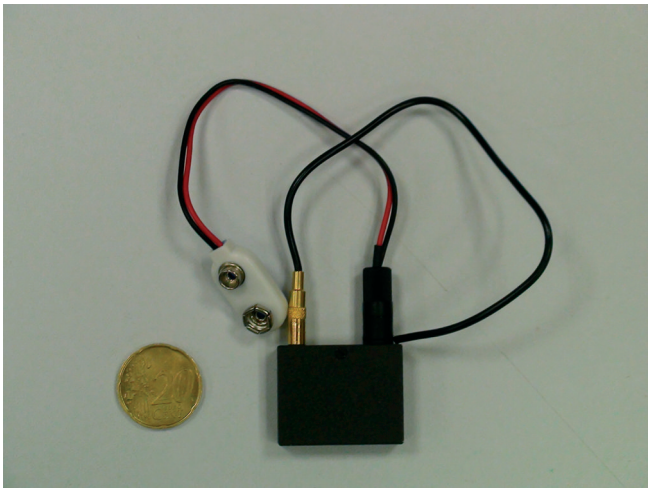


Figure 8.3 A bugging device.

Their autonomy is, on average, a few days in standby mode and between 6 and 8 h of continuous transmission, depending on the battery capacity. Their transmission life can be unlimited if connected to mains supply.

The great advantage is that, when on standby mode, they emit electromagnetic waves periodically when communicating on the control channel the data required to remain on the network to the nearest radio base station. For this reason, they are difficult to detect; if they are not transmitting yet, they can always be identified due to non-linear seam detectors and, in any case, can be rendered ineffective with jammers that will be discussed later.

If they are identified, it is theoretically possible to trace the source, referring to law enforcement agencies that, via the installed SIM card, can obtain the owner's identity. Unfortunately, this is not always possible because SIM cards are mostly purchased using false documentation and turn out to be owned by non-existent, innocent or even deceased persons.

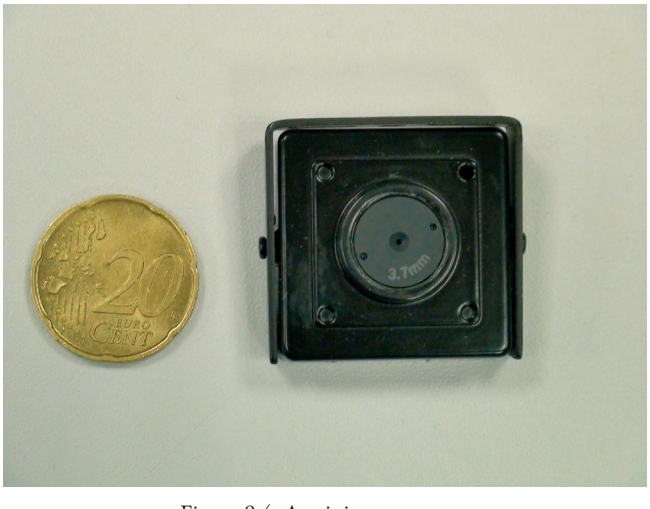


Figure 8.4 A miniature camera.

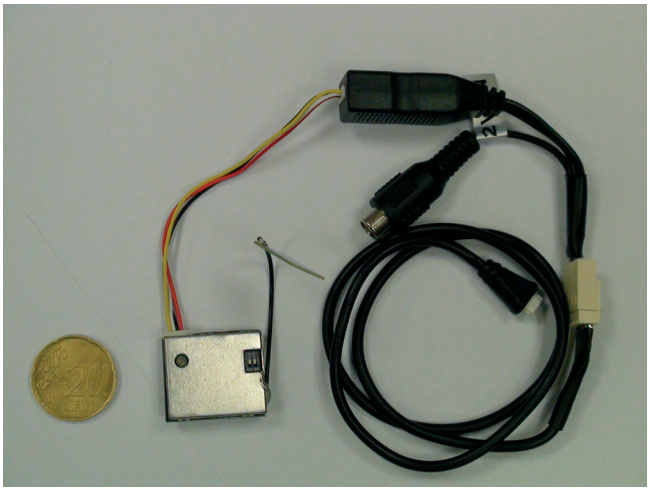


Figure 8.5 Video transmission stage for a miniature camera.

8.2.1.3 Mains carrier bugging devices

Mains carrier bugging devices are connected directly to the grid from which electricity can be accessed to power them, and send the signal that contains environmental sounds captured in the form of electrical signals in the very low frequency (VLF) band (60 to 500 kHz). This signal is transmitted via the power line and partially transmits in the form of electromagnetic waves which, given their relatively low frequency, are not detected by most anti-bugging tools, which generally operate at higher frequencies.

Concerning reception, it is necessary to connect a suitable receiver to the power line, possibly before the electricity meter as the meter greatly weakens the signal emitted by the bugging device. This receiver is usually powered from the power line itself and picks up the signal emitted by the bugging device, making a bugging range up to a distance of a few hundred metres possible, due to the ease with which the signal can transmit on the electric power lines.

8.2.1.4 Infrared bugging devices

Infrared bugging devices encode environmental sounds captured by way of an infrared beam which can reach a distance of 100 m.

The great advantage/disadvantage is that infrared waves are optical radiations which transmit in a straight line. This means that the emission can only be intercepted if the beam is interrupted. In this way, the interceptor does not receive any signal and it is certain that the bugging device has been discovered.

Another great disadvantage is that infrared waves are greatly weakened by glass. This means that if the bugging device is located within an area with windows, the beam will have great difficulty in transmitting outside the same area, rendering the bugging operation difficult. For this reason, in most cases, the microphone is installed inside while the transmitter is installed outside, within view of the bugging device receiver.

Furthermore, infrared waves are greatly weakened by fog and rain, making it difficult to transmit the beam outdoors over long distances in adverse weather conditions.

The great advantage is that their emissions can be intercepted only when the beam is broken, revealing an intrusion in the communication channel, which is definitely not an easy task because

infrared waves are not visible to the human eye and only can be visible with suitable detectors discussed below.

8.2.1.5 Ultrasonic bugging devices

Ultrasonic bugging devices encode environmental sounds picked up by way of ultrasonic waves, typically at a frequency of 40 kHz, which can reach a distance of tens of metres.

The great advantage/disadvantage is that ultrasonic waves are inaudible and propagated by bouncing of the walls in the controlled environments: if the controlled environment has wallpaper or if the doors are closed, ultrasonic waves cannot reach the outside environment where they are captured by the bugging device receiver.

These bugging devices can be easily detected by suitable counter-bugging devices which are discussed below.

8.2.1.6 Telephone bugging devices

Telephone bugging devices are of two types:

1. bugging devices installed directly in the telephone apparatus;
2. bugging devices installed on the line between the telephone and the exchange.

These bugging devices are installed parallel to the telephone line, with two simple terminals, from which they extract both the energy required for their operation (relatively small) and the audio signal of the telephone conversation.

They activate as soon as you pick up the telephone and start transmitting the conversation via electromagnetic waves.

For this reason, when you perform a debugging operation of a suspected environment, among the various operations to be performed, it is necessary to lift the telephone handset, if present, in order to activate any possible telephone bugging devices installed in the instrument and to check using the debugging device for the presence of suspicious signals.

In some cases, these bugging devices are installed along the line between the apparatus and the exchange which can also be located a few hundred metres from the site in question. Often they are installed in junction boxes that are found in the street and which are easily accessible. In this case, emission is more difficult to detect around the suspected environment because lifting of the telephone handset results in electromagnetic wave emissions at low strength, at a distance that can be substantial compared to the environment itself and, for this reason, this emission cannot be detected by the debugging devices being used within the environment. In this case, the appropriate tools used are those that use reflectometry technology by which the appropriate signals are sent on the telephone line and the time it takes to get a return echo is measured: based on that time, any unauthorised devices connected parallel to the line can be traced.

The emission power of these bugging devices is modest, not having a great amount of electrical power inherently available from the telephone network, unless recourse is made to auxiliary batteries that make it possible to bypass the necessity for using electrical energy from the telephone line itself. For this reason, the flow rate is relatively low, forcing the eavesdropper to remain close to the suspected site, unless the bugging device is installed along the telephone line: in this case, as has been said, the eavesdropper can be situated at distances of a few hundred metres from the suspected site.

8.2.2 Directional microphones

Directional microphones are special microphones capable of significantly increasing the sounds coming from specific directions.

In addition to the use of very sensitive microphones and an adequate electronic amplification range, they use pointing devices and concentration of sound consisting mostly of antennae that can pick up sound coming from the direction to which they are directed, focussing it within the core of these same antennae, where the microphone is located.

Obviously, along with the desired sound, a large amount of environmental sound is also picked up, which is partially erased electronically by the device in order to make the voice clearly audible.

Their range can vary from tens of metres to 100 m: the higher the sensitivity of the microphone and the magnitude of the capacity of the antenna, the greater the distance that can be reached.

On the one hand, concentration antennae play a role of primary importance, but on the other hand they are very cumbersome objects that can attract the attention of passers-by, making the presence of a bugging device conspicuous. For this reason, capturing devices of various sizes are used which are less conspicuous, though their performance is inferior.

Listening is usually done via headphones to prevent the captured sound being picked up and re-emitted through the speaker and picked up again by the microphone, resulting in the typical annoying whistle of the so-called Larsen effect.

These microphones are equipped with a device for automatic gain control that prevents sudden loud sounds, such as the sound of a horn of a vehicle in transit that can deafen the listener.

The main problem with directional microphones, as is the case with all microphones, is the effect of wind that can produce a very annoying shrill in the microphone which can reach noise levels inhibiting the effective capturing of the voice.

8.2.3 Environmental bugging using laser devices

When conducting normal activities within an environment, such as working and talking, this produces sounds that induce vibrations on the perimeter windows of the same area.

If you point a laser beam at these windows, the vibrations of the glass induced by internal environmental sounds modulate the beam that is partially reflected by the glass. If you point a very sensitive optical receiver, which operates at the wavelength of the laser beam, the receiver is able to demodulate the reflected laser beam, reproducing the environmental sounds.

It is obvious that the more effective this phenomenon, the weaker and thinner the glass, as with normal glass, where optimal results are achieved, and poor results are achieved with double glazing (which is increasingly used to reduce the heat loss) or reinforced glass (which is extremely rigid and robust).

Laser can function at visible wavelengths, for which the pointing method occurs with sight, or within the infrared range, in which the pointing is focused into a viewfinder. Obviously, if visible frequencies are used, the beam can also be seen from inside the environment, clearly showing the directional origin of the beam, given the high directional characteristics of lasers.

Because of this high directional ability, it is possible to aim the laser at a distance of tens of metres or more. The only problem with long distances is not the problem with aiming, but that of air turbulence, especially during the day and also on hot days, tending to distort the beam itself, which sees these distortions as noise. The latter is then added to the modulation emitted by the environmental sounds on the glass, rendering those sounds incomprehensible if the level of air turbulence becomes too strong.

8.2.4 Trackers using GPS technology

Trackers using GPS technology are small devices usually attached illegally to cars to track their movements.

They use, in fact, the GPS system to gather information about positioning.

There are two types of devices:

1. devices that transmit the location via cellular technology;
2. devices that store the information internally about movements (stand-alone).

Trackers using cellular technology send the positional information via a data connection (usually via the general packet radio service (GPRS) system) or via short message service (SMS). Given the necessity for continuous transmission, they must have an autonomous energy supply and are, therefore, equipped with a battery of appropriate size. For this reason, their size is very small.

Stand-alone trackers, on the contrary, store the data related to movements internally that are then downloaded directly from the device, which must be removed from its original location on the car and, optionally, be replaced by another similar device to continue, without interruption, the tracking operation. As these devices do not have to transmit any data, their size is relatively small and their autonomy is much higher compared to trackers that use cellular technology.

Obviously, if you intend to track a person in real time, it is necessary to have access to trackers that use cellular technology, whereas if you prefer tracking that is not in real time, then stand-alone trackers can be used.

These devices are very insidious and seriously violate the privacy of individuals, as with all other devices discussed in this chapter. Fortunately, if you suspect that you are the object of a surveillance operation, you can disable these devices using suitable jammers that will be discussed later.

8.2.5 Mobile phone bugging devices

Mobile phone bugging devices are standard phones in which spy software can be installed to monitor all activities that occur on the phone, such as conversations, calls and SMS, and send them to a desired number.

To install the bugging software, it is necessary for the interceptor to have access to the phone for the time necessary to install the control software and subsequently to return the phone to its rightful owner without the latter being aware of it. Alternatively, these modified phones are given as gifts to the person to be bugged. For this reason, it is important to be careful when accepting mobile phones as gifts.

Depending on the model, the espionage techniques differ.

Regarding the number being called or the calling number, this number is recorded by the installed bugging software and is sent to the interceptor via SMS which leaves no trace on the phone itself. Obviously, this activity results in abnormal SMS traffic and can alert an attentive user. In this sense, some telephone companies offer a service for sending SMS with the cost payable by the recipient and in this case, interception activities become entirely transparent to the intercepted user who is not charged with the additional costs.

With regard to voice calls, these are usually stored in audio file format invisible to the phone owner, which can be downloaded without the owner's knowledge via the Bluetooth port. This requires that the eavesdropper approaches the owner of the phone to ensure there is Bluetooth available and within a few seconds, depending on the size of the recorded audio file, the download takes place.

In other cases, if your computer has a data connection, the audio file is downloaded via the computer using, for example, the email utility, taking care not to leave any trace on the phone.

If the phone has an integrated GPS receiver, the bugging software can be programmed to send the relative data to the phone's location, rendering the phone itself an effective tracer.

8.2.6 Other devices

In addition to the devices discussed above, there are also other devices for surveillance including:

1. stethoscopic microphones;
2. miniature audio and video recorders;
3. computer keyboard keystroke recorders or key catchers;
4. bugging software for computers;
5. portable document scanners.

8.2.7 Stethoscopic microphones

Stethoscopic microphones are microphones that are attached directly on the wall adjacent to the room that you want to spy on. They are equipped with a suitable amplifier and audio listening occurs directly via the headset. Virtually, all the devices are equipped with an output for the audio recorder and many of them are equipped directly with an integrated digital recorder. In this way, you can leave the device directly in position and in recording mode and listen to the sounds intercepted at a later date. The audio output can be connected directly to a bugging device that can send the signal remotely, exposing itself, however, to the risk of the emitted signals being intercepted. The quality of the intercepted sound depends directly on the conductive capacity of the walls of the intercepted area that act as an attenuator/filter and that can excessively distort the environmental sound rendering it unintelligible.

8.2.8 Miniature audio and video recorders

Miniature audio and video recorders are characterised by their extremely small size as they are solid-state devices. Because of their technological make-up, they are characterised by low power consumption and a large memory capacity that enables them to record for many hours.

They can be kept in specific containers, in order to be carried around and to record directly the conversation with the person you talk to, or be hidden in innocuous devices such as pens, clock radios, watches, glasses and ties. Recorders in universal serial bus (USB) pen drives are also currently available on the market, which seem to be absolutely harmless and are capable of storing several hours of recording.

Do not forget that the majority of handheld mobile phones come with an audio recording function that can be used as normal recorders to record conversations with the person you talk to.

These devices are very difficult to intercept, especially if they are contained in normal common devices or in any case well concealed on persons.

It is different for cameras for which there are suitable devices, which will be discussed below, capable of revealing their existence.

8.2.9 Keystroke recorders on a computer keyboard (key catcher)

Keystroke recorders on a computer keyboard or key catchers consist of a device that must be inserted between the output jack of the keyboard and the computer. Unless the user is particularly expert, it is very difficult to detect such a device as it resembles a standard adapter that fits the output format of the keyboard pin to the input jack on a computer.

These devices record all the keystrokes that are made on a computer keyboard: being equipped with a good memory capacity, they are capable, depending on the degree of activity that characterises the bugged computer, of recording the keystrokes made over a few months, after which the device must be removed from the computer and an empty one will be inserted.

These devices are totally transparent to antivirus software which are not able to detect them, thereby making them very effective. Once extracted from the bugged computer, these devices are inserted into the computer's interceptor, which, by using special software, can see all keystrokes made on the bugged computer according to the day and time, optimising the bugging activities.

The only way to locate these devices is by visual inspection, even if this can be very difficult for a non-expert user, as these devices have the same features of a normal adapter and are of the same dimensions as the latter.

A variant of these devices is a spy keyboard that, while being in all respects the same as a standard keyboard, is capable of transmitting remotely, by means of electromagnetic waves, information related to each key pressed. These keyboards are much more insidious because of the pins, as the latter must be periodically replaced to read the data contained within, exposing the eavesdropper to identification, as the keyboards, once installed, are able to perform their work of interception continuously.

8.2.10 Bugging software for computers

Bugging software for computers are programs that are installed on the computer to be bugged, without the user's knowledge, and are capable of showing remotely from a desired location all the work performed on the user's bugged computer. Obviously, the bugged computer must be connected to a communication network in order to carry out this operation from a remote location. This bugging software is difficult to detect as the software has been developed not to slow down at all the computer on which it has been installed and to be undetectable by any antivirus, even by the most powerful and sophisticated ones.

Transmission of the data is, mostly, performed through standard email that does not leave any trace on the email sending program. Emails may consist of text files that contain all the characters typed on the keyboard in a given time interval that is shown in the email itself or images of the screen (screenshot) that show the eavesdropper the user's screen when the email is sent. Other data being bugged can also include sites accessed on the Internet or the programs used on a computer. While email texts do not require a large bandwidth, the files being relatively small, emails containing screenshots may be relatively large and can generate heavy traffic in the case where they are sent with a high frequency, generating a suspicious slowdown, during transmission, for an experienced user.

Such software, as has been said, are very insidious because they are difficult to detect even by advanced antivirus programs. The only recommended solution, if you suspect that your computer has been the subject of the undesired installation of these programs, requires the full recovery of the computer itself through low-level formatting of the hard disk, re-installation of the operating system, re-installation of the programs used and restoration of the data stored on your computer before the formatting.

8.2.11 Portable document scanners

Portable document scanners are characterised by their very small size (the length being the same as a standard A4 page and the width being a few millimetres).

They are used by hand to scan a sheet and to store within them a digital image of the same sheet which can be downloaded at a later date onto a computer.

Their big advantage is their ability to operate in poor lighting conditions or even in the absence of light. In this sense, they are irreplaceable compared to high-resolution cameras, which, even if are able to photograph an entire sheet very rapidly, are not able to operate properly in low-light conditions or in the total absence of light unless a flash is used, which would, however, be very visible.

They are equipped with a large storage capacity and are capable of storing, depending on the memory size, a large number of sheets.

8.3 Devices and techniques for protection against environmental bugging

There are many anti-bugging devices that are commercially available, each aimed at identifying the relative bugging device.

These devices can be relatively expensive due to the high technological content they contain.

In most cases, for conducting proper debugging, it is necessary to use multiple devices, step by step, in order to remove the presence of particularly sophisticated bugging devices.

The principal anti-bugging devices are:

1. scanners;
2. broadband bugging device detectors;
3. bugging device detectors based on cellular technology;
4. spectrum analysers;
5. multifunction spectrum analysers;
6. multifunction devices;
7. non-linear junction revealers (NLJRs);
8. hidden camera revealers;
9. electromagnetic scramblers (jammers);
10. audio scramblers (jammers);
11. telephone ciphers.

These devices are discussed below.

Regarding protection techniques against bugging, the principal one is represented by TEMPEST which is analyzed in section 8.3.15.

8.3.1 Scanners

Scanners are broadband receivers, which are able to demodulate the received signal over a relatively extensive bandwidth, depending on the characteristics of the scanner. Obviously, the cost increases as the reception band expands. Professional scanners are capable of receiving signals with frequency between tens of kilohertz and tens of gigahertz. They are able to demodulate signals modulated in Amplitude Modulation (AM) (complete or single-sideband modulation (SSB), wide Frequency Modulation (FM) and narrow FM and possibly other types of modulations.

Scanners are the basic tools for bugging detection that operate with standard modulation (AM or FM) in plain text.

In fact, if the bugging devices are of the plain text modulation type (AM, FM and SSB), they can be detected with a normal broadband scanner by placing it in the environment in question, raising the listening volume and scanning the entire available spectrum. If, at a certain point, a loud whistling sound can be heard, it means that the scanner has centred the frequency of the bugging device and the whistle is determined by triggering of the Larsen effect, due to the received signal being fed back into the environment and picked up by the bugging device transmitting it back to the scanner, activating the typical whistle due to the feedback loop described above.

The frequency at which the whistling is received corresponds to the emission frequency of the bugging device.

Once the frequency has been centred, if the scanner is equipped with a power indicator of the signal being received, the listening volume can be lowered, it being possible to move around the environment towards the direction along which there is an increase in the signal itself until reaching the maximum attainable. In this situation, you can be sure that you have arrived as close as possible to

the bugging device and it is necessary to proceed with a visual and manual inspection in that, as has been said, bugging devices are hidden in the most unlikely places.

First, it is important to emphasise that scanners are not able to locate bugging devices that operate with non-plain text modulation.

Moreover, this approach may not be optimal if the bugging device is equipped with a remote activation command. In fact, in this case, the eavesdropper can listen remotely to the type of checking activities that are being performed within the bugged environment and immediately turn off the bugging device which, not emitting further signals in radio frequency or microwave, is no longer detectable by the scanner.

However, the deactivated bugging devices can always be identified by NLJRs that are discussed in section 8.3.7.

8.3.2 Broadband bugging device detectors

Broadband bugging device detectors are, usually, small battery-powered portable devices that operate as scanners that search, analysing the whole reception spectrum, for signals transmitting with greater power. The operating frequency, in professional models, ranges from 100 kHz up to tens of gigahertz, operating in a very wide bandwidth in which, practically, bugging devices of all kinds transmit.

As bugging devices transmit with a relatively low power, the same bugging devices are received with a high-intensity within a few metres from the broadband detector and it is possible to detect them easily.

Once operated, they immediately commence searching by scanning the entire spectrum that they are capable of receiving. The simplest devices are limited to a frequency search; however, they are not equipped with a frequency meter, and the intensity of the received signal must be indicated on a display. There are more advanced models (and more expensive models) that are equipped with a frequency meter that allows the emission frequency to be read with precision.

To avoid false alarms due to the presence, around the environment to be cleared, of high-frequency sources, most of the devices are equipped with a speaker through which received signal can be heard, the same being able to demodulate, similar to scanners and in most cases, only signals transmitted in plain text (AM, FM and SSB). Once the speaker has been activated, if the typical whistle of the Larsen effect is heard, it is certain that there is a bugging device within the environment and it is necessary to proceed in a manner similar to that indicated in the previous paragraph.

These broadband detectors are in any case also able to detect bugging devices that operate with non-standard or encryption modulation, intercepting the signal with precision: the only problem is that they are not able to demodulate its transmitted signal. In any way, due to the signal strength indication shown on the monitor, it is possible to move around the room in question towards where there is an increase in the signal until reaching the maximum level received, and then to proceed, in a manner similar to the previous case, with a manual and visual inspection (Figures 8.6 and 8.7).

When using these devices, it is important to turn off all sources of electromagnetic emission in the environment, paying particular attention to computers, especially if equipped with wireless capabilities, to Wi-Fi access points, Bluetooth devices, mobile phones, cordless phones, etc. In fact, most broadband detectors are also capable of detecting the output frequencies of mobile phones, as these frequencies can be used by related bugging devices, and there may be false signals. In addition, as wireless systems operate in the ISM band, which is usually receivable by these devices, there may be a false alarm due to the omission of an activated wireless system. On the other hand, it is also very important to check the ISM band because, as mentioned, there are bugging devices and miniature cameras that work directly in this band in order to be mistaken for the other wireless devices that operate in the same band.



Figure 8.6 A portable broadband bugging device detector – front view.

8.3.3 Bugging device detectors based on cellular technology

Bugging device detectors based on cellular technology operate in a manner similar to broadband bugging device detectors and, in many cases, are integrated with them; the detectors being available on the market are capable of detecting both radio frequency or microwave bugging devices and bugging devices based on cellular technology.

These are capable of detecting any bugging devices based on various cellular technologies (GSM, GPRS, UMTS, etc.) that operate within a radius of about 10 m from the same.

Bugging devices based on cellular technology are easy to detect if they transmit because the electromagnetic emission is continuous, but are difficult to detect if they are on standby because the emission is sporadic and limited only to those moments in which the bugging device carries out monitoring communications with the base radio station nearest to the operator relative to the SIM used by the same bugging device. In the latter case, the communication is limited to a few seconds and this amount of time may not be sufficient to locate the bugging device.

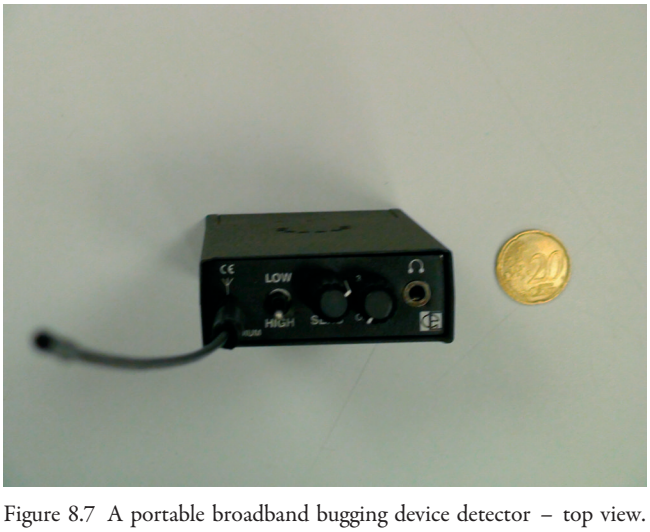


Figure 8.7 A portable broadband bugging device detector – top view.



Figure 8.8 A bugging device detector based on cellular technology.

During the debugging operation, it is absolutely essential to turn off all mobile phones to avoid triggering false alarms. The debugging operations are similar to those discussed previously.

These detectors are able to detect the emissions characterised by a short duration (burst) typical of the sending of an SMS, with particular reference to the GPS locator devices, allowing the latter to be located, on the understanding that the same may be suitably rendered harmless by special jammers which are described later.

They are also very useful to know if your mobile phone has been tampered with by the insertion of a bugging program. In this case, it is possible to perform a normal call or send an SMS and then bring the phone to the detector to see if it is engaged in the illegal activity of the sending of an SMS containing the data of the call to any interceptor (Figure 8.8).

8.3.4 Spectrum analysers

Spectrum analysers are very versatile tools, used in many environments, which allow frequencies to be read on the horizontal axis of a screen and the relative intensity on the vertical axis. Of course, it is possible to widen or narrow the horizontal axis in such a manner as to widen or narrow the scope of interest. If the range is very wide, spectral lines of various emission sources will appear on the screen whose width depends on the bandwidth of the signal being emitted. If the range is very narrow, it is possible to suitably widen the spectral line relative to the source to read the bandwidth. In this case, the spectral line is transformed into a curve with a bell-shaped characteristic, as shown in Figure 8.9.

Spectrum analysers are not easy to use because, as the bands are very crowded, it is not possible to distinguish the emission peaks of a bugging device from the emissions of other transmitters. It is possible only if the bugging device emits with sufficient intensity that the relative peak is much higher than the other peaks, unless an intense source such as a radio transmitter and video is nearby.

Once the peak is identified, if the spectrum analyser is capable of demodulation, it is possible to hear the demodulated signal directly which, if released into the environment, generates the characteristic whistle due to the Larsen effect.



Figure 8.9 Spectrum analyser showing in detail the emission – frequency – intensity curve of an electromagnetic source.

In any case, once the peak of interest has been found, it is possible to proceed with the identification of the device to see if it is caused by a bugging device present in the environment under suspicion. In this case, it is necessary to replace the omnidirectional antenna by an appropriate directional antenna, pointing it in the direction that provides the greatest signal on the screen. After that you need to move in that direction, towards the increased signal, until reaching position of maximum signal reception. Once this is done, the next step is a visual and manual inspection, as indicated previously.

The spectrum analyser is completely useless for bugging devices that operate using direct spread spectrum technology in that, first, you would have a very wide emission curve and an amplitude usually less than the background noise, which is not visible on the screen, and, second, there would be peaks in rapid motion within the frequency hopping band that would be so rapid as to not be visible to the human eye on the screen.

The same is true for burst bugging devices which, as previously mentioned, record environmental sound, compress it and then send everything together in a rapid burst. Such bugging devices, given that at the same time a spectrum analyser observes the emission band, generate an emission peak that remains on the screen as long as the emission continues and disappears once the bugging device ceases transmission.

Usually, the more expert interceptors tend to use bugging devices, whose emission frequency is very close to that of a powerful radio or video signal present in the area, in such a way that the emission peak of the bugging device is disguised by the emission peak of the other audio or video signal. These are the so-called “crouched” bugging devices. The only way to detect such bugging devices entails a long and thorough analysis of the emission spectrum, expanding suitably the band in such a way as to detect the presence of weak signals (bugging devices) side by side with stronger signals.

Detection via a spectrum analyser is not a simple task for the layman.

To make detection using spectrum analyser more efficient, long-duration spectrum analysis is often required, which can be performed by the spectrum analyser if it is equipped with this feature, or by a computer connected to the spectrum analyser if the latter has a suitable communication port. During long-duration spectrum analysis, the axes show the time and frequency while the colouring of the trace indicates the power detected, using the same colours for the signals received with the same intensity. The relative spectrogram provides very useful information to detect most radio frequency or microwave bugging devices irrespective of the emission mode.

8.3.5 Multifunction spectrum analysers

Multifunction spectrum analysers are very sophisticated devices and are, therefore, very expensive, being equipped with advanced functionality. They are usually portable devices, contained in a case that integrates the instrument.

Their frequency range extends from 10 kHz up to tens of gigahertz, being capable of detecting almost any type of bugging device, including those that use mains carriers. To be able to operate within such a range, these devices use different antennae which are integrated within a portable case in order to make the debugging process easier, and the device comes with an automatic switch according to the bandwidth analyser.

These devices coordinate the environmental noise with the electromagnetic signals that they receive searching for a link between the same environmental noise and the temporal or spectral signal trends that are gradually analysed. They are able to demodulate the majority of signals (AM long wave, AM short wave, FM long wave, FM short wave, SSB, Phase Alternating Line (PAL) video, National Television System Committee (NTSC) video, Système Electronique Couleur Avec Mémoire – Sequential Colour with Memory (SECAM) video, etc.).

They are also capable of detecting, under right conditions, the insidious direct spectrum emissions (direct expansion or frequency hopping) and burst emissions. They are equipped with memory capabilities in such a way as to record all the signals received in a given observation period for checking at a later stage using the software tools on a computer.

Because they do not emit any sound into the environment, if not required, they do not reveal their presence to the interceptor who listens to the sound being intercepted, preventing him from disabling the bugging device, if possible, to render it impossible to intercept.

It should always be in mind that, when conducting debugging, normal activities within the environment should be performed and the work that is being conducted should not be discussed to prevent the eavesdropper from realising that a debugging operation is taking place and thus taking appropriate precautions.

These devices can be equipped with a triangulation function that identifies, with precision, the position where the device in question is emitting.

The great advantage of these devices is that they can operate fully automatically: once in place, they can be activated and alone can perform detecting operations, scanning the entire reception band.

They may be left to operate continuously within the environment to be controlled in such a way as to produce a suitable spectrogram from which it is possible, at a later time, after appropriate analysis, to also identify burst-type bugging devices, characterised by an emission of short duration.

8.3.6 Multifunction devices

Multifunction devices are sophisticated devices and are, therefore, relatively expensive, which are able to integrate within a portable instrument a series of features that can ensure, by using different probes:

1. radio frequency or microwave audio and video bugging devices operating with all the major types of modulation;
2. infrared or laser beam bugging devices;
3. mains carrier bugging devices;
4. telephone bugging devices.

The use of such devices is not easy for the amateur user and, for this reason, must be used by trained professionals.

Their great advantage is their ability to detect different types of bugging devices using a single tool, subject to the previous use of appropriate probes, making the debugging work extremely easy because



Figure 8.10 A multifunction device.

you do not have to bring to the location a series of tools dedicated to each type of specific bugging device (Figures 8.10 to 8.15).

8.3.7 Non-linear junction detectors

Non-linear junction revealers (NLJR) are an essential tool for detecting any kind of electronic devices, even if these are turned off, because their operation is not based on the monitoring of the emission of the apparatuses, rather on appropriate signals emitted by themselves which stimulate the electronic devices to react with a response emission that is analysed by the detectors in order to assess the presence or absence of electronic devices.

It is clear that the use of these devices must be limited to experts in the field, who are aware of the advantages and disadvantages of such devices as these are equipped with an alarm to detect any electronic device that may not necessarily be a bugging device and also equipped with an alarm for oxidised metal objects that have a behaviour similar to non-linear junctions that are present in the electronic components. In this sense, a good detection device must be able to discriminate between a non-linear junction belonging to an electronic component and a non-linear junction caused by oxidised metal.

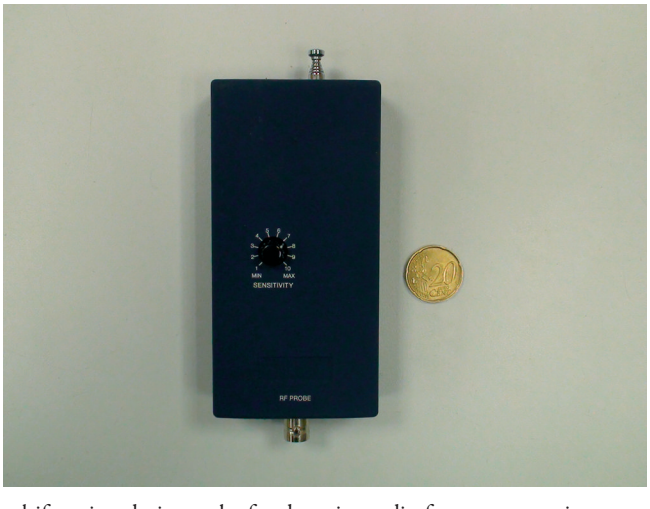


Figure 8.11 A multifunction device probe for detecting radio frequency or microwave bugging devices.



Figure 8.12 A multifunction device probe for detecting infrared or microwave bugging devices.

NLJRs are equipped with an integrated antenna that emits a suitable electromagnetic signal that interacts with every object located within its range of capability. If this signal detects a non-linear junction (such as those contained in electronic components, e.g., transistors and microchips), it is reissued from the junction itself together with its higher level harmonics, precisely due to the non-linearity of the junction. This non-linear behaviour also occurs when two metals with different characteristics, suitably oxidised, are placed in contact with each other. Because, in both cases, the non-linear behaviour is different, if the instrument is able to analyse the harmonic content of feedback, as will be shown later, the instrument itself can differentiate between an electronic component and a simple oxidised metal.

The voltage–current characteristics of the non-linear junction of a semiconductor and a non-linear junction caused by two oxidised conductors coming into contact are shown in Figure 8.16.

As can be seen in the figure, the two characteristics, the voltage and the current, have a very similar behaviour in the first quadrant but a totally different behaviour in the third quadrant.



Figure 8.13 A multifunction device probe for detecting miniature cameras or mains carrier bugging devices.



Figure 8.14 A multifunction device probe for detecting hidden microphones.

To distinguish between the two types of junctions, the second and third harmonics are mostly taken into consideration, which are, in turn, generated by a signal emitted by NLJR: if the intensity of the second harmonic is higher than the intensity of the third harmonic, then there is the presence of a non-linear junction due to the presence of an electronic component while if the opposite is the case, then it is a matter of a false junction due to the contact of two oxidised metals.

The ability to analyse the third harmonic requires the presence of separate receivers for both the second and third harmonics, providing in any case superior performance at a higher cost, given the greater complexity of the electronics present in the NLJR. These receivers must be carefully separated, to discriminate with precision, between the intensity of the second harmonic and that of the third harmonic, giving the NLJR a high capacity for discrimination and a reduced number of false alarms.

To improve the analytical capacity of NLJR, the main signal emitted is, in most cases, frequency-modulated and the return signal is demodulated by producing an audio signal that is very useful for bugging operations. Very often, in fact, when an alarm signal is received due to the presence of a non-linear junction created by oxidised metals within a wall, if the wall is hit, a typical interruption of the



Figure 8.15 A multifunction device probe for analysing telephone lines for detecting relative bugging devices.

audio can be heard due to the fact that the junction itself is interrupted as a result of the impact while the audio signal continues to be the same if the junction is genuine and, therefore, does not stop with a simple hit.

Most of the devices can emit the signal continuously or in pulses by modulating both in FM and AM. Pulsed mode is used to reduce the power consumption and, consequently, to lengthen the life of the battery that powers the devices.

Because NLJRs operate in radio frequency and some frequencies may be already occupied by other transmitters, which continuously generate false alarms, it is very important that these devices are capable of operating on multiple frequencies in such a way as to find easily those free from disturbances in the areas to be debugged.

Correct balance between emission power and reception sensitivity is also very important. In fact, as can be seen in Figure 8.16, most non-linear behaviour of semiconductor junctions is obtained from low voltage levels, where the curve is more non-linear, and a localised linear behaviour is assumed by the curve itself from high levels of both direct and reverse voltages. This means that the power of the emitted signal to stimulate the junction to emit harmonics must be relatively low in order to obtain the maximum harmonic response. As the power is reduced, the receiver must be very sensitive to pick up the returned harmonics. For this reason, it is preferable to have a low-power transmitter combined with a very sensitive receiver rather than a very powerful transmitter combined with a low-sensitivity receiver.

The best devices are equipped with an automatic gain switch which automatically reduces the gain when the receiver tends to saturate due to the vicinity of the NLJR to junctions, while it increases when the NLJR is moved away from junctions.

The detection distance is a crucial factor. It is obvious that for the NLJR to achieve high detection distances, it must emit with considerable power, but this can be counter-productive as high emission powers tend to saturate the non-linear junctions that are within close range, thus compromising the results. For this reason, the NLJRs need not reach high distances and, in any case, the detection distance must be variable and controlled automatically by the instrument to always ensure the optimum detection of non-linear junctions.

The ergonomics of the instrument are very important. In fact, the best instruments are those that are relatively light and allow direct reading of the graphical results on the probe in terms of power output, the second harmonic signal received and the third harmonic received. In this way, the

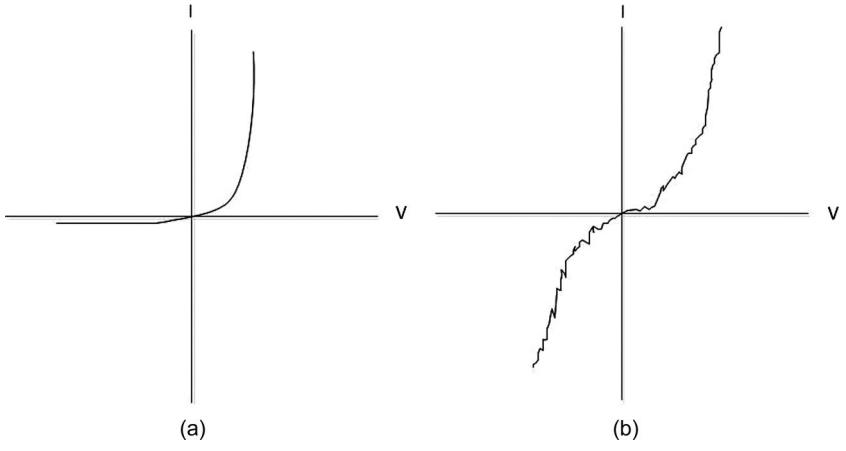


Figure 8.16 The voltage–current characteristics of the non-linear junction of a semiconductor (a) and a non-linear junction caused by two oxidised conductors coming into contact (b).

exploration of a large environment is relatively easy due to both the light weight to be carried and the ability to read directly the relevant parameters on the probe for the purposes of detection.

8.3.8 Hidden miniature camera detectors

Hidden miniature camera detectors are portable, battery-powered devices that reveal the physical presence of a camera within an environment. They are equipped with a viewer through which the environment must be observed. Because of appropriate lighting techniques, as soon as the lens of a camera is illuminated, it appears suitably coloured and brilliant looking through the viewer.

Its detection distance is a few metres.

To improve the effect of detection, it is recommended that the brightness of the environment is reduced as much as possible in such a way that the camera lens (which may be particularly small if the camera is properly hidden) appears as a bright point when looking through the detector lens.

The whole environment must be carefully monitored with such a device, paying particular attention to apparatuses such as clock radios, clocks and similar objects, perhaps received as a gift, within which a camera may be hidden.

As has been said, miniature camera detection is a physical exercise because the same is seen through the viewer as a brilliant point.

Obviously, if the same operates in radio frequency or microwave, it is always possible to detect it with the devices discussed previously, under right conditions. Figure 8.17 shows a typical miniature camera detector.

8.3.9 Wireless remote camera detectors

Wireless remote camera detectors are portable devices, equipped with liquid crystal display (LCD) screen, which detect the presence of wireless cameras that operate in the ISM band. They promptly show the image of the wireless camera and the relative signal level. From the received image and signal level, it is possible to trace back immediately to the positioning of these within the bugged environment.

They are very easy to use as they need simply to be turned on in the environment to be debugged and, if there is a wireless camera, the framed image is shown on the device's LCD screen. Figure 8.18 shows an image of a wireless camera detector.



Figure 8.17 A miniature camera detector.



Figure 8.18 A wireless camera detector.

8.3.10 Electromagnetic jammers

Electromagnetic jammers are special devices that emit appropriate sequences of electromagnetic disturbance over the frequencies used by detection devices that operate with a well-established technology in order to "blind" them and make communications impossible.

It should be remembered that, generally, the law expressly states that damage or disruption of electronic communications is forbidden, and for this reason, these devices are only discussed for the sake of completeness of the topic.

Because they can only cover limited environments, their emission power is low and consequently well below the exposure limits of humans to electromagnetic fields.

Depending on what they are used for, they are able to disable all communication of mobile phones based on all technologies, Wi-Fi or Bluetooth devices.

They are very useful in that they render ineffective all bugging devices that use cellular technology or any hazardous devices that can be activated remotely by mobile phones (bomb detonators).

They are also very useful when wishing to terminate communications within an environment for security reasons.

They also jam GPS signals, which "blind" any trackers installed in your car. They are very small in size, which means that they can be inserted directly into the cigarette lighter socket, through which they are charged. Their emission power is very limited, given the high sensitivity of GPS receivers, and as such they do not show exposure risk to individuals of electromagnetic fields.

8.3.11 Jammers for audio devices

Jammers for audio devices are of two types:

1. audible frequency jammers;
2. inaudible frequency jammers.

Audible frequency jammers generate sounds which introduce white noise into the environment capable of rendering useless any recorders or microphone bugging devices present in the environment. They practically saturate the microphones in the environment, rendering them incapable of recording voices. Not even using voice analytical software, it is possible to extract the voice sounds picked up by microphones.

Inaudible frequency jammers, on the other hand, use subsonic frequencies. They enable the capture ability of any microphones present in the environment, saturating them. They tend not to work correctly if the recorders are enclosed in metal casing.

8.3.12 Jammers for laser beam bugging devices

We have seen previously that if you point a laser beam at a glass perimeter of a room, this beam is modulated by the vibrations of the glass due to the capture of environmental sounds. If you are able to demodulate the partially reflected laser beam, it is possible to intercept the environmental sounds.

To avoid this, you can resort to installing reinforced glass, characterised by a high resistance to the transmission of sounds and vibrations, but this installation can be particularly expensive if the glazed surfaces are very thick.

A more economical solution is available with acoustic jammers, which are applied directly to the glass. These jammers, characterised by their small size, are powered by a white noise generator. Under the influence of the generator, the scramblers generate vibrations, which propagate along the entire glazed surface to which they are applied, which overlap, disguised by the vibrations produced by the environmental sounds, thus nullifying the effect of the bugging by means of a laser beam focused from the outside.

8.3.13 Encrypted phones

Encrypted phones, in fact, use cryptography to secure communications when using non-secure channels of communication used by normal telephone networks. Obviously, both users must have the same apparatus and must have exchanged the encryption codes, using a secure channel. The best way to perform the exchange is by meeting in person, if possible, the person with whom you wish to communicate. It is also highly advisable to change the encryption code frequently, although this may be difficult if the other person is far away. In this sense, some advanced apparatuses use published encryption codes for the prior exchange of the encryption key. In this way, you are sure to change the key for every communication, maximising the highest level of security for these communications.

Telephones can be of the following types:

1. landline;
2. cellular;
3. fax;
4. chat and video chat programs.

Regarding landlines, these are standard apparatuses that come with all the electronics needed to encrypt communications which mostly occur by digitising the voice signal and using the most secure digital encryption algorithms. The encryption system used may be secret or published key type.

Concerning mobile phones, these are standard apparatuses in which a suitable encryption software is installed, which needs to be activated when an encrypted call is to be performed. Also in this case, the encryption system may be secret or private published key.

Regarding faxes, these are standard apparatuses integrated with all the electronics necessary to encrypt the signal that they send over telephone lines. Their size is practically the same as normal faxes.

Regarding chat and video chat programs, there are numerous software programs that allow messages to be exchanged or to make a voice or video call via a standard computer. These programs, given the inadequate security of the Internet, use encryption systems that vary from program to program. Using these programs, which are available mostly free of charge on the Internet and can be used at zero cost, we have a perfect secure communication system for text, voice or video..

Is it also possible to install these chat and video chat programs on the mobile phone: in this case, an encrypted system of secure communication is obtained at zero cost.

8.3.14 Software utilities

There are several software utilities that can be used for protection against bugging.

A top utility is hard drive formatters. In fact, very often, computers are discarded without first erasing the data contained on the hard drives, exposing them to a high risk of loss of sensitive or confidential data. Even if executing standard formatting, this process does not delete the files on disk, but reorganises the disk in such a way as to delete the associations to these files, which are always present but can be overwritten. In this sense, formatters take the trouble to erase and overwrite the entire hard drive several times to be certain that there is no more trace of any data. The use of such software is highly recommended before the disposal of any computer on which sensitive activities have been carried out.

Another function is represented by the generators and operators of encrypted virtual drives. These functions create virtual encrypted drives on the hard drive which, in many cases, are not visible to the normal user, further protecting the data contained on them. They are very useful even in the case where, despite the fact that the network protections have been applied as shown previously, a virus or a Trojan horse infects the computer. If virtual encrypted drives are used, the data contained on them are secure because these drives are not accessible; if the virus or Trojan horse were able to access them, they would find the encrypted data impossible to access.

8.3.15 TEMPEST

Every electronic device, when it is running, emits electromagnetic fields which, if intercepted and processed, could, in theory, be a source of considerable loss of information. In practice, any device using microchips, transistors, diodes or any other electronic component emits electromagnetic waves. These inadvertent emissions can pose a serious threat from the viewpoint of interception in that they can escape from an environment not only by emissions into space but also through inductive paths that can include electrical lines and air conditioning pipes.

This problem becomes particularly critical when we have to work with computers, faxes, monitors, external hard drives, compact disc (CD) and digital video disc (DVD) players, scanners, printers, etc., that embody devices operating at high frequencies and bandwidths and, therefore, produce a high amount of electromagnetic waves that radiate out into space.

If an eavesdropper is adequately equipped with the appropriate instrumentation, he can receive such emissions from a considerable distance from environments where there are electronic devices. He could record these emissions and then analyse them, patiently, later with appropriate analytical tools.

In a standard computer that performs standard operations, these inadvertent emissions do not pose a risk, but on a computer that runs confidential or classified operations, these represent a serious risk and are a dangerous source for the loss of information.

Consequently, international rules and standards have been issued describing how devices or environments should be set up to avoid these inadvertent losses. These are simply a series of standard guidelines that were then checked and verified by the US National Security Agency (NSA).

In this sense, a computer is built according to these standards and is equipped with a special container made of heavy metal, proper shielding, a suitable power supply system that prevents the on-board electronic emissions being propagated via the power supply line, together with a number of other changes designed to eliminate inadvertent emissions. These modifications, of course, make the computer more expensive than a standard computer that does not comply with these requirements.

The best standard in this area is Transient Electromagnetic Pulse Emanation Standard (TEMPEST). It includes technical security countermeasures; standards and instrumentation required to prevent, or minimise, the presence of security risks using technical means. TEMPEST is only a reference for the protection against bugging.

There are, however, other unofficial names for TEMPEST including Transient Emanations Protected from Emanating Spurious Transmissions, Transient Electromagnetic Pulse Emanation Standard and many others.

The TEMPEST concept was conceived in 1918 by Herbert Yardley and his staff, who were recruited by the United States Army to develop methods to detect, intercept and counteract enemy transmitters. Initial research discovered that standard unmodified devices allowed the leakage of classified information to potential eavesdroppers. As a result, a series of techniques was developed to suppress these inadvertent emissions. In any event, the acronym TEMPEST was coined in the late 1960s and the early 1970s and was replaced, however, by the term "EMSEC" (emission security).

TEMPEST and related science also include the design of electronic circuits to reduce inadvertent emissions and defects such as shields and earthing. This science also includes methods for the screening of electromagnetic waves, alarms, insulation circuits, etc.

TEMPEST involves the elimination or reduction of the transients generated by signals used in communications and the resulting harmonics. These signals, and the related harmonics, radiating into space, allow a potential eavesdropper to pick up the signal, reconstruct it and analyse it.

A device that complies with the TEMPEST standard is constructed according to standard technical guidelines. In this way, electromagnetic emissions have been reduced through appropriate techniques of shielding or by other technologies. This reduction operation has been accomplished to such an extent that it is virtually impossible for an eavesdropper to acquire confidential information through inadvertent emissions.

There are several TEMPEST classification levels depending on specific applications and the required security level.

While TEMPEST represents all those standards aimed at the reduction or elimination of inadvertent emissions by using technologies such as shielding and earthing, Signal Intelligence (SIGINT) represents all those technologies aimed at the interception and analysis of inadvertent compromising emissions.

In 1985, Dutch engineer Wim van Eck published an article that covered the potential methods for intercepting video monitors. The corresponding receiver that was conceived was designed to intercept old video monitors that utilised a composite video signal without resorting to suitable shielding of electromagnetic emissions. This video signal is of baseband type and is characterised by a relatively broad bandwidth that causes a high inadvertent emission.

Van Eck's article aroused great interest among the research community and stimulated research in this area. Obviously, the results achieved then, with the monitors used at that time, were characterised by a high electromagnetic emission and are not currently achievable with existing monitors characterised by ultra-low emissions. What van Eck presented is currently known as raster analysis or raster analysis and identification (RAID) that constitutes the reconstruction of composite broadband signals, or raster, which is based on repetitive signal synchronisation. Raster analysis, or RAID, is usually performed during the debugging of an environment to determine if a raster signal present in the environment is related to a potential bugging device. Raster analysis is used mainly for the evaluation of the modulated video signals such as those generated by television transmitters or by miniature video cameras using radio frequency or microwave. In this sense, the emissions that occur with a computer monitor are not modulated, being the same as baseband signals. Baseband signals tend to be very strong near the monitor and computer and are essentially covered by three fields:

1. magnetic fields;
2. electric fields;
3. electromagnetic fields.

Electric and magnetic fields are independent of each other in the vicinity of the monitor, in the so-called near field region, although they tend to merge in the form of electromagnetic fields that propagate outside the near field region, where the far field region begins. Obviously, the greater the distance from the monitors and the smaller the amplitude of the fields, the greater the need to use antennae having high gain and very sensitive receivers.

To avoid these problems, a low-emission monitor is used along with various active shielding devices available on the market.

In many cases, in order to have an environment with a high level of security, privacy and confidentiality, it is necessary to perform a series of operations, such as the protection of walls, ventilation ducts and screens, the filtering and earthing of electrical systems and telephone systems, within the environment and to have appropriate active shielding devices that serve to make the environment TEMPEST compliant. The cost for this operation can be relatively high but this allows an environment vulnerable to bugging, with any one of the devices or technologies seen so far, to be virtually impenetrable, making it extremely secure for conversation and business conducted within.

8.4 Procedures and guidelines for suspected environmental bugging

When there are genuine suspicions for being bugged, rash attitudes should be entirely avoided along with discussion of the matter by voice or by phone in the suspected environments.

The first thing to do, if the suspicions are justified, is to contact the police explaining the reasons for contacting them. The police will take all the necessary steps.

If you wish to contact a private party, it is absolutely necessary to be certain about their professionalism, and technical and management skills on the matter that may be an individual or a specialist company. Below, this party will be referred to as the debugger.

Unfortunately, there are no specialist registers for this and the best thing is to always contact the police to acquire the names of debuggers with specialised and proven expertise. Care must always be taken when searching on sources such as the Internet because, it not being appropriate, for confidentiality reasons, to advertise their references or the names of the customers that have benefited from their services, this does not necessarily ensure the professionalism and technical expertise of the debugger.

If necessary, professional associations specialising in the security field can be contacted who can recommend some names of their members, who have been registered for some time and, therefore, well known to the associations who operate in the environmental debugging sector.

Significant information on the debuggers can include their professional background, their expertise in the sector and, where appropriate, their annual turnover.

Once a suitable debugger has been found, contacting them directly from the phone in the environment under suspicion or from your mobile phone should be avoided at all costs as these could be under surveillance.

In this case, it is recommended that the first contact should be through a public phone, which is increasingly difficult to find, or through a landline or mobile phone which is definitely not under surveillance.

Looking for debuggers from the same town should be avoided so as not to raise suspicions when they arrive at your place to perform debugging operations or at least not from the same neighbourhood if you are in a big city.

The situation should not be revealed directly over the phone; instead, it is advisable to arrange a meeting at the debugger's premises, better yet, in a random place, preferably outdoors and in public.

To help the debugger understand the situation better, it is necessary to provide in advance some useful information such as a phone number from which it is safe to be contacted with; the address of local suspects, details useful for the description of the environments, such as what premises are equipped with telephones and the positioning of the telephone switchboard, if present; any plans for the suspect environments including the furniture present; the presence or absence of a false ceiling; a description of the electrical system, the telephone system and the computer network; a list of telephone users, including mobile phones, for the past few months, together with any bills and any detailed call records; and information about the neighbours of local suspects if they are in the same building.

You should be wary of debuggers who arrive with cars covered with garish advertisements for the industry, as this type of business must be conducted in the most sober and reserved manner possible.

Once the debuggers are in possession of this information, they may deem it sufficient and can prepare an estimate of costs or they may require a preliminary inspection that must be in the late afternoon or evening, if these are offices, in order to be sure that all employees are outside the premises in question and may not suspect the commencement of debugging operations, seeing unknown persons roaming the premises with all the necessary equipment for such operations.

It is good to acquire information about how to perform the debugging operation and the type of tools that will be used, from which you may already have an idea of the technical and instrumental skills, having acquired a plenty of technical information after reading this chapter.

Regarding costs, it is better to agree these beforehand to avoid surprises. Obviously, the greater the degree of expertise and experience of the debugger, along with the quality and quantity of debugging devices used, the higher the cost. On the other hand, even security and confidentiality have a cost and that cost must be commensurate with the cost of the loss and theft of confidential information due to possible surveillance. Usually, costs may be charged on an hourly or daily basis (in this case, it is necessary to agree beforehand to the duration of the intervention) or it can be per room or even per square metre of the area and a large proportion can be incurred depending on the number of telephone apparatuses present.

Once the fee for the debugging operation has been agreed, the work can commence. As already stated, the operation must be performed in the late afternoon or evening to be sure that all staff have left and in order not to raise suspicions which would result in the eavesdropper removing the bugging device, if possible, or removing it and re-installing it at a later time.

Depending on the size of the areas to be cleaned, and on the limited time available, as one should work, as already stated, in the late afternoon/evening/night, the operation can be carried out in one session, or it may be necessary to perform several times.

As bugging devices can operate in burst mode, it may necessary for the debugger to leave the instruments for the whole night, retrieving them early on the following morning before the arrival of employees.

In many cases, the debugger can perform the debugging over the weekend in order to have considerable uninterrupted amount of time to perform all the debugging operations required.

In addition to the normal instrument checks, which can occur step by step or in parallel, it may be necessary to remove and inspect panels, false ceilings, electrical sockets, electrical junction boxes, telephone junction boxes, data jack boxes, ceilings, electronic and electrical devices, etc.

It is very important, as has already been said above, to avoid speaking about the type of activity in progress as such conversations could be intercepted by the eavesdropper who could take all the necessary precautions to avoid being discovered. In this case, the debugger will advise about the type of behaviour and the type of environmental noise to be generated, for the best results for the debugging operation.

If the debugger finds a bugging device, it is imperative not to touch anything, to leave everything as it is and to immediately notify the police who will carry out all procedures for the case.

Obviously, in both situations where a bugging device is found or where nothing is found, the environment is only secure at that given moment, as a prospective eavesdropper could install a bugging device the following day. For this reason, if you believe that your work is susceptible to bugging operations, it is necessary to perform intermittent debugging operations.

In extreme cases, you can buy a multifunction device that is not particularly high on performance and learn how to use it to perform low-level and frequent debugging. In doing so, the risk of surveillance is reduced; however, it is entirely eliminated when a debugging operation is performed by a qualified debugger, with extensive technical and professional expertise and with top-class professional equipment.

This page intentionally left blank

BIBLIOGRAPHY

Chapter 1

A. Aragón-Zavala & S. Saunders, *Antennas and Propagation for Wireless Communication Systems*, Wiley, 2007.
A. F. Molisch, *Wireless Communications*, Wiley, 2011.
A. Gokhale, *Introduction to Telecommunications*, Delmar Cengage Learning, 2004.
A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.
A. Luzzatto & G. Shirazi, *Wireless Transceiver Design: Mastering the Design of Modern Wireless Equipment and Systems*, Wiley, 2007.
A. M. Noll, *Introduction to Telecommunication Electronics*, Artech, 1995.
A. R. Mishra, *Fundamentals of Cellular Network Planning and Optimisation: 2G/2.5G/3G. Evolution to 4G*, Wiley, 2004.
A. S. Tanenbaum & D. J. Wetherall, *Computer Networks*, Prentice Hall, 2010.
A. Z. Dodd, *Essential Guide to Telecommunications*, Prentice Hall, 2005.
B. Dunsmore & T. Skandier, *Telecommunications Technologies Reference*, Cisco Press, 2002.
B. Forouzan & F. Mosharraf, *Computer Networks: A Top-Down Approach*, McGraw-Hill, 2011.
B. Sosinsky, *Networking Bible*, Wiley, 2009.
B. S. Davie, A. Farrel, L. L. Peterson & P. Zheng, *Wireless Networking Complete*, Morgan Kaufmann, 2009.
B. S. Davie & L. L. Peterson, *Computer Networks: A Systems Approach*, Morgan Kaufmann, 2011.
B. S. Manoj & C. S. R. Murthy, *Ad Hoc Wireless Networks: Architectures and Protocols*, Prentice Hall, 2004.
C. Andersson, *GPRS and 3G Wireless Applications: Professional Developer's Guide*, Wiley, 2001.
C. Cox, *Essentials of UMTS*, Cambridge University Press, 2008.
C. J. Weisman, *Essential Guide to RF and Wireless*, Prentice Hall, 2002.
C. N. Herrick, *Telecommunications Wiring*, Prentice Hall, 2000.
C. R. Nassar, *Telecommunications Demystified*, Newnes, 2001.
C. Sayre, *Complete Wireless Design*, McGraw-Hill Professional, 2008.
C. Smith, *3G Wireless Networks*, McGraw-Hill Osborne Media, 2006.
C. Stross, *Wireless*, Ace Hardcover, 2009.
D. E. Comer, *Computer Networks and Internet*, Prentice Hall, 2008.
D. M. Dobkin, *RF Engineering for Wireless Networks: Hardware, Antennas, and Propagation*, Newnes, 2004.
D. M. Pozar, *Microwave and RF Design of Wireless Systems*, Wiley, 2000.
D. P. Agrawal & Q. A. Zeng, *Introduction to Wireless and Mobile Systems*, CL-Engineering, 2010.
E. Hossain, D. I. Kim & V. K. Bhargava, *Cooperative Cellular Wireless Networks*, Cambridge University Press, 2011.
E. Levy-Abegnoli, P. Grossetete & C. Popoviciu, *Deploying IPv6 Networks*, Cisco Press, 2006.
E. McCune, *Practical Digital Wireless Signals*, Cambridge University Press, 2010.
E. Perahia & R. Stacey, *Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11n*, Cambridge University Press, 2008.

F. C. Berry, B. A. Black, P. S. Di Piazza, B. A. Ferguson, & D. R. Voltmer, *Introduction to Wireless Systems*, Prentice Hall, 2011.
F. J. Derfler & L. Freed, *How Networks Work*, Que, 2004.
G. Gomez & R. Sanchez, *End-to-End Quality of Service Over Cellular Networks: Data Services Performance Optimization in 2G/3G*, Wiley, 2005.
G. Heine, *GSM Networks: Protocols, Terminology and Implementation*, Artech, 1998.
G. J. Mullett, *Basic Telecommunications: The Physical Layer*, Delmar Cengage Learning, 2002.
G. J. Mullett, *Wireless Telecommunications Systems and Networks*, Delmar Cengage Learning, 2005.
H. Karl & A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, Wiley-Interscience, 2007.
H. L. Bertoni, *Radio Propagation for Modern Wireless Systems*, Prentice Hall, 2000.
H. Newton, *Newton's Telecom Dictionary: Telecommunications, Networking, Information Technologies, the Internet, Wired, Wireless, Satellites and Fiber*, Flatiron Publishing, 2011.
H. R. Anderson, *Fixed Broadband Wireless System Design*, Wiley, 2003.
IEEE Communications Society, *A Guide to the Wireless Engineering Body of Knowledge*, Wiley-IEEE Press, 2009.
I. F. Akyildiz & M. C. Vuran, *Wireless Sensor Networks*, Wiley, 2010.
I. F. Akyildiz & X. Wang, *Wireless Mesh Networks*, Wiley, 2009.
I. Poole, *Cellular Communications Explained: From Basics to 3G*, Newnes, 2006.
J. Cache, J. Wright, & V. Liu, *Hacking Exposed Wireless*, McGraw-Hill Osborne Media, 2010.
J. Carr, *The Technician's Radio Receiver Handbook: Wireless and Telecommunication Technology*, Newnes, 2001.
J. D. McCabe, *Network Analysis, Architecture, and Design*, Morgan Kaufmann, 2007.
J. F. Kurose & K. W. Ross, *Computer Networking: A Top-Down Approach*, Addison Wesley, 2009.
J. G. Van Bosse & F. U. Devetak, *Signaling in Telecommunication Networks*, Wiley-Interscience, 2006.
J. Geier, *Deploying Voice Over Wireless LANs*, Cisco Press, 2007.
J. Geier, *Designing and Deploying 802.11n Wireless Networks*, Cisco Press, 2010.
J. H. Green, *The Irwin Handbook of Telecommunications*, McGraw-Hill, 2005.
J. Heiskala & J. Terry, *OFDM Wireless LANs: A Theoretical and Practical Guide*, Sams, 2001.
J. Highhouse, *Guide to Telecommunications Cable Splicing*, Delmar Cengage Learning, 2008.
J. Leary & P. Roshan, *802.11 Wireless LAN Fundamentals*, Cisco Press, 2010.
J. R. Pierce & A. M. Noll, *Signals: The Science of Telecommunications*, W H Freeman & Co, 1990.
J. Ross, *The Book of Wireless: A Painless Guide to Wi-Fi and Broadband Wireless*, No Starch Press, 2008.
J. Ross, *Network Know-How: An Essential Guide for the Accidental Admin*, No Starch Press, 2009.
J. S. Edwards & G. S. Rogers, *Introduction to Wireless Technology*, Prentice Hall, 2003.
J. Shen & L. Song, *Evolved Cellular Network Planning and Optimization for UMTS and LTE*, CRC Press, 2010.
J. Ugarkar, *The Essentials of Telecommunications Management: A Simple Guide to Understanding a Complex Industry*, AuthorHouse, 2010.
J. Unger, *Deploying License-Free Wireless Wide-Area Networks*, Cisco Press, 2003.
J. Wells, *Multi-Gigabit Microwave and Millimeter-Wave Wireless Communications*, Artech, 2010.
K. Iniewski, *Convergence of Mobile and Stationary Next-Generation Networks*, Wiley, 2010.
K. D. Wong, *Fundamentals of Wireless Communication Engineering Technologies*, Wiley, 2012.
L. Harte, *Wireless Technology Basics, Signals, Modulation Types, and Access Technologies*, Althos, 2004.
M. Ciampa & J. Olenewa, *Wireless# Guide to Wireless Communications*, Course Technology, 2006.
M. Cole, *Introduction to Telecommunications: Voice, Data, and the Internet*, Prentice Hall, 2001.
M. Gast, *802.11 Wireless Networks: The Definitive Guide*, O'Reilly Media, 2005.
M. Gerla & D. Raychaudhuri, *Emerging Wireless Technologies and the Future Mobile Internet*, Cambridge University Press, 2011.
M. Newman, *Networks: An Introduction*, Oxford University Press, 2010.
M. Rosengrant, *Introduction to Telecommunications*, Prentice Hall, 2006.
M. Rupp, *Video and Multimedia Transmissions Over Cellular Networks: Analysis, Modelling and Optimization in Live 3G Mobile Networks*, Wiley, 2009.
M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Prentice Hall, 1987.
M. Subramanian, *Network Management: Principles and Practice*, Addison Wesley, 1999.
M. S. Obaidat & N. A. Boudriga, *Fundamentals of Performance Evaluation of Computer and Telecommunications Systems*, Wiley-Interscience, 2010.
M. W. Lucas, *Network Flow Analysis*, No Starch Press, 2010.
N. Allen, *Network Maintenance and Troubleshooting Guide: Field Tested Solutions for Everyday Problems*, Addison-Wesley Professional, 2009.
N. Blaunstein, *Radio Propagation in Cellular Networks*, Artech, 2000.
N. Bulusu & S. Jha, *Wireless Sensor Networks: A Systems Perspective*, Artech, 2005.
N. Hunn, *Essentials of Short-Range Wireless*, Cambridge University Press, 2010.

N. J. Muller, *Wireless A to Z*, McGraw-Hill Professional, 2002.
P. Gralla & E. Lindley, *How Wireless Works*, Que, 2005.
P. Oppenheimer, *Top-Down Network Design*, Cisco Press, 2010.
P. Viswanath & D. Tse, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.
P. M. Shankar, *Introduction to Wireless Systems*, Wiley, 2001.
R. Aiello, D. Bensky, P. Chandra, D. M. Dobkin, F. Dowla, D. Lide, B. A. Fette, D. B. Miron & R. Olexa, *RF & Technologies: Know It All*, Newnes, 2007.
R. A. Gershon, *Telecommunications and Business Strategy*, Routledge, 2008.
R. Blake & L. Chartrand, *Wireless Communication Technology*, Delmar Cengage Learning, 2000.
R. Frobenius & A. W. Scott, *RF Measurements for Cellular Phones and Wireless Data Systems*, Wiley-IEEE Press, 2008.
R. Horak, *Telecommunications and Data Communications Handbook*, Wiley, 2008.
R. K. Ahuja, T. L. Magnanti, & J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993.
R. K. Rao & G. Radhamani, *WiMAX: A Wireless Technology Revolution*, Auerbach Publications, 2007.
R. L. Freeman, *Telecommunication System Engineering*, Wiley-Interscience, 2004.
R. L. Freeman, *Fundamentals of Telecommunications*, Wiley-IEEE Press, 2005.
R. M. Young & M. Y. Rhee, *CDMA Cellular Mobile Communications and Network Security*, Prentice Hall, 1998.
R. Marg, J. Smith & J. Woodhams, *Controller-Based Wireless LAN Fundamentals: An End-to-End Reference Guide to Design, Deploy, Manage, and Secure 802.11 Wireless Networks*, Cisco Press, 2010.
R. Panko & J. Panko, *Business Data Networks and Telecommunications*, Prentice Hall, 2010.
R. Price, *Fundamentals of Wireless Networking*, Career Education, 2004.
S. Gezici, I. Guvenc, U. C. Kozat, & Z. Sahinoglu, *Reliable Communications for Short-Range Wireless Systems*, Cambridge University Press, 2011.
S. Gibilisco, *Handbook of Radio and Wireless Technology*, McGraw-Hill Professional, 1998.
S. Haykin & M. Moher, *Modern Wireless Communications*, Prentice Hall, 2004.
S. Hooda, S. McFarland, M. Sambi, & N. Sharma, *IPv6 for Enterprise Networks*, Cisco Press, 2011.
S. R. Chalup, C. J. Hogan, & A. Limoncelli, *The Practice of System and Network Administration*, Addison-Wesley Professional, 2007.
T. Dean, *Guide to Telecommunications Technology*, Course Technology, 2002.
T. Dean, *Network + Guide to Networks*, Course Technology, 2009.
T. Plevyak & V. Sahin, *Next Generation Telecommunications Networks, Services, and Management*, Wiley-IEEE Press, 2010.
T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, 2002.
V. Garg, *Wireless Communications & Networking*, Morgan Kaufmann, 2007.
W. C. Lindsey & M. K. Simon, *Telecommunication Systems Engineering*, Dover Publications, 2011.
W. Dargie & C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*, Wiley, 2010.
W. Goralski, *The Illustrated Network: How TCP/IP Works in a Modern Network*, Morgan Kaufmann, 2008.
W. Lee, *Wireless and Cellular Telecommunications*, McGraw-Hill Professional, 2005.
W. Lewis, *LAN Switching and Wireless, CCNA Exploration Companion Guide*, Cisco Press, 2008.
W. Soyinka, *Wireless Network Administration: A Beginner's Guide*, McGraw-Hill Osborne Media, 2010.
W. Stallings, *Wireless Communications & Networks*, Prentice Hall, 2004.

Chapter 2

A. Bruen & M. A. Forcinito, *Cryptography, Information Theory, and Error-Correction: A Handbook for the 21st Century*, Wiley-Interscience, 2004.
A. G. Konheim, *Computer Security and Cryptography*, Wiley-Interscience, 2007.
A. J. Elbirt, *Understanding and Applying Cryptography and Data Security*, Auerbach Publications, 2009.
A. McAndrew, *Introduction to Cryptography with Open-Source Software*, CRC Press, 2011.
A. Menezes, P. Van Oorschot & S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.
A. Stanoyevitch, *Introduction to Cryptography with Mathematical Foundations and Computer Implementations*, Chapman and Hall/CRC, 2010.
A. W. Dent & C. J. Mitchell, *User's Guide to Cryptography and Standards*, Artech, 2004.
B. Forouzan, *Cryptography & Network Security*, McGraw-Hill Science/Engineering/Math, 2007.
B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Wiley, 1996.
C. Boyd & A. Mathuria, *Protocols for Authentication and Key Establishment*, Springer, 2010.
C. Kollmitzer & M. Pivk, *Applied Quantum Cryptography*, Springer, 2010.

C. Paar, J. Pelzl, & B. Preneel, *Understanding Cryptography: A Textbook for Students and Practitioners*, Springer, 2010.
C. Swenson, *Modern Cryptanalysis: Techniques for Advanced Code Breaking*, Wiley, 2008.
C. A. Deavours & L. Kruh, *Machine Cryptography and Modern Cryptanalysis*, Artech, 1985.
D. Bishop, *Introduction to Cryptography with Java Applets*, Jones and Bartlett Publishers, Inc., 2002.
D. E. Robling, *Cryptography and Data Security*, Addison-Wesley Publishing Company, 1982.
D. J. Bernstein, J. Buchmann, & E. Dahmen, *Post-Quantum Cryptography*, Springer, 2010.
D. J. Rogers, *Broadband Quantum Cryptography*, Morgan & Claypool, 2011.
D. R. Stinson, *Cryptography: Theory and Practice*, Chapman and Hall/CRC, 2005.
D. Welsh, *Codes and Cryptography*, Oxford University Press, 1988.
E. Biham & O. Dunkelman, *Techniques for Cryptanalysis of Block Ciphers*, Springer, 2012.
F. B. Wrixon, *Codes, Ciphers, Secrets and Cryptic Communication: Making and Breaking Secret Messages from Hieroglyphs to the Internet*, Black Dog & Leventhal Publishers, 2005.
G. Blackwood, *Mysterious Messages: A History of Codes and Ciphers*, Dutton Juvenile, 2009.
G. G. A. Ganesh & P. Thorsteinson, *NET Security and Cryptography*, Prentice Hall, 2003.
G. Van Assche, *Quantum Cryptography and Secret-Key Distillation*, Cambridge University Press, 2006.
H. C. A. Van Tilborg & S. Jajodia, *Encyclopedia of Cryptography and Security*, Springer, 2011.
H. F. Gaines, *Cryptanalysis: A Study of Ciphers and Their Solution*, Dover Publications, 1989.
H. X. Mel & D. M. Baker, *Cryptography Decrypted*, Addison-Wesley Professional, 2000.
I. F. Blake, G. Seroussi, & N. P. Smart, *Advances in Elliptic Curve Cryptography*, Cambridge University Press, 2005.
I. Shparlinski, *Finite Fields: Theory and Computation: The Meeting Point of Number Theory, Computer Science, Coding Theory and Cryptography*, Springer, 2010.
J. Buchmann, *Introduction to Cryptography*, Springer, 2004.
J. C. Graff, *Cryptography and E-Commerce*, Wiley, 2000.
J. C. A. Van der Lubbe, *Basic Methods of Cryptography*, Cambridge University Press, 1998.
J. Daemen & V. Rijmen, *The Design of Rijndael: AES – The Advanced Encryption Standard*, Springer, 2012.
J. Hershey, *Cryptography Demystified*, McGraw-Hill Professional, 2002.
J. Hoffstein, J. Pipher, & M. Silverman, *An Introduction to Mathematical Cryptography*, Springer, 2010.
J. Katz & Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*, Chapman and Hall/CRC, 2007.
J. Knudsen, *Java Cryptography*, O'Reilly Media, 1998.
J. Talbot & D. Welsh, *Complexity and Cryptography: An Introduction*, Cambridge University Press, 2006.
J. Viega & M. Messier, *Secure, Programming Cookbook for C and C++: Recipes for Cryptography, Authentication, Input Validation & More*, O'Reilly Media, 2003.
J. Weiss, *Java Cryptography Extensions: Practical Guide for Programmers*, Morgan Kaufmann, 2004.
K. Lek & N. Rajapakse, *Cryptography: Protocols, Design and Applications*, Nova Science Pub Inc, 2011.
K. Schmeh, *Cryptography and Public Key Infrastructure on the Internet*, Wiley, 2003.
L. C. Washington, *Elliptic, Curves: Number Theory and Cryptography*, Chapman and Hall/CRC, 2008.
L. D. Smith, *Cryptography: The Science of Secret Writing*, Dover Publications, 1955.
L. Gerritzen, D. Goldfeld, M. Kreuzer, G. Rosenberger, & V. Shpilrain, *Algebraic Methods in Cryptography*, American Mathematical Society, 2006.
L. R. Knudsen & M. J. B. Robshaw, *The Block Cipher Companion*, Springer, 2011.
M. J. Hinek, *Cryptanalysis of RSA and Its Variants*, Chapman and Hall/CRC, 2009.
M. Mogollon, *Cryptography and Security Services: Mechanisms and Applications*, CyberTech Publishing, 2008.
M. Rosing, *Implementing Elliptic Curve Cryptography*, Manning Publications, 1998.
M. Schroeder, *Number Theory in Science and Communication: With Applications in Cryptography, Physics, Digital Information, Computing, and Self-Similarity*, Springer, 2010.
M. Welschenbach, *Cryptography in C and C++*, A-Press, 2005.
N. Ferguson & B. Schneier, *Practical Cryptography*, Wiley, 2003.
N. Ferguson, B. Schneier, & T. Kohno, *Cryptography Engineering: Design Principles and Practical Applications*, Wiley, 2010.
N. Koblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.
N. L. Biggs, *Codes: An Introduction to Information Communication and Cryptography*, Springer, 2008.
N. Moldovyan & A. Moldovyan, *Innovative Cryptography*, Charles River Media, 2006.
N. Smart, *Cryptography: An Introduction*, McGraw-Hill College, 2004.
O. Goldreich, *Foundations of Cryptography: A Primer*, Now Publishers Inc, 2005.
O. Goldreich, *Foundations of Cryptography: Volume 1, Basic Tools*, Cambridge University Press, 2007.
O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*, Cambridge University Press, 2009.
P. B. Janeczko & J. LaReau, *Top Secret: A Handbook of Codes, Ciphers and Secret Writing*, Candlewick, 2006.

P. M. Higgins, *Number Story: From Counting to Cryptography*, Springer, 2008.
R. A. Mollin, *An Introduction to Cryptography*, Chapman and Hall/CRC, 2006.
R. E. Chen, *Cryptography Research Perspectives*, Nova Science Pub Inc, 2009.
R. E. Lewand, *Cryptological Mathematics*, The Mathematical Association of America, 2000.
R. E. Smith, *Internet Cryptography*, Addison-Wesley Professional, 1997.
R. Oppliger, *Contemporary Cryptography*, Artech, 2011.
R. R. Dube, *Hardware-based Computer Security Techniques to Defeat Hackers: From Biometrics to Quantum Cryptography*, Wiley, 2008.
R. Tao, *Finite Automata and Application to Cryptography*, Springer, 2009.
S. Burnett & S. Paine, *RSA Security's Official Guide to Cryptography*, McGraw-Hill Osborne Media, 2001.
S. C. Coutinho, *The Mathematics of Ciphers: Number Theory and RSA Cryptography*, A K Peters/CRC Press, 1999.
S. Loepp & W. Wootters, *Protecting Information: From Classical Error Correction to Quantum Cryptography*, Cambridge University Press, 2006.
S. Murphy & F. Piper, *Cryptography: A Very Short Introduction*, Oxford University Press, 2002.
S. Singh, *Fermat's Enigma: The Epic Quest to Solve the World's Greatest Mathematical Problem*, Anchor, 1998.
S. Vaudenay, *A Classical Introduction to Cryptography: Applications for Communications Security*, Springer, 2010.
T. S. Denis, *Cryptography for Developers*, Syngress, 2007.
V. V. Yashchenko, *Cryptography: An Introduction*, American Mathematical Society, 2002.
W. Mao, *Modern Cryptography: Theory and Practice*, Prentice Hall, 2011.
W. Stallings, *Cryptography and Network Security: Principles and Practice*, Prentice Hall, 2010.
W. Trappe & L. C. Washington, *Introduction to Cryptography with Coding Theory*, Prentice Hall, 2005.

Chapter 3

A. Cheddad, *Digital Image Steganography: Concepts, Algorithms, and Applications*, VDM Verlag Dr. Müller, 2009.
A. Desoky, *Noiseless Steganography: The Key to Covert Communications*, Auerbach Publications, 2012.
C. S. Lu, *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property*, Idea Group Publishing, 2005.
E. Cole, *Hiding in Plain Sight: Steganography and the Art of Covert Communication*, Wiley, 2003.
F. Y. Shih, *Digital Watermarking and Steganography: Fundamentals and Techniques*, CRC Press, 2007.
G. Kipper, *Investigator's Guide to Steganography*, Auerbach Publications, 2003.
H. Singh, *Reconstructing Printed Document Using Text Based Steganography: Documentary Reconstruction*, LAP Lambert Academic Publishing, 2011.
J. Bloom, I. Cox, M. Miller, J. Fridrich, & T. Kalker, *Digital Watermarking and Steganography*, Morgan Kaufmann, 2007.
J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*, Cambridge University Press, 2009.
K. Bailey & K. Curran, *Steganography*, BookSurge Publishing, 2005.
L. M. Marvel, *Image Steganography for Hidden Communication*, Storming Media, 2000.
M. Kumar, *Steganography and Steganalysis of JPEG Images: A Statistical Approach to Information Hiding and Detection*, LAP Lambert Academic Publishing, 2011.
N. Shashidhar, *Efficient Steganography with Provable Security Guarantees*, ProQuest, UMI Dissertation Publishing, 2011.
P. Wayner, *Disappearing Cryptography: Information Hiding: Steganography & Watermarking*, Morgan Kaufmann, 2008.
R. Böhme, *Advanced Statistical Steganalysis*, Springer, 2010.
S. Katzenbeisser & F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech, 1999.
U.S. Government. *Implementation of Spread Spectrum Image Steganography*, Books LLC, 2011.
Z. Duric, S. Jajodia, & N. F. Johnson, *Information Hiding: Steganography and Watermarking – Attacks and Countermeasures*, Springer, 2000.

Chapter 4

A. Baskurt, M. Daoudi, & J. L. Dugelay, *3D Object Processing: Compression, Indexing and Watermarking*, Wiley, 2008.

A. C. Baldoza, *Data Embedding for Covert Communications, Digital Watermarking, and Information Augmentation*, Storming Media, 2000.

A. Khan & R. Ullah, *Digital Image Watermarking: Image authentication and Its Recovery after Distortion*, VDM Verlag Dr. Müller, 2011.

A. M. Al-Haj, *Advanced Techniques in Multimedia Watermarking: Image, Video and Audio Applications*, Information Science Reference, 2010.

A. M. Kothari, *Real Time analysis of Digital Watermarking Techniques: Fundamental Introduction and MATLAB Based Implementation of Image Watermarking*, LAP Lambert Academic Publishing, 2011.

A. Singhal, *Digital Colored Image Tattooing: A Neoteric, Robust and Chaotic Approach to Image Protection & Authentication Using Imperceptible Digital Watermarking*, LAP Lambert Academic Publishing, 2011.

A. Visvanathan, *Application of Binary Image in Digital Audio Watermarking*, LAP Lambert Academic Publishing, 2011.

B. Furht, E. Muharemagic, & D. Socek, *Multimedia Encryption and Watermarking*, Springer, 2010.

B. Liu & M. Wu, *Multimedia Data Hiding*, Springer, 2011.

C. S. Lu, *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property*, Idea Group Publishing, 2005.

C. Y. Lin, H. Yu, & W. Zeng, *Multimedia Security Technologies for Digital Rights Management*, Academic Press, 2006.

E. Elbasi, *Multimedia Security: Digital Image and Video Watermarking*, VDM Verlag, 2008.

F. Y. Shih, *Digital Watermarking and Steganography: Fundamentals and Techniques*, CRC Press, 2007.

H. C. Huang, L. C. Jain, & J. S. Pan, *Intelligent Watermarking Techniques*, World Scientific Pub Co Inc, 2004.

H. Sasaki, *Intellectual Property Protection for Multimedia Information Technology*, IGI Global, 2007.

J. Bloom, I. Cox, M. Miller, J. Fridrich, & T. Kalker, *Digital Watermarking and Steganography*, Morgan Kaufmann, 2007.

J. Eggers & B. Girod, *Informed Watermarking*, Springer, 2002.

L. Harte, *Introduction to Digital Rights Management (DRM); Identifying, Tracking, Authorizing and Restricting Access to Digital Media*, Althos, 2006.

L. M. Cheng & H. Y. Leung, *Study of Digital Image Watermarking in Curvelet Domain: Applications of Digital Watermarking in Frequency Domain*, VDM Verlag Dr. Müller, 2011.

M. Arnold, M. Schmucker, & S. D. Wolthusen, *Techniques and Applications of Digital Watermarking and Content Protection*, Artech, 2003.

M. Barni & F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*, CRC Press, 2004.

M. Kumar, *Digital Image Watermarking Using Contourlet Transform*, LAP Lambert Academic Publishing, 2012.

N. Cvejic & T. Seppanen, *Digital Audio Watermarking Techniques and Technologies: Applications and Benchmarks*, IGI Global, 2007.

P. Loo, *Digital Watermarking using Complex Wavelets: A Review of Digital Watermarking and the Use of Wavelets in Improving the Fidelity of Watermarked Images*, LAP Lambert Academic Publishing, 2010.

P. Wayner, *Disappearing Cryptography: Information Hiding: Steganography & Watermarking*, Morgan Kaufmann, 2008.

R. Kher, R. Thanki, & D. Vyas, *Comparative Analysis of Digital Watermarking Techniques: Analysis of Digital Image Watermarking Techniques in Spatial and Transform Domain*, LAP Lambert Academic Publishing, 2011.

S. Samuel, *World of Watermarking: Digital Rights Management for JPEG Images*, VDM Verlag Dr. Müller, 2009.

T. Kalker & J. K. Su, *Digital Watermarking Explained*, John Wiley & Sons Ltd, 2003.

X. He, *Signal Processing, Perceptual Coding and Watermarking of Digital Audio: Advanced Technologies and Models*, IGI Global, 2011.

X. He, *Watermarking in Audio: Key Techniques and Technologies*, Cambria Press, 2008.

Z. Duric, S. Jajodia, & N. F. Johnson, *Information Hiding: Steganography and Watermarking – Attacks and Countermeasures*, Springer, 2000.

Chapter 5

A. A. G. Ghorbani, W. Lu, & M. Tavallaee, *Network Intrusion Detection and Prevention: Concepts and Techniques*, Springer, 2009.
A. Chuvakin & C. Peikari, *Security Warrior*, O'Reilly Media, 2004.
A. Fadia, *Network Security: A Hacker's Perspective*, Course Technology PTR, 2006.
A. Lockhart, *Network Security Hacks: Tips & Tools for Protecting Your Privacy*, O'Reilly Media, 2006.
B. Bailey, H. Carr, & C. Snyder, *Management of Network Security*, Prentice Hall, 2009.
B. Schneier, *Secrets and Lies: Digital Security in a Networked World*, Wiley, 2004.
C. Carr & C. Snyder, *Data Communications and Network Security*, McGraw-Hill/Irwin, 2006.
C. Eagle, S. Harris, A. Harper, & J. Ness, *Gray Hat Hacking: The Ethical Hacker's Handbook*, McGraw-Hill Osborne Media, 2007.
C. Hunt, *Active Defense: A Comprehensive Guide to Network Security*, Sybex Inc, 2001.
C. Jackson, *Network Security Auditing*, Cisco Press, 2010.
C. Kaufman, R. Perlman, & M. Speciner, *Network Security: Private Communication in a Public World*, Prentice Hall, 2002.
C. McNab, *Network Security Assessment: Know Your Network*, O'Reilly Media, 2007.
D. De Capite, *Self-Defending Networks: The Next Generation of Network Security*, Cisco Press, 2006.
D. Jacobson, *Introduction to Network Security*, Chapman and Hall/CRC, 2008.
D. Mackey, *Web Security for Network and System Administrators*, Course Technology, 2003.
D. Melnichuk, *The Hacker's Underground Handbook: Learn How to Hack and What It Takes to Crack Even the Most Secure Systems!* CreateSpace, 2010.
D. J. Teumim, *Industrial Network Security*, International Society of Automation, 2010.
F. B. Wrixon, *Codes, Ciphers, Secrets and Cryptic Communication: Making and Breaking Secret Messages from Hieroglyphs to the Internet*, Black Dog & Leventhal Publishers, 2005.
E. Cole, *Network Security Bible*, Wiley, 2009.
E. Cole, J. Conley, D. Gollmann, R. L. Krutz, R. Reese, B. Reisman, & M. Ruebush, *Wiley Pathways Network Security Fundamentals*, Wiley, 2007.
E. Maiwald, *Network Security: A Beginner's Guide*, McGraw-Hill Osborne Media, 2003.
E. Maiwald, *Fundamentals of Network Security*, McGraw-Hill Osborne Media, 2003.
E. Skoudis & T. Liston, *Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses*, Prentice Hall, 2006.
G. A. Donahue, *Network Warrior*, O'Reilly Media, 2011.
G. G. A. Ganesh & P. Thorsteinson, *NET Security and Cryptography*, Prentice Hall, 2003.
G. Holden, *Guide to Firewalls and Network Security: Intrusion Detection and VPNs*, Course Technology, 2003.
G. Kurtz, S. McClure, & J. Scambray, *Hacking Exposed*, McGraw-Hill Osborne Media, 2005.
G. Schudel & D. J. Smith, *Router Security Strategies: Securing IP Network Traffic Planes*, Cisco Press, 2008.
H. Carr, B. Bailey, & C. Snyder, *Management of Network Security*, Prentice Hall, 2009.
H. J. Mattord & M. E. Whitman, *Management of Information Security*, Course Technology, 2010.
J. Andress, *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice*, Syngress, 2011.
J. Andress & T. Wilhelm, *Ninja Hacking: Unconventional Penetration Testing Tactics and Techniques*, Syngress, 2010.
J. Andress & S. Winterfeld, *Cyber Warfare: Techniques, Tactics and Tools for Security Practitioners*, Syngress, 2011.
J. Clarke & N. Dhanjani, *Network Security Tools: Writing, Hacking, and Modifying Security Tools*, O'Reilly Media, 2005.
J. C. Graff, *Cryptography and E-Commerce: A Wiley Tech Brief*, Wiley, 2000.
J. Erickson, *Hacking: The Art of Exploitation*, No Starch Press, 2008.
J. Frahim & Q. Huang, *SSL Remote Access VPNs*, Cisco Press, 2008.
J. Kouns & D. Minoli, *Security in an IPv6 Environment*, Auerbach Publications, 2008.
J. M. Stewart, *Network Security, Firewalls, and VPNs*, Jones & Bartlett Learning, 2010.
J. Novak & S. Northcutt, *Network Intrusion Detection*, Sams, 2002.
J. R. Vacca, *Computer and Information Security Handbook*, Morgan Kaufmann, 2009.
J. R. Vacca, *Network and System Security*, Syngress, 2010.
J. Scambray, *Hacking Exposed Windows: Microsoft Windows Security Secrets and Solutions*, McGraw-Hill Osborne Media, 2007.
J. Weiss, *Java Cryptography Extensions: Practical Guide for Programmers*, Morgan Kaufmann, 2004.
K. E. Himma, *Internet Security: Hacking, Counterhacking, and Security*, Jones & Bartlett Learning, 2006.

K. Frederick, S. Northcutt, R. W. Ritchey, S. Winters, & L. Zeltser, *Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers, and Intrusion Detection Systems*, Sams, 2002.
K. Jamsa & L. Klander, *Hacker Proof (General Interest)*, Delmar Cengage Learning, 2002.
K. Kent, S. Northcutt, R. W. Ritchey, S. Winters, & L. Zeltser, *Inside Network Perimeter Security*, Sams, 2005.
K. Schmeh, *Cryptography and Public Key Infrastructure on the Internet*, Wiley, 2003.
L. Klander & E. J. Renehan, *Hacker Proof: The Ultimate Guide to Network Security*, Jamsa Pr, 1997.
M. Ciampa, *Security + Guide to Network Security Fundamentals*, Course Technology, 2011.
M. D. Spivey, *Practical Hacking Techniques and Countermeasures*, Auerbach Publications, 2006.
M. E. Whitman & H. J. Mattord, *Principles of Information Security*, Course Technology, 2011.
M. Gregg, *Build Your Own Security Lab: A Field Guide for Network Testing*, Wiley, 2008.
M. Kaeo, *Designing Network Security*, Cisco Press, 2003.
M. O'Neill, *Web Services Security*, McGraw-Hill Osborne Media, 2003.
M. Pinto & D. Stuttard, *The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws*, Wiley, 2007.
N. Dhanjani, B. Hardin, & B. Rios, *Hacking: The Next Generation*, O'Reilly Media, 2009.
N. Krawetz, *Introduction to Network Security*, Charles River Media, 2006.
O. Santos, *End-to-End Network Security: Defense-in-Depth*, Cisco Press, 2007.
P. Chandra, M. Messier, & V. Viega, *Network Security with OpenSSL*, O'Reilly Media, 2002.
P. Engebretson, *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy*, Syngress, 2011.
P. Hope & B. Walther, *Web Security Testing Cookbook: Systematic Techniques to Find Problems Fast*, O'Reilly Media, 2008.
R. Bejtlich, *The Tao of Network Security Monitoring: Beyond Intrusion Detection*, Addison-Wesley Professional, 2004.
R. Bragg, M. Rhodes-Ousley, & K. Strassberg, *Network Security: The Complete Reference*, McGraw-Hill Osborne Media, 2003.
R. Panko, *Corporate Computer and Network Security*, Prentice Hall, 2009.
R. R. Dube, *Hardware-based Computer Security Techniques to Defeat Hackers: From Biometrics to Quantum Cryptography*, Wiley, 2008.
R. Weaver, *Guide to Network Defense and Countermeasures*, Course Technology, 2006.
S. Convery, *Network Security Architectures*, Cisco Press, 2011.
S. Garfinkel, A. Schwartz, & G. Spafford, *Practical Unix & Internet Security*, O'Reilly Media, 2003.
S. Hogg & E. Vyncke, *IPv6 Security*, Cisco Press, 2008.
S. McClure, G. Kurtz, & J. Scambray, *Hacking Exposed: Network Security Secrets and Solutions*, Osborne Publishing, 1999.
S. McClure & S. Shah, *Web Hacking: Attacks and Defense*, Addison-Wesley Professional, 2002.
S. M. Bellovin, W. R. Cheswick, & A. D. Rubin, *Firewalls and Internet Security: Repelling the Wily Hacker*, Addison-Wesley Professional, 2003.
S. Shah, *Hacking Web Services*, Charles River Media, 2006.
T. M. Thomas & D. Stoddard, *Network Security First-Step*, Cisco Press, 2012.
T. Wilhelm, *Professional Penetration Testing: Creating and Operating a Formal Hacking Lab*, Syngress, 2009.
V. J. R. Winkler, *Securing the Cloud: Cloud Computer Security Techniques and Tactics*, Syngress, 2011.
V. Liu, J. Scambray, & C. Sima, *Hacking exposed web applications*, McGraw-Hill Osborne Media, 2010.
V. Liu & B. Sullivan, *Web Application Security: A Beginner's Guide*, McGraw-Hill Osborne Media, 2011.
W. C. Easttom, *Network Defense and Countermeasures: Principles and Practices*, Prentice Hall, 2005.
W. J. Buchanan, *Introduction to Security and Network Forensics*, Auerbach Publications, 2011.
W. Stallings, *Cryptography and Network Security: Principles and Practice*, Prentice Hall, 2010.
W. Stallings, *Network Security Essentials: Applications and Standards*, Prentice Hall, 2010.
Y. Bhaiji, *Network Security Technologies and Solutions*, Cisco Press, 2008.

Chapter 6

A. E. Earle, *Wireless Security Handbook*, Auerbach Publications, 2006.
A. Fadia, *Hacking Mobile Phones*, Course Technology PTR, 2005.
A. Holt & C. H. Huang, *802.11 Wireless Networks: Security and Analysis*, Springer, 2010.
A. Mishra, *Security and Quality of Service in Ad Hoc Wireless Networks*, Cambridge University Press, 2008.
B. Baker, C. Hurley, R. Rogers, & F. Thornton, *Wardriving and Wireless Penetration Testing*, Syngress, 2007.
B. Carter & R. Shumway, *Wireless Security End to End*, Wiley, 2002.

B. Fleck & B. Potter, *802.11 Security*, O'Reilly Media, 2002.
C. Hurley, M. Puchol, R. Rogers, & F. Thornton, *WarDriving: Drive, Detect, Defend: A Guide to Wireless Security*, Syngress, 2004.
C. R. Elden & T. M. Swaminatha, *Wireless Security and Privacy: Best Practices and Design Techniques*, Addison-Wesley Professional, 2002.
D. Bensky, T. Bradley, P. Chandra, C. Hurley, C. Lanthem, S. Rackley, J. F. Ransome, J. Rittinghouse, T. Stapko, G. L. Stefanek, F. Thornton, & J. S. Wilson, *Wireless Security: Know It All*, Newnes, 2008.
D. Miller, K. Sankar, & S. Sundaralingam, *Cisco Wireless LAN Security*, Cisco Press, 2004.
D. Z. Du, X. Shen & Y. Xiao, *Wireless Network Security*, Springer, 2010.
E. Cayirci & C. Rong, *Security in Wireless Ad Hoc and Sensor Networks*, Wiley, 2009.
F. Anjum & P. Mouchtaris, *Security for Wireless Ad Hoc Networks*, Wiley-Interscience, 2007.
G. Held, *Securing Wireless LANs: A Practical Guide for Network Managers, LAN Administrators and the Home Office User*, Wiley, 2003.
G. Schäfer, *Security in Fixed and Wireless Networks: An Introduction to Securing Data Communications*, Wiley, 2004.
H. Chaouchi & M. Laurent-Maknavicius, *Wireless and Mobile Networks Security*, Wiley-ISTE, 2009.
H. Imai, K. Kobara, & M. G. Rahman, *Wireless Communications Security*, Artech, 2005.
J. Cache, V. Liu, & J. Wright, *Hacking Exposed Wireless*, McGraw-Hill Osborne Media, 2010.
J. Cache, V. Liu, & J. Wright, *Hacking Wireless 2.0*, Anaya Multimedia, 2011.
J. Geier, *Implementing 802.1X Security Solutions for Wired and Wireless Networks*, Wiley, 2008.
J. Geier, *Designing and Deploying 802.11n Wireless Networks*, Cisco Press, 2010.
J. Kempf, *Wireless Internet Security: Architecture and Protocols*, Cambridge University Press, 2008.
J. Li, Y. Pan, & Y. Xiao, *Security and Routing in Wireless Networks*, Nova Science Pub Inc, 2005.
J. Lopez & J. Zhou, *Wireless Sensor Network Security*, IOS Press, 2008.
J. Ma, Z. Ma, & C. Wang, *Security Access in Wireless Local Area Networks: From Architecture and Protocols to Realization*, Springer, 2009.
J. Misic & V. Misic, *Wireless Personal Area Networks: Performance, Interconnection, and Security with IEEE 802.15.4*, Wiley, 2008.
J. F. Ransome & J. Rittinghouse, *Wireless Operational Security*, Digital Press, 2004.
J. R. Vacca, *Guide to Wireless Network Security*, Springer, 2006.
K. D. Wong, *Fundamentals of Wireless Communication Engineering Technologies*, Wiley, 2012.
K. Makki, S. Makki, S. K. Makki, N. Pissinou, & P. Reiher, *Mobile and Wireless Network Security and Privacy*, Springer, 2010.
K. Munegowda, *Design and Implementation of WLAN Authentication and Security: Building Secure Wireless LAN by EAP-MD5, EAP-TLS and EAP-TTLS Protocols*, LAP Lambert Academic Publishing, 2010.
K. Roebuck, *Wireless Security: High-impact Strategies – What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity*, Vendors, Tebbo, 2011.
K. V. Gavrilenko, A. A. Mikhailovsky, & A. Vladimirov, *Wi-Foo: The Secrets of Wireless Hacking*, Addison-Wesley Professional, 2004.
K. V. Gavrilenko, A. A. Mikhailovsky, & A. Vladimirov, *Wi-Foo II: The Secrets of Wireless Hacking*, Addison-Wesley Professional, 2007.
L. Barken, E. Bermel, J. Eder, M. Fanady, A. Koebrick, M. Mee, & M. Palumbo, *Wireless Hacking: Projects for Wi-Fi Enthusiasts: Cut the Cord and Discover the World of Wireless Hacks!* Syngress, 2004.
L. R. Dondeti & T. Hardjono, *Security in Wireless LANS and MANS*, Artech, 2005.
M. Laurent-Maknavickius & H. Chaouchi, *Mobile and Wireless Networks Security*, World Scientific Publishing Company, 2008.
M. Ma, Y. Zhang, & J. Zheng, *Handbook of Research on Wireless Security*, Information Science Reference, 2008.
M. Maxim & D. Pollino, *Wireless Security*, McGraw-Hill, 2002.
M. Mcfadden, S. Morrissey, M. Schearer, B. Smith, & J. Varsalone, *Defense against the Black Arts: How Hackers Do What They Do and How to Protect against It*, CRC Press, 2011.
M. Nakhjiri & M. Nakhjiri, *AAA and Network Security for Mobile Access: Radius, Diameter, EAP, PKI and IP Mobility*, Wiley, 2005.
M. Y. Rhee, *Mobile Communication Systems and Security*, Wiley-IEEE Press, 2009.
M. Y. Rhee & M. Y. Young, *CDMA Cellular Mobile Communications and Network Security*, Prentice Hall, 1998.
M. Ciampa, *CWSP Guide to Wireless Security*, Course Technology, 2006.
N. Asokan, P. Ginzboorg, S. Holtmanns, P. Laitinen, & V. Niemi, *Cellular Authentication for Mobile and Internet Services*, Wiley, 2008.
N. Boudriga, *Security of Mobile Communications*, Auerbach Publications, 2009.
N. O'Farrell & E. Ouellet, *Hackproofing Your Wireless Network*, Syngress, 2002.

P. Chandra, *Bulletproof Wireless Security: GSM, UMTS, 802.11, and Ad Hoc Security*, Newnes, 2005.
P. C. Lekkas & R. K. Nichols, *Wireless Security: Models, Threats, and Solutions*, McGraw-Hill Professional, 2001.
P. Muller, H. Sharif, & S. Y. Tang, *WiMAX Security and Quality of Service: An End-to-End Perspective*, Wiley, 2010.
P. Stavroulakis, *Terrestrial Trunked Radio – TETRA: A Global Security Tool*, Springer, 2011.
R. D. Vines, *Wireless Security Essentials: Defending Mobile Systems from Data Piracy*, Wiley, 2002.
R. Flickenger & R. Weeks, *Wireless Hacks: Tips & Tools for Building, Extending, and Securing Your Network*, O'Reilly Media, 2005.
R. Liu & W. Trappe, *Securing Wireless Communications at the Physical Layer*, Springer, 2009.
S. A. Ahson & M. Ilyas, *Handbook of Wireless Local Area Networks: Applications, Technology, Security, and Standards*, CRC Press, 2005.
S. Fogie & C. Peikari, *Maximum Wireless Security*, Sams, 2002.
S. Miller, *Wi-Fi Security*, McGraw-Hill Professional, 2003.
S. Misra, S. C. Misra, & I. Woungang, *Guide to Wireless Ad Hoc Networks*, Springer, 2009.
S. Powell & J. P. Shim, *Wireless Technology: Applications, Management, and Security*, Springer, 2009.
U.S. Department of Commerce, *Wireless Network Security 802.11, Bluetooth and Handheld Devices*, CreateSpace, 2002.
V. Ramachandran, *BackTrack 5 Wireless Penetration Testing Beginner's Guide*, Packt Publishing, 2011.
W. A. Arbaugh & J. Edney, *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*, Addison-Wesley Professional, 2003.
W. Osterhage, *Wireless Security*, Science Publishers, 2011.
W. Soyinka, *Wireless Network Administration: A Beginner's Guide*, McGraw-Hill Osborne Media, 2010.
Z. S. Bojkovic, D. A. Milovanovic, & K. R. Rao, *Wireless Multimedia Communications: Convergence, DSP, QoS, and Security*, CRC Press, 2008.

Chapter 7

A. B. Johnston, *SIP: Understanding the Session Initiation Protocol*, Artech, 2009.
A. B. Johnston & D. M. Piscitello, *Understanding Voice Over IP Security*, Artech, 2006.
A. B. Johnston & H. Sinnreich, *Internet Communications Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol*, Wiley, 2006.
A. D. Keromytis, *Voice Over IP Security: A Comprehensive Survey of Vulnerabilities and Academic Research*, Springer, 2011.
A. Takanen & P. Thermos, *Securing VoIP Networks: Threats, Vulnerabilities, and Countermeasures*, Addison-Wesley Professional, 2007.
B. Baskin, J. J. Kanclirz, & T. Porter, *Practical VoIP Security*, Syngress, 2006.
D. Minoli, *Voice Over IPv6: Architectures for Next Generation VoIP Networks*, Newnes, 2006.
H. Dwivedi, *Hacking VoIP: Protocols, Attacks, and Countermeasures*, No Starch Press, 2008.
J. F. Durkin, *Voice-Enabling the Data Network: H.323, MGCP, SIP, QoS, SLAs, and Security*, Cisco Press, 2002.
J. F. Ransome & J. Rittinghouse, *Voice Over Internet Protocol (VoIP) Security*, Digital Press, 2004.
K. Archer, C. Cothren, R. Davis, D. Di Censo, T. Good, G. White, & D. Williams, *Voice and Data Security*, Sams, 2001.
K. O. Detken, *VoIP Security*, Hanser Fachbuchverlag, 2007.
M. Collier & D. Endler, *Hacking Exposed VoIP: Voice Over IP Security Secrets & Solutions*, McGraw-Hill Osborne Media, 2006.
M. Gough, *Video Conferencing Over IP: Configure, Secure, and Troubleshoot*, Syngress, 2006.
M. Gough & T. Porter, *How to Cheat at VoIP Security*, Syngress, 2007.
N. Wittenberg, *Understanding Voice Over IP Technology*, Delmar Cengage Learning, 2009.
P. C. Lekkas & R. K. Nichols, *Wireless Security: Models, Threats, and Solutions*, McGraw-Hill Professional, 2001.
P. K. Verma & L. Wang, *Voice Over IP Networks: Quality of Service*, Pricing and Security, Springer, 2011.
P. Park, *Voice Over IP Security*, Cisco Press, 2008.
R. B. Bates & D. Gregory, *Voice & Data Communications Handbook*, McGraw-Hill Osborne Media, 2006.
R. Barnes & K. H. Wolf, *VoIP Emergency Calling: Foundations and Practice*, Wiley, 2011.
R. F. St. Fries, *Voice Security*, Vde Verlag GmbH, 2008.
S. A. Ahson & M. Ilyas, *SIP Handbook: Services, Technologies, and Security of Session Initiation Protocol*, CRC Press, 2008.
S. A. Ahson & M. Ilyas, *VoIP Handbook: Applications, Technologies, Reliability, and Security*, CRC Press, 2008.
S. Dugan, *Cisco Voice Over IP Security*, Syngress, 2005.

S. Katsikas, *Communications and Multimedia Security – Volume 3*, Springer, 1997.
S. S. Chakraborty, H. Fathi, & R. Prasad, *Voice Over IP in Wireless Heterogeneous Networks: Signaling, Mobility and Security*, Springer, 2010.
U. Abend, J. Floroiu, J. Kuthan, H. Schulzrinne, & D. Sisalem, *SIP Security*, Wiley, 2009.

Chapter 8

ACM IV Security Services. *Countering Hostile Surveillance: Detect, Evade, and Neutralize Physical Surveillance Threats*, Paladin Press, 2008.
ACM IV Security Services. *Secrets of Surveillance: A Professional Guide to Tailing Subjects by Vehicle, Foot, Airplane, and Public Transportation*, Paladin Press, 1993.
ACM IV Security Services. *Surveillance Countermeasures: A Serious Guide to Detecting, Evading, and Eluding Threats to Personal Privacy*, Paladin Press, 1994.
B. Bruno, *Serious Surveillance for the Private Investigator*, Paladin Press, 1992.
B. Graham & K. McGowan, *101 Spy Gadgets for the Evil Genius*, McGraw-Hill/TAB Electronics, 2011.
C. Doyle & G. M. Stevens, *Privacy: Wiretapping and Electronic Eavesdropping*, Nova Science Pub Inc, 2002.
H. Eisenson, *Scanners and Secret Frequencies*, Paladin Press, 1994.
J. K. Petersen, *Understanding Surveillance Technologies: Spy Devices, Privacy, History & Applications*, Auerbach Publications, 2007.
J. Pickard, *Scanner Modifications and Antennas*, Paladin Press, 1999.
J. Raban, *Surveillance*, Vintage, 2008.
L. B. J. Taylor, *Electronic Surveillance*, Franklin Watts, 1987.
M. Chesbro, *The Privacy Handbook: Proven Countermeasures for Combating Threats to Privacy, Security, and Personal Freedom*, Paladin Press, 2002.
M. L. Shannon, *Bug Book: Everything You Ever Wanted to Know About Electronic Eavesdropping but Were Afraid to Ask*, Paladin Press, 2000.
N. Zaenglein, *The Covert Bug Book: How to Find Eavesdropping Devices and Stop Them Dead*, Paladin Press, 2007.
P. Brookes, *Electronic Surveillance Devices*, Newnes, 2001.
R. H. Dunwell, L. N. Shustov, & S. A. Vakin, *Fundamentals of Electronic Warfare*, Artech, 2001.
R. Jones, *Covert Intelligence: Electronic Eavesdropping*, CRB Research, 1990.
R. M. Marston, *Security Electronics Circuits Manual*, Newnes, 1998.
R. N. Jones, *Electronic Eavesdropping Techniques and Equipment*, Thomas Investigative Pubns Inc, 1993.
S. Bugman, *The Basement Bugger's Bible: The Professional's Guide to Creating, Building, and Planting Custom Bugs and Wiretaps*, Paladin Press, 1999.
S. Charrett, *Electronic Circuits And Secrets of an Old-Fashioned Spy*, Paladin Press, 1999.
T. Larsen, *Bench-Tested Circuits for Surveillance And Countersurveillance Technicians*, Paladin Press, 1997.
T. Monahan, *Surveillance in the Time of Insecurity*, Rutgers University Press, 2010.
T. S. Huang & Z. Zhu, *Multimodal Surveillance: Sensors, Algorithms, and Systems*, Artech, 2007.
W. Arrington, *Now Hear This! Electronic Eavesdropping Equipment Designs*, Sheffields Electronics Co., 1997.
W. G. Staples, *Everyday Surveillance: Vigilance and Visibility in Postmodern Life*, Rowman & Littlefield Publishers, 2000.
W. H. Turner, *How to Avoid Electronic Eavesdropping and Privacy Invasion*, Paladin Press, 1983.

This page intentionally left blank

INDEX

A
Access control lists (ACLs), 307–308
Access point breach technologies, 544
Access points attack tools, 563
Access policies, 549–550
ACK (Acknowledgement), 308
Acknowledgement frames, 462
Active desynchronisation attack, in wired network
  ACK packets, 350–351
  adding commands to packet, 350
  steps, 351–352
  synchronised connection, 352–353
  TCP connection, 349–350
Active hackers
  and resilient steganography, 244
  supraliminal channels, 244–245
ActiveX components, 389–391
Address Resolution Protocol (ARP), 93–94
AddRoundKey transformation, 210–212
Ad hoc mode/ networking, 20, 464
Adjudication, 145–146
Advanced Encryption Standard (AES), 144, 472, 533, 591
  algorithm, description of, 210–211
  byte arrays, 206–207
  bytes, 206
  decryption, 213
  encryption, 212
  history, 205
  input and output, 206
  key expansion function, 212–213
  mathematical preliminaries, 210
  rational schema, 211–212
  security, 213–214
  States
    AddRoundKey transformation, 210
    input and output, 207
    MixColumns transformation, 209
    operation schema, 208
    ShiftRows transformation, 209
    SubBytes transformation, 209
Advanced Research Projects Agency Network (ARPANET), 16
AES. *See* Advanced Encryption Standard (AES)
Air mail, 236
Algorithms, cryptographic
  AES standard, 210–211
  block and stream ciphers, 174–175
  block chaining mode, 174
  choice of, 175
  cipher block chaining mode, 171–172
  communication channels, 176–177
  compression, 178
  definition, 138
  destruction of information, 178
  electronic codebook mode, 171
  encryption for storage, 177
  feedback cipher mode, 173, 174
  hardware and software, encryption via, 177–178
  output-feedback mode, 174
  self-synchronising stream ciphers, 173
  stream ciphers, 172–173
  symmetric-key and public-key, 175–176
  synchronous stream ciphers, 174
  types, 139
  violation, 140
American National Standards Institute (ANSI), 518
Amplitude modulation (AM), 604, 605

Analogue methods, 587
Analogue modulation
  amplitude modulation, 34–35
  frequency modulation, 35–36
  phase modulation, 36–37
Analogue scramblers, 572
Analogue to digital (A/D), 571
Annualised loss expectancy (ALE), 548
Annualised rate of occurrence (ARO), 547
Anonymity Key (AK), 490
Antenna
  beam width, 501
  diversity of, 502
  fresnel zone, 502–503
  gain of, 500–501
  multiple paths, 501–502
  path loss, 501
  polarisation, 500
  types
    directional antennae, 503
    home-built antennae, 504
    omnidirectional antennae, 503
Anti-bugging devices
  audible frequency jammers, 615
  broadband detectors, 605–606
  cellular technology, 606–607
  electromagnetic jammers, 615
  encrypted phones, 616–617
  hidden miniature camera detectors, 614
  inaudible frequency jammers, 616
  laser beam bugging devices, jammers, 616
  multifunction devices, 609–612
  multifunction spectrum analysers, 609
  non-linear junction detectors, 610–614
  scanners, 604–605
  software utilities, 617
  spectrum analysers, 607–608
  TEMPEST, 617–619
  wireless remote camera detectors, 614–615
Antivirus, 407–408
Applets, 385
Application layer, 13
  domain name system, 109–110
  electronic mail (email)
    functions, 111
    message transfer agents, 112–113
    user agents, 112
  multimedia
    audio compression, 125–126
    digital audio, 124–125
    internet radio, 127
    streaming audio, 126–127
    video, 129–131
    video compression, 131–136
    voice over IP, 127–129
  World Wide Web (WWW)
    architecture, 114
    client side, 114–116
    dynamic web documents, 120–122
    history, 113
    Hypertext Transfer Protocol, 122
    performance improvement, 122–124
    server side, 116–119
    static web documents, 120
Application-level firewall, 304
Application programming interfaces (APIs), 387
Arbitrary names, 144
ARPANET. *See* Advanced Research Projects Agency NETwork (ARPANET)
Association request frame, 460
Association response frame, 460
A5 Stream ciphers, 199–200
Asynchronous Transfer Mode (ATM), 18–19
Attacks, on wired networks
  analysis
    administrative contact, 412
    automatic scanner, 415
    commands, 415
    domain name, 412
    FIN scanner, 414
    nslookup, 413
    physical addres, 412
    port scanner, 414
    scanner ping, 414
    zone transfer, 413
  execution of
    brute-force, 419
    hidden accounts, 416
    man in the middle, 416–417
    Smurf and *spoofing,* 418–419
    teardrop, 417–418
    three-packet handshake, 417
Audible frequency jammers, 615
Audio compression, 125–126
Audit phase, 510
Authentication, 459–460
  bluetooth, 471–472
  commercial transactions, on Internet, 376
  cryptography, 138
    dictionary and salt attacks, 157
    and key exchange, 158
    one-way functions, 157
    public-key cryptography, 157
  Secure Socket Layer, 334
  virtual private networks, 363
  wired networks security, 358–360
  wireless networks, 459–460

Authentication and key management (AKM), 531
Authentication header (AH), 553–554
Authentication management field (MFA), 490

B
Band shift, 574
Base station (BS), 476
Basic service set (BSS), 424
Bastion host, 298
Beacon frames, 459
Beam width, 501
B frames (bidirectional), 135
Bi-phase encodings, 33–34
Bit error rate, 277
Black hat hacker, 293
Blind-watermarking, 273
Block chaining mode (BCM), 174
Block ciphers, 174–175, 196
Block Mode, 484
Bluejacking, 454
Bluetooth, 70–72
  application and presentation level, 469
  authentication process, 471–472
  baseband level, 469
  cipher, 472, 473
  data link level, 468
  devices, 605
  elements, 467
  encryption keys, 472–473
  frequency selection module, 470
  key negotiation, 473
  L2CAP connection, 470–471
  master unit, 470
  network level, 468–469
  physical layer, 468
  security architecture, 467–469
  security manager, 471
  session layer, 469
  Special Interest Group (SIG), 467
  specifications, 467
  spoofing process, 470
  transport level, 469
  vulnerability of, 474
Blum integers, 185
Breach technologies
  802.1x vulnerabilities, 541–542
  access point
    HTTP, 544
    RADIUS, 544
    SNMP, 544
    Telnet, 544
  DoS attacks, 540
  MAC filtering attacks, 540
  MIC attacks, 542
  RADIUS vulnerabilities, 541
  WEP, 538–540
  wireless gateways, 543
  WPA and 802.11i attacks, 543
Broadband bugging device detectors, 605–606
Broadband wireless, 69–70
Brute-force attacks, 419
Bugging devices
  anti-bugging devices
    audible frequency jammers, 615
    broadband detectors, 605–606
    cellular technology, 606–607
    electromagnetic jammers, 615
    encrypted phones, 616–617
    hidden miniature camera detectors, 614
    inaudible frequency jammers, 616
    laser beam bugging devices, jammers, 616
    multifunction devices, 609–612
    multifunction spectrum analysers, 609
    non-linear junction detectors, 610–614
    scanners, 604–605
    software utilities, 617
    spectrum analysers, 607–608
    TEMPEST, 617–619
    wireless remote camera detectors, 614–615
  directional microphones, 599–600
  GPS technology, 600–601
  keystroke recorders, 602–603
  laser devices, 600
  miniature audio, 602
  miniature cameras, 593, 597
    cellular technology, 596–597
    infrared, 598–599
    mains carrier, 598
    microwave, 594–595
    radio frequency, 594–595
    telephone, 599
    ultrasonic devices, 599
  mobile phone, 601
  portable document scanners, 603
  procedures and guidelines, 619–621
  software, computers, 603
  stethoscopic microphones, 602
  video recorders, 602
Byte arrays, 206–207
Bytes, 206

C
Call control, 452
Calling line identification (CLID), 497
CAPSTONE, 230–231
Carrier sense multiple access/collision avoidance (CSMA/CA)
  acknowledgement frames, 462

clear to send frame, 461
data frame, 461–462
request to send, 461
Carrier sense multiple access with collision detection (CSMA/CD), 424
Cascading, 197
Category 5 wiring (CAT5), 422
CDMA. *See* Code division multiple access (CDMA)
Cellular phone, 616
Cellular phone technology, 452–453, 596–597, 606–607
code division multiple access, 476
first generation, 475
GSM standard, 476–478
MMS service, 485–487
second generation, 476
SMS service, 478–495
UMTS standard, 487–491
Cellular telephone network
code division multiple access, 53–55
first generation/analogue voice/1G, 52–53
second generation/digital voice/2G, 53
third generation/voice and digital data/3G, 55
CGI scripts
attacks, 395
HTML text files, 393
HTTP connection, 394
security issues, 397–399
servers, 394
specifications, 394–395
transaction, 394
Challenge Handshake Authentication Protocol (CHAP), 521
Channel vocoder, 581–582
Chat and video chat programs, 616–617
Chinese Remainder Theorem, 183–184
The Chinese, steganography in, 234
Cipher, 138, 472, 473
Cipher block chaining (CBC) mode, 171–172
Cipher combination, 197
Ciphertext, 137
Circuit-level firewall, 304–305
Classic source-filter model, 567–568
Classless inter-domain routing (CIDR), 91
Cleartext, 137
Clear to send frame (CTS), 461
Client-server communication mode, 3, 4
CLIPPER, 230
Coaxial cable, 47
Code division multiple access (CDMA), 53–55, 476
Code-excited linear prediction (CELP), 584
Code obfuscation, 246
Commercial transactions, on Internet
authentication, 376
buyer additional information, 379–380
confidentiality, 377
credit cards, 380–381
digital signatures, 376–377
electronic cash, 377–378
plaintext, 376
Secure Electronic Transmission (SET), 381–382
Common Cryptographic Architecture (CCA), 225–226
COMP 128, 478
Complexity theory, 180–181
Computer algorithms, 144
Computers, bugging software, 603
Computer viruses
bomb, 404–405
concealment, 402–404
considerations, 400
developments, 400
prevention
access control, 406–407
antivirus, 407–408
BIOS, 407
cyclic redundancy check, 407
FDISK activates, 407
heuristic analysers, 409
protection, 409–411
replication, 400–402
trojan horses, 406
worm viruses, 405–406
Conference of European Posts and Telecommunications (CEPT), 476
Confidentiality, 137
Congestion control algorithms
feedback mechanism, 85
open and closed loop solutions, 85
packets, 84
routers, 84
Connection-oriented networks
ATM, 18–19
ethernet, 19–20
X. 25 and frame relay, 18
Constructive steganography, 238–239
Continuous synchronisation, 571–572
Control signals, 452
Cookies, 119, 391–392
Copyright protection, watermarking role in, 273
Cordless phones, 455, 605
Cover, 237, 238, 247
Cover-object, 237, 238
Cracking tools, 563
Credit cards, 380–381
Crosstalk, 32
Crouched bugging devices, 608

Cryptanalysis, 137, 139
  analogue methods, 587
  digital ciphers, 588
  linear prediction vocoder cryptanalysis, 588
  spectrograph, 586–587
  tools and parameters, 586
Cryptapix, 260
Cryptoanalytic attacks, 139–141, 146
Cryptography
  Advanced Encryption Standard (*See* Advanced Encryption Standard (AES))
  applications
    CAPSTONE, 230–231
    CLIPPER, 230
    Common Cryptographic Architecture (CCA), 225–226
    IBM keys, 223–224
    ISO authentication, 226–228
    Kerberos, 224
    kryptonight, 225
    Message Security Protocol (MSP), 228
    Pretty Good Privacy (PGP), 229
    Privacy Enhanced Mail (PEM), 228
    Public-Key Cryptographic Standards (PKCS), 230
    SESAME, 225
    smart card, 229
    STU-III, 224
    TIS/PEM, 228
  authentication, 138
    dictionary and salt attacks, 157
    and key exchange, 158
    one-way functions, 157
    public-key cryptography, 157
  block ciphers, 196
  cipher combination, 197
  complexity theory, 180–181
  computer algorithms, 144
  confidentiality, 137
  cryptographic algorithm
    block chaining mode, 174
    block ciphers and stream ciphers, 174–175
    choice of, 175
    cipher block chaining mode, 171–172
    communication channels, 176–177
    compression, 178
    definition, 138
    destruction of information, 178
    electronic codebook mode, 171
    encryption for storage, 177
    feedback cipher mode, 173, 174
    hardware and software, encryption via, 177–178
    output-feedback mode, 174
    self-synchronising stream ciphers, 173
    stream ciphers, 172–173
    symmetric-key and public-key, 175–176
    synchronous stream ciphers, 174
    types, 139
    violation, 140
Data Encryption Standard (DES)
  algorithm, 187–191
  differential and linear analysis, 193–195
  final permutation, 193
  operation, 187, 188
  P-box permutation, 193
  proposal criteria, 187
  S-boxes, 192
  security of, 191, 193
  variant of, 195–196
delegated signature, 161
digitally certified email, 162
digital signature
  algorithms and technologies, 152
  attacks against public-key, 153
  with encryption, 153
  manual signature, 149
  multiple signatures, 152
  non-repudiation and, 152
  oneway hash functions, 151
  public-key, 150–151
  and stamping, 151
  symmetric and arbitrator, 149–150
discrete logarithms, 186
division of secret, 159
factorisation, 185
group signature, 161–162
hybrid cryptosystems, 149
information theory, 178–180
integrity, 138
keys
  compromising of, 169–170
  destruction, 170
  escrow, 162
  generation, 166–167
  lifespan of, 170
  management, 166
  storage, 169
  transfer, 168
  update, 169
  verification, 168
keys exchange
  Diffie-Hellman, 220–221
  digital signature, 155–156
  encrypted key exchange (EKE), 221–222
  interlock protocol, 155
  keys and messages transmission, 156
  man-in-the-middle attack, 154–155
  public-key cryptography, 154

station-station protocol, 221
symmetric cryptography, 154
multiple public-key cryptography, 158–159
non-repudiation, 138
numbers theory
Blum integers, 185
Chinese Remainder Theorem, 183–184
Euler's totient function, 183
Fermat's little theorem, 183
Galois field, 185
generators, 185
inverse modulo of number, 182–183
Jacobi symbol, 184–185
Legendre symbol, 184
maximum common divisor, 182
modular arithmetic, 181–182
prime number, 182
quadratic residues, 184
resolution for coefficients, 183
one-time pad, 143–144
one-way functions, 147–148, 165
one-way hash functions
birthday attack, 203
characteristics, 203
description, 148
message authentication code (MAC), 205
public-key algorithms, 204
secure hash value, 204
symmetric block algorithms, 204
optimal key length, 165–166
prime numbers generation, 186
protection archives, 160
protocol
adjudication, 145–146
arbitrary names, 144
attacks on, 146
properties, 144
self-reinforcing, 146
pseudo-random sequences generators
additive generators, 200
congruent linear generators, 197–198
multiple streams, 200–201
PKZIP, 200
shift records, 198–199
stream ciphers, 199–200
public-key algorithms/asymmetric algorithms
communication by, 148–149
description, 139
digital signature algorithm, 217–219
elliptic curve, 217
key management, 171
length of, 164
RSA algorithm, 215–217
quantum cryptography, 222–223
random and pseudo-random sequences, 153
random sequence generators
computer clock, 202
distillation of randomness, 203
keyboard latency typing, 202
polarisation and correlation, 202
random noise, 201–202
replacement and transposition ciphers, 141–142
schematisation, 138, 139
secret sharing, 159
stamping services
arbitration solution, 160
improved arbitration solution, 160–161
symmetric key algorithms
communication by, 147
description, 139
length of, 162–164
using keys, 168–169
vs. steganography, 233
XOR operation/exclusive OR, 142–143
Cryptology, 137
Cryptosystem, 139
CSMA/CA. *See* Carrier sense multiple access/collision avoidance (CSMA/CA)
Cyclic Redundancy Check (CRC), 407, 485

D

Data Encryption Standard (DES), 144, 591
algorithm, 187–191
differential and linear analysis, 193–195
final permutation, 193
operation, 187, 188
P-box permutation, 193
proposal criteria, 187
S-boxes, 192
security of, 191, 193
S tools, 266
variant of, 195–196
Data frame, 461–462
Data link layer, 13
bridges, 74
datagram subnet., 76
devices location, 75
hub connection, 76, 78
mobile hosts, 77
multiple LANs connection, 75
networks and security, 74
packets switching, 78
remote networks connection, 75
router, 79
switches, 79
virtual circuit subnet, 77
Data link physical layer
functional aspects, 56

physical level transmission errors, 56–75
purpose, 55
services, 55–56
Data noise, steganography in, 242–243
Data stash, 261
Data tracking, watermarking role in, 273
DCT. *See* Discrete cosine transform (DCT)
De-authentication frame, 460
Debugger, 619–621
Decryption
AES standard, 213
definition, 137
Delegated signature, 161
Demilitarised zone (DMZ), 554
Denial-of-service (DoS) attack, 346, 451
DES. *See* Data Encryption Standard (DES)
Detection error, 277
Device-based firewalls, 320
Device verification, 509
de Vigenere, Blaise, 235
D frames (encoded DC), 135
Diffie-Hellman algorithms, 220–221
Digital audio, 124–125
Digital ciphers, 588
Digital encryption, 580
Digitally certified email, 162
Digital scramblers, 572
Digital Signature Algorithm (DSA), 144
Digital signatures
algorithms and technologies, 152
attacks against public-key, 153
commercial transactions, on internet, 376–377
with encryption, 153
manual signature, 149
multiple signatures, 152
non-repudiation and, 152
oneway hash functions, 151
public-key, 150–151
public-key algorithms
message sign, 218
NIST proposal, 217–218
operation, 218
via discrete logarithms, 219
secure-HTTP (S-HTTP) protocol, 331–332
and stamping, 151
symmetric and arbitrator, 149–150
Digital subscriber line (DSL), 425
Digital to analogue (D/A), 571
Digital watermarking
algorithm requirements, 274–275
applications, 273–274
evaluation of systems
metric difference of distortions, 276
objective evaluation, 277
parameters, 275–276
ROC graph, 277
testing phases, 276
TPF, 278
evolution and standardisation, 278–279
fingerprint, 291–292
history and terminology, 271–272
principles, 272–273
strength requirements
attacks on, 286
detection, 290
IFPI, 285–286
malfunction of detector, 287–288
signal decrease, 286–287
system architectures, 290–291
watermark counterfeiting, 288–289
technologies
message fusion, in document, 284
pixels/blocks selection, 279–280
signal formatting, 283–284
video images, 285
watermark detector optimisation, 284–285
work selection space, 280–283
types, 272–273
watermark removal algorithms, 278
Directional antennae, 503
Directional microphones, 599–600
Direct Sequence (DS), 254
Direct spread spectrum technology, 608
Disassociation frame, 460
Disaster prevention and recovery
division of, 421
network disasters
configuration saving, 426
media, 421
single points of failure, 425–426
topology, 423–425
server disasters
application server provider, 431
clustering, 429–430
continuity groups, 427
data backup, 430–431
RAID, 427–428
redundant servers, 428–429
server recovery, 431–432
simulation, 432
Discrete cosine transform (DCT), 252–253, 577
Discrete Fourier Transform (DFT), 251, 577
Discrete logarithms, 186
Distortion steganography
digital images, 257
written texts, 256–257
Distributed coordination function (DCF), 462–463
Distribution system (DS), 458

Domain name system (DNS), 109–110
Domain transformation steganography
  concealing information and data compression, 254
  DCT domain, 252–253
  digital sound, 253–254
DoS tools, 563
Double encryption, 197
Dual-homed host firewall, 306
Dynamic Host Configuration Protocol (DHCP), 94
Dynamic packet filtering, firewalls
  ACK, 313–314
  attacker, conditions for, 314
  example of, 313
  FIN scan, 313, 317
  implementation, 315
  port scan block, 316
  rules, 312
  session establishment, 315
  and static filter, 312
  SYN, 313
  traffic behaviour, 313
Dynamic web documents, 120–122

E
The Egyptians, steganography in, 233
Eigen-beam forming, 68
Electric fields, 619
Electromagnetic fields, 619
Electromagnetic interference (EMI), 422
Electromagnetic jammers, 615
Electromagnetic spectrum transmissions, 48–49
Electromagnetic wave, 446–448
Electronic cash system, 377–379
Electronic codebook mode (ECM), 171
Electronic mail (email)
  functions, 111
  message transfer agents, 112–113
  user agents, 112
Emission security (EMSEC), 618
Encapsulating secure payload (ESP), 554
Encrypted key exchange (EKE), 221–222
Encrypted phones, 616–617
Encryption. *See* Cryptography
Enhanced Data rates for GSM Evolution (EDGE), 55
Enigma, 142
Ethernet, 19–20
Euler's totient function, 183
European Telecommunications Standards Institute (ETSI), 476
Executable files, steganography in, 246
Exposed station problem, 64
Exposure factor (EF), 547
EXPTIME, 181
Extended basic service set (EBSS), 464–465
Extended service set (ESS), 424
Extensible Authentication Protocol (EAP), 518
  Flexible Authentication via Secure Tunnelling (EAP-FAST), 526–528
  Message Digest algorithm 5 (EAP-MD5), 522–523
  Over Local Area Network (EAPOL), 519–520
  Transport Layer Security (EAP-TLS), 523–524
  Tunnel Transport Layer Security (EAP-TTLS), 524–525
Extensible Markup Language (XML), 120

F
Factorisation, 185
Fax, 616
Feedback cipher mode, 173, 174
Fermat's little theorem, 183
Fibre Distributed Data Interface (FDDI), 423–424
FIN (Final), 308
Fingerprint
  application, 291
  classification, 292
  digital data definitions, 291–292
  watermarking role in, 273
Firewall
  application-level firewall, 304
  architectures
    dual-homed host firewall, 306
    screened host firewall, 307
    screened subnet firewall, 307
  bastion host, 298
  circuit-level firewall, 304–305
  data traffic flow, 297
  design of, 299–300
  device-based firewalls, 320
  disadvantages, 337–338
  dynamic packet filtering
    ACK, 313–314
    attacker, conditions for, 314
    example of, 313
    FIN scan, 313, 317
    implementation, 315
    port scan block, 316
    rules, 312
    session establishment, 315
    and static filter, 312
    SYN, 313
    traffic behaviour, 313
  functional characteristics
    good log, 322
    hiding NAT, 321–322
    LDAP, 323
    PAT, 322
    private addressing, 320–321
    RADIUS, 323

static NAT, 322
translation of addresses, 320, 321
VPNs, 322–323
introduction to, 301–302
ISO/OSI model, 302
limits of, 300
Linux-based firewalls, 319
location of, 323–324
Macintosh-based firewalls, 318
for Microsoft Windows, 319–320
network access, 301
network-level firewalls, 303–304
network security assessments
level A, 326–327
level B, 325–326
level C, 324–325
level D, 324
parameters, 297–298
proxy servers, 305–306
risk regions, 300–301
screening router, 299, 302
software and hardware, 298
stateful filtering, 317
static filtering packet
complications, 310
ICMP, 308–310
mean transfer unit, 310–311
packet filter, 307–308
TCP traffic, 308
traffic flags, 308
UNIX-based firewalls, 318–319
First generation/analogue voice/1G telephone network, 52–53, 475
The First World War, steganography, 236
Fixed telephone network, 51–52
Formant vocoder, 581
FORWARDABLE flag, 374
Fourier transform, 26
Frame management
association response and request, 460
authentication, 459–460
Beacon, 459
carrier sense multiple access/collision avoidance, 460–462
disassociation and de-authentication, 460
distributed coordination function, 462–463
distribution system, 458
fragment and retry, 458
fragmentation, 462
interframe spacing, 463
point coordination function, 463
power management, 458
probe response and request, 459
protocol version, 458
service set identifier, 463
type, 458
wired equivalent privacy, 458–459
Frequency division multiplexing (FDM), 40–45
Frequency hopping (FH), 254
Frequency hopping spread spectrum (FHSS), 63
Frequency modulation (FM), 604, 605
Frequency shift keying (FSK), 571
Fresnel zone, 502–503

G

Galois field, 185
Gateway interface (CGI), 120, 121
Gateway mobile switching centre (GMSC), 477
General Packet Radio Service (GPRS), 55, 477, 601
General source-filter model, 568–569
Generative steganography, 238
GirolamoCardano, 234
Global positioning system (GPS) technology, 491–492, 600–601
Global System for Mobile communications (GSM), 585, 596
GPS technology. *See* Global positioning system (GPS) technology
The Greeks, steganography in, 233–234
Grey hat hacker, 293
Groupe Special Mobile, 476–478
Group master key (GMK), 529
Group signature, 161–162

H

Hacking and hackers
active hackers, 244–245
bluejacking, 454
cordless phone driving, 455
malicious hackers, 244–245
motivation of, 453
war chalking, 454
war dialing, 455
war drivers, 453, 455–456
war flying, 454
war walker, 454
X10 driving, 455
Hermeticstego, 262
Hertz, Heinrich, 445
Heuristic analysers, 409
Hidden account attacks, 416
Hidden messages, 258–259
Hidden miniature camera detectors, 614
Hide in picture
Blowfish, 263
Rijndael, 264
Hiding Network Address Translation (hiding NAT), 321–322

High-rate DSSS (HR-DSSS), 63
HiperACCESS, 73
HIPERLAN, 72–73
HiperLINK, 73
Home-built antennae, 504
Home location register (HLR), 480
H.323 protocols, 127–128
HTTP protocol. *See* Hypertext Transfer Protocol (HTTP)
Hybrid cryptosystems, 149
Hybrid tools, 562
Hypertext Markup Language (HTML), 115, 120
Hypertext Transfer Protocol (HTTP), 122, 544. *See also* Secure-HTTP (S-HTTP) protocol

I

IBM keys, 223–224
IEEE 802.11 families, 70
  802.11 a standard, 65–66
  802.11 b standard, 66
  802.11 g standard, 66
  802.11 i standard, 66–67
    advanced encryption standard, 533
    features, 533–534
    robust security network, 530–532
    temporal key integrity protocol, 532–533
    and WPA, 543
  802.11 n standard, 67–69
  protocol stack, 63
IEEE 802 working groups, 24
IEEE 802.1x standard
  applicant device, 519
  authentication server, 518
  authenticator, 519
  EAPOL, 519–520
  vulnerabilities, 541–542
I frames (intra-coded), 135
Image authentication, watermarking role in, 274
Inaudible frequency jammers, 616
Independent basic service set (IBSS), 424, 464
Industrial, scientific and medical (ISM) band, 457, 595
Information element data (IEDx), 482
Information element identifier (IEIx), 482
Information element length (IELx), 482
Information theory, 178–180
  availability, 450
  confidentiality, 449–450
  integrity, 450
Infrared, 474
  bugging devices, 598–599
  and millimetre wave transmissions, 50
Infrastructure mode, 464–465
Initial synchronisation, 570–571
Injective steganography, 238
Institute of Electrical and Electronics Engineers (IEEE), 456
Integrity, 138
Interframe spacing, 463
International Data Encryption Algorithm (IDEA), 267
International entities, 22, 24
International Federation for the Phonographic Industry (IFPI), 285–286
International Organization for Standardization/Open System Interconnection (ISO/OSI). *See also* Open System Interconnection (OSI)
  firewall, 302
  personal area networks, 468–469
  voice security, 590
International Phonetic Alphabet, 567
International Telecommunication Union (ITU), 22, 487
Internet
  architecture, 17
  ARPANET, 16
  NSFNET, 16–17
  TCP transport protocol, 100–104
  UDP transport protocol, 99
  usage, 17
Internet Control Message Protocol (ICMP), 93, 537
  type 5 field, 310
  values of, 309
  values of code, 310
Internet Header Length (IHL), 88
Internet radio, 127
Internet service provider (ISP), 465
Internetwork/internet, 8
Inter-realm key, 370–371
Intrusion detection system (IDS)
  advantages, 339
  configuration, 344–346
  data packet, 340
  and firewall, 341
  fusion of, 343–344
  installation on host, 342–343
  interruption of session, 341
  limitations, 339–340
  non-integral protection, 342
Inverse modulo of number, 182–183
Invisible communication, 246
ISO authentication, 226–228
ISO OSI model. *See* International Organization for Standardization/Open System Interconnection (ISO/OSI)

J

Jacobi symbol, 184–185
Jammers
  audible frequency jammers, 615

inaudible frequency jammers, 616
laser beam bugging devices, 616
Java Archive (JAR), 387
Java language, wired networks security
applets, 385
application programming interfaces (APIs), 387
developments, 384–385
Java Archive (JAR), 387
persistent objects, 386
sandbox security model, 386
strings, 387
Java virtual machine (JVM), 121, 385
JPEG, 131–133

K
Kerberos keys exchange, 224
access an archive, 374–375
authentication, 368
distributed system, 365–366
inter-realm key, 370–371
operating mechanism, 366
plaintext request, 368–370
realm key, 370–371
ticket flags, 372–374
tickets comparision, 367
transmission mode, 368
trusted server, 365–367
vulnerability, 375–376
Kerckhoffs, Auguste, 235
Key Confirmation Key (KCK), 533
Keys
catchers, 602–603
compromising of, 169–170
destruction, 170
escrow, 162
exchange of
Diffie-Hellman, 220–221
digital signature, 155–156
encrypted key exchange (EKE), 221–222
interlock protocol, 155
keys and messages transmission, 156
man-in-the-middle attack, 154–155
public-key cryptography, 154
station-station protocol, 221
symmetric cryptography, 154
expansion function, 212–213
generation, 166–167
lifespan of, 170
management, 166
storage, 169
transfer, 168
update, 169
verification, 168
Keystroke recorders, 602–603
Kryptonight, 225

L
Landline phone, 616
Laplace filtering, 243
Laptops, 511
Laser beam bugging devices, 600, 616
Laser diode, 48
Layer network, 13
Least Significant Bit (LSB), 242
Legendre symbol, 184
LFSR based stream ciphers, 199
Light emitting diodes (LEDs), 48
Light wave transmissions, 50
Lightweight Directory Access Protocol (LDAP), 323
Lightweight Extensible Authentication Protocol (LEAP), 526
Linear feedback shift register (LFSR), 199–200
Linear prediction (LP) analysis, 569
Linear prediction coefficient (LPC), 583
Linear prediction modelling (LPM), 569–570
Linear prediction vocoder cryptanalysis, 588
Linear Predictive Coding (LPC), 569
Linux-based firewalls, 319
Liquid crystal display (LCD) screen, 614
Local area networks (LANs)
bus and ring technology, 6
size, 5
subnet connection, 7
topology, 6
transmission, 5–6
Local area network (LAN) standards, 466–467
Local wireless networks. *See* Wireless networks
Long-duration spectrum analysis, 608

M
Macintosh-based firewalls, 318
Magnetic fields, 619
Malicious codes, 451
Malicious hackers, 244–245
Manchester encoding, 33–34
Man in the middle attacks, 416–417
Mary Queen of Scots, 235
Master site (MS), 492
Maxwell, James Clerk, 445
MDC, 268
Medium access control sub-layer
10Base2 wiring, 59
broadcast channels, 57
data link layer switching, 74–79
dynamic assignment, 58
Ethernet, 59–60

multiplexing techniques, 57
networks, 57
nomenclature, 59
protocols, 58–59
switch, 60
UTP category, 61–62
Wireless networks (*See* Wireless LAN (WLAN))
Mesh networks, 466
Message authentication Code (MAC), 205, 542
Message integrity control (MIC), 529, 533, 542
Message Security Protocol (MSP), 228
Metropolitan area networks (MANs), 6
Microsoft Windows, firewalls for, 319–320
Microwave bugging device, 594–595
Microwave transmissions, 49–50
Miniature audio, 602
Miniature cameras, 593, 597
cellular technology, 596–597
infrared bugging devices, 598–599
mains carrier bugging device, 598
microwave, 594–595
radio frequency, 594–595
telephone, 599
ultrasonic devices, 599
MixColumns transformation, 209, 211
Mobile equipment (ME), 488
Mobile networks, routing algorithm on, 82–84
Mobile phone, 605
bugging devices, 601
Mobile station (MS), 476
Mobile Station International Subscriber Directory Number (MSISDN), 476
Mobile station roaming number (MSRN), 477
Mobile switching centre (MSC), 53, 480
Mobile technology. *See* Cellular phone technology
Mobile user devices
laptops, 511
PDA devices, 512
portable scanners, 512
smart phone, 512–513
tablet computers, 511–512
Wi-Fi phones, 513
MPEG, 133–136
Multicast mode, 4
Multifunction devices, 609
hidden microphones, 612
infrared bugging devices, 611
mains carrier bugging devices, 611
microwave bugging devices, 610, 611
miniature cameras, 611
radio frequency, 610
telephone lines, 612
Multifunction spectrum analysers, 609
Multimedia
audio compression, 125–126
digital audio, 124–125
internet radio, 127
streaming audio, 126–127
video, 129–131
video compression, 131–136
voice over IP, 127–129
Multimedia messaging service (MMS), 485–487
Multiple input multiple output (MIMO), 67, 68
Multiple paths, 501–502
Multiple public-key cryptography, 158–159
Multiplexing
analogue and digital connection, 45
electromagnetic spectrum, 44
frequency division multiplexing, 40–45
light propagation, 43
medium distance call, 44
QPSK constellation, 45
telephone network, 44
time division multiplexing, 42, 46
total refraction and reflection, 43
wavelength division multiplexing, 42, 45–46

N

Navigation Satellite Timing and Ranging (NAVSTAR), 491
Network(s)
connection-oriented
ATM, 18–19
ethernet, 19–20
X. 25 and frame relay, 18
internet
architecture, 17
ARPANET, 16
NSFNET, 16–17
usage, 17
wireless LAN
802.11 multiple cell network, 22–23
ethernet standard, 20
individual units, 20, 21
periodic signals, 21
radio base station/AP, 20
sinusoidal signals, 22, 23
Network address translation (NAT), 92
Network attacks
active desynchronisation attack, 349–353
denial-of-service (DoS) attack, 346
hyperlink *spoofing,* 355
number sequence anticipation attack, 346–348
sniffer attacks, 348–349
spoofing attack, 353–355

TCP protocol hijack, 348
web *spoofing,* 355–358
Network design, 509
Network disasters
configuration saving, 426
media, 421
single points of failure, 425–426
topology, 423–425
Network hardware
broadcast networks, 4
internet, 8
local network, 5–6
metropolitan area network, 6
point-to-point networks, 4
wide area network, 6–7
wireless networks, 7–8
Network Interface Device (NID), 52
Network layer
congestion control algorithms, 84–86
connection between networks, 87–88
datagram and virtual circuit subnets, 80
on Internet
ARP, 93–94
autonomous systems, 88
DHCP, 94
ICMP, 93
IP datagram, 88
IP, division and classes, 89, 90
IP transmission, 94–95
IPv4 to IPv6, 88–89
main network into subnets, 91
NAT, 92–93
quality of service, 86–87
routing algorithm
adaptive and non-adaptive, 81
applications, 82
connections state, 82
distance vector, 81
flooding, 81
on mobile networks, 82–84
optimality, 81
services, 79–80
store-and-forward mechanism, 79
Network-level firewalls, 303–304
Network operation modes, 3
Network security assessments, wired
level A, 326–327
level B, 325–326
level C, 324–325
level D, 324
Network service access points (NSAPs), 97
Network software
connection and without connection, 10
layers, 9–10
protocols, 8–9
service primitives, 10–11
services and protocols, 11, 12
Noise, signal in transmission, 31–32
Non-blind watermarking, 272–273
Non-linear junction detectors (NLJR), 610–614
detection distance, 613
electromagnetic signal, 611
electronic component, 610
oxidised metals, 612
pulsed mode, 613
radio frequency, 613
voltage-current characteristics, 611, 613
Non-repudiation, 138
Non-return-to-zero invert (NRZI) encoding, 33
Non-return-to-zero level (NRZ-L) encoding, 33
NSFNET. *See* US National Science Foundation NET (NSFNET)
Numbers theory
Blum integers, 185
Chinese Remainder Theorem, 183–184
Euler's totient function, 183
Fermat's little theorem, 183
Galois field, 185
generators, 185
inverse modulo of number, 182–183
Jacobi symbol, 184–185
Legendre symbol, 184
maximum common divisor, 182
modular arithmetic, 181–182
prime number, 182
quadratic residues, 184
resolution for coefficients, 183

O

Omnidirectional antennae, 503
One-time pad, 143–144
One-way functions, 147–148, 165
One-way hash functions
birthday attack, 203
characteristics, 203
description, 148
message authentication code, 205
public-key algorithms, 204
secure hash value, 204
symmetric block algorithms, 204
Open key authentication, 515–516
OpenPuff, 265
Open System Interconnection (OSI) model
application layer, 13
data link layer, 13
diagram of, 13
layer network, 13
physical layer, 12–13

presentation layer, 13
session layer, 13
transport layer, 13
vs. TCP/IP model, 15
Operating systems, steganography in, 246
Optical fibre, 47–48
Orthogonal frequency division multiplexing (OFDM), 63
OSI model. *See* Open System Interconnection (OSI) model
Output-feedback mode (OFB), 174

P
Packet data optimised (PDO) technology, 493
Packet data unit (PDU), 470, 484
Packet switching digital technology, 16
Pair transient key (PTK), 529
Pairwise master key identifier (PMKID), 531
Pairwise temporal key (PTK), 534
Password Authentication Protocol (PAP), 521
Password management policies, 548–549
P-box permutation, 193
Peer-to-peer communication mode, 3, 5
Perl language, 396–397
Personal area networks (PANs), 4
bluetooth
application and presentation level, 469
authentication process, 471–472
baseband level, 469
cipher, 472, 473
data link level, 468
elements, 467
encryption keys, 472–473
frequency selection module, 470
key negotiation, 473
L2CAP connection, 470–471
master unit, 470
network level, 468–469
physical layer, 468
security architecture, 467–469
security manager, 471
session layer, 469
Special Interest Group (SIG), 467
specifications, 467
spoofing process, 470
transport level, 469
vulnerability of, 474
infrareds, 474
ultra-wide band, 474
Zigbee, 474–475
Personal computer (PC), 478
Personal digital assistant (PDA) devices, 512
Personal wireless networks. *See* Wireless networks
P frames (predictive), 135
Phase shift keying (PSK), 571
Phonemes and phones, 565, 567
Physical layer, 12–13, 456–457
analogue modulation
amplitude modulation, 34–35
frequency modulation, 35–36
phase modulation, 36–37
analogue signals and data, 32–33
digitisation, 40
multiplexing
analogue and digital connection, 45
electromagnetic spectrum, 44
frequency division multiplexing, 40–45
light propagation, 43
medium distance call, 44
QPSK constellation, 45
telephone network, 44
time division multiplexing, 42, 46
total refraction and reflection, 43
wavelength division multiplexing, 42, 45–46
non-periodical signals, 26–28
numerical data encoding, 33–34
numerical signals modulation, 37–38
periodic signals, 25–26, 29–30
quadrature amplitude modulation, 38
sampling and digitising, 38–40
spectral representation, of signals, 26, 28–29
transmission channel
bandwidth, of signal, 30
maximum speed, 30–31
signal alterations, 31–32
wireless networks, 456–457
Physical security, 551
PKZIP, 200
Plaintext/cleartext, 137
Plosive sounds, 566
Point coordination function (PCF), 463
Point-to-point service, 479
Polarisation, 500
Portable document scanners, 603
Portable scanners, 512
Port Address Translation (PAT), 322
Porta, Giovanni, 234
Post Office Protocol 3 (POP3), 112
Presentation layer, 13, 109
Pretty Good Privacy (PGP), 229
Prime number, 182
Prime numbers generation, 186
Privacy Enhanced Mail (PEM), 228
Private watermarking, 272–273
Probe request frame, 459
Probe response frame, 459
Professional scanners, 604
Protected access credential (PAC), 526

Protected Extensible Authentication Protocol (PEAP), 526
Protection archives, 160
PROXIABLE flag, 374
Proxy servers, 305–306
Pseudo-random sequences generators
  additive generators, 200
  congruent linear generators, 197–198
  multiple streams, 200–201
  PKZIP, 200
  shift records, 198–199
  stream ciphers, 199–200
PSH (Push), 308
PSPACE class, 181
Public access management policies, 550
Public-key algorithms/asymmetric algorithms
  communication by, 148–149
  description, 139
  digital signature algorithm, 217–219
    message sign, 218
    NIST proposal, 217–218
    operation, 218
    via discrete logarithms, 219
  elliptic curve, 217
  key management, 171
  length of, 164
  RSA algorithm, 215–217
Public-Key Cryptographic Standards (PKCS), 230
Public key steganography, 240–241
Public switch telephone network (PSTN), 51, 589
Public watermarking, 273
Pure steganography, 239

Q
Quadrature amplitude modulation (QAM), 38
Quantum cryptography, 222–223

R
Radio frequency, 594–595
Radio frequency identification (RFID), 456
Radio network subsystem (RNS), 488
Radio transmissions, 445
Radio waves transmissions, 49
RADIUS. *See* Remote Authentication Dial In User Service (RADIUS)
Random sequence generators, 153
  computer clock, 202
  distillation of randomness, 203
  keyboard latency typing, 202
  polarisation and correlation, 202
  random noise, 201–202
Raster analysis/raster analysis and identification (RAID), 618
Rational schema, AES standard, 211–212
Realm key, 370–371
Real-Time Protocol (RTP), 590
Redundant Array of Inexpensive Disk (RAID), 427–428
Reference models
  ISO OSI model, 11–13
  TCP/IP model, 13–14
Remote Authentication Dial In User Service (RADIUS), 323, 541, 544
Repeater, wireless network, 465, 466
Replacement ciphers, 141–142
Request to send (RTS), 461
Residual-excited linear prediction (RELP), 584
Return on investment (ROI), 506
Return-to-zero (RZ) encoding, 33
Rijndael, 205, 210, 264
Robust security network (RSN), 530–532
Robust security network information element (RSN IE), 529
Robust security standard, 529
Rogue access points, 452
Rotor machine, 142
Router-based VPN, 364
RSA (Rivest, Shamir and Adleman) standard, 144
RST (Reset), 308
Rubber-hose cryptanalysis, 140

S
Sandbox security model, 386
Satellites, 491
Satellite transmission, 50–51
S-boxes, 192
Scanners, 604–605
Scanning tools, 562
Schott, Gaspar, 234
Screened host firewall, 307
Screened subnet firewall, 307
Screening router, 299, 302
Scripts and security issues
  CGI standard, 392–395, 397–399
  languages, 395–396
  Perl language, 396–397
Second generation/digital voice/2G telephone network, 53, 476
The Second World War, steganography, 236
Secret communication
  operation, 237–238
  public key steganography, 240–241
  pure steganography, 239
  secret key steganography, 240
Secret key steganography, 240
Secret sharing, 159
Secure Electronic Transmission (SET), 381–382

Secure-HTTP (S-HTTP) protocol
applications, 330–331
bandwidth exchange, 329
characterisation, 327
creation of, 328
designs, 330
digital signatures, 331–332
encryption, 328–329
key exchange, 328–329
messages, 328
signature, 328
verification, 329–330
Secure Socket Layer (SSL), 499
authentication, 334
browsers and servers, 336–337
characterisation, 333
connections, 335–336
disadvantages, 337–338
encryption, 334–335
position in IP, 333
S/MIME, 338–339
Security policy
wired networks
acceptance and applicability, 296
adequate education, 441
assets to be protected, 442
confidential information, 437
confidentiality, 297
corporate management, 432–433
cost of attack, 295
email account, 438–439
good security policy, 296
implementation, 433
intellectual resources, 294
network attacks, 295
organisation resources, 436
physical resources, 294
privacy, 441
privileges and administration password, 440–441
process, 442–443
protection and process, 433, 434
resources of perception, 294
risk elements, 436–437
time resources, 294
users access, 440
users information, 434–435
vulnerabilities, 438
wireless networks
access policies, 549–550
drafting of, 546
guidelines, 546
impact analysis, 548
password management policies, 548–549
physical security, 551
procedures, 546
public access management policies, 550
risk assessment, 547–548
standards, 545–546
Security tools, 563
Selective steganography, 238
Self-synchronising stream ciphers, 173
Semiprivate/semi-blind watermarking, 273
Server disasters
application server provider, 431
clustering, 429–430
continuity groups, 427
data backup, 430–431
RAID, 427–428
redundant servers, 428–429
server recovery, 431–432
Service Discovery Protocol (SDP), 470
Service set identifier (SSID), 463, 516
Services stations, 64–65
SESAME, 225
Session layer, 13, 108–109
Shared key authentication, 514–515
Shielded twisted pair (STP), 47
Shift factor (SF), 576
ShiftRows transformation, 209, 211
Short message-application layer (SM-AL), 480
Short message-link layer (SM-LL), 480
Short message-relay layer (MS-RL), 480
Short message service (SMS), 478–495, 601
Short message service centre (SMSC), 479
Short message service-GatewayMSC (SMS-GatewayMSC), 479
Short message service-Internet working MSC (SMS-IWMSC), 479
Short message-transfer layer (SM-TL), 480
Short Messaging Entity (SME), 480
Signal Intelligence (SIGINT), 618
Signal-to-noise ratio (SNR), 448–449
SilentEye, 270
Simple Network Management Protocol (SNMP), 343–344, 544
Single loss expectation (SLE), 547–548
Single-sideband modulation (SSB), 604, 605
Single-use tab, 143–144
Sinusoidal model, 585
Site analysis, wireless networks
execution of, 506–508
financial controls, 508–509
technical standards, 508
Sliding window, 579
Smart card, 229
Smart phone, 512–513
S/MIME, 338–339
Smurf and *spoofing* attacks, 418–419

Sniffing tools, 562
SNMP, 544
Social engineering, 451–452
Special Interest Group (SIG), 467
Spectrograph, 586–587
Spectrum analysers, 607–608
Speech spectrum, 574
Spoofing, 450–451
Spoofing attack, 353–355
Spread spectrum systems, 254–256
  advantages, 256
  band expansion, 254
  direct sequence and frequency hopping, 254
  theory of, 255
Stamping services
  arbitration solution, 160
  improved arbitration solution, 160–161
Stateful filtering, 317
States
  AddRoundKey transformation, 210
  input and output, 207
  MixColumns transformation, 209
  operation schema, 208
  ShiftRows transformation, 209
  SubBytes transformation, 209
Static filtering packet, firewalls
  complications, 310
  ICMP, 308–310
  mean transfer unit, 310–311
  packet filter, 307–308
  TCP traffic, 308
  traffic flags, 308
Static Network Address Translation (static NAT), 322
Static web documents, 120
Station-station protocol, 221
Statistical steganography, 256
Steganography
  active hackers
    and resilient steganography, 244
    supraliminal channels, 244–245
  adaptive and non-adaptive algorithms, 243
  attacks and analysis, 257–258
  data noise, information concealment in, 242–243
  distortion methods
    digital images, 257
    written texts, 256–257
  domain transformation methods
    concealing information and data compression, 254
    DCT domain, 252–253
    digital sound, 253–254
  hidden messages, 258–259
  history
    air mail, 236
    Auguste Kerckhoffs, 235
    Bishop John Wilkins, 235
    Blaise de Vigenere, 235
    The Chinese, 234
    The Egyptians, 233
    The First World War, 236
    Gaspar Schott, 234
    George Washington, 235
    Giovanni Porta, 234
    GirolamoCardano, 234
    The Greeks, 233–234
    Johannes Trithemius, 234
    Margaret Thatcher, 237
    Mary Queen of Scots, 235
    The Second World War, 236
    The Vietnam War, 237
  invisible communication, 246
  malicious hackers, 244–245
  models, 238–239
  practical examples
    cryptapix, 260
    data stash, 261
    hermeticstego, 262
    hide in picture, 263–264
    OpenPuff, 265
    SilentEye, 270
    S tools, 266–269
  secret communication
    operation, 237–238
    public key steganography, 240–241
    pure steganography, 239
    secret key steganography, 240
  security systems
    perfect security, 241
    secrets messages detection, 242
  spread spectrum systems, 254–256
  statistical methods, 256
  substitution methods
    in binary images, 250–251
    cover areas and parity bits, 249
    least significant bits, 248
    palette-based images, 249–250
    pseudo-random permutation, 248
    quantisation and dithering, 250
    reserved/unused space, on computers, 251
  vs. cryptography, 233
  written text, information concealment in, 245–246
Stego-key, 237, 247
Stego-object, 237–238, 247
Stethoscopic microphones, 602
S tools
  Data Encryption Standard (DES), 266
  International Data Encryption Algorithm (IDEA), 267

MDC, 268
triple DES, 269
Store-and-forward/packet-switched operation, 6–7
Stream ciphers
A5, 199–200
block ciphers and, 174–175
design and analysis, 199, 200
LFSR based, 199
synchronising, 172–173
Streaming audio, 126–127
STU-III, 224
SubBytes transformation, 209, 211
Subnet, 6–7
Subscriber identity module (SIM) card, 476, 596
Substitution steganography
in binary images, 250–251
cover areas and parity bits, 249
least significant bits, 248
palette-based images, 249–250
pseudo-random permutation, 248
quantisation and dithering, 250
reserved/unused space, on computers, 251
Substitutive steganography, 238
Symmetric key algorithms
communication by, 147
description, 139
length of, 162–164
SYN (Synchronise), 308
Synchronous stream ciphers, 174

T

Tablet computers, 511–512
TCP/IP model. *See* Transfer Control Protocol/Transmission Control Protocol/Internet Protocol (TCP/IP) model
TCP protocol hijack, 348
Teardrop attacks, 417–418
Telephone bugging devices, 599
Telephone network, 16
Telnet, 544
Telocator Alphanumeric Protocol (TAP), 485
Temporal Key Integrity Protocol (TKIP), 529, 532–533
Temporary Mobile Subscriber Identity (TMSI), 488
Terminal Emulation Link NETwork (TELNET), 14
Terrestrial trunked radio (TETRA), 492–495
Text Mode, 484
Thatcher, Margaret, 237
Third-Generation Partnership Project (3GPP), 485
Third generation/voice and digital data/3G telephone network, 55
Three-packet handshake attacks, 417
Time division multiplexing (TDM), 42, 46
Time domain scrambler (TDS), 577
Time element scrambling, 580
Timer synchronisation function (TSF), 464
TP-data-coding-scheme (TP-DCS), 481
TP-more-message-to-send (TP-MMTS), 482
TP-user-data-header-indicator (TP-UDHI), 482
TP-user-date (TP-UD), 482
TP-valid-period (TP-VP), 482
Transfer Control Protocol/Transmission Control Protocol/Internet Protocol (TCP/IP) model, 589
application layer, 14–15
host-to-network layer, 15
internet layer, 14
transport layer, 14
transport protocol
arrival time density, 103–104
1-bit flags, 101
congestion management, 102
data exchanges, 100
main ports, 100
structure, 101
vs. OSI model, 15
Transform-based scrambling (TBS), 576
Transient Electromagnetic Pulse Emanation Standard (TEMPEST), 617–619
Transmission over guided media
coaxial cable, 47
optical fibre, 47–48
twisted pair, 46–47
Transport layer, 13
application, 96–97
connection-oriented service, 96
DISCONNECT, 97
network performance
fast processing, 107
measurement, 105–106
performance problems, 104–105
protocols for high-performance, 107–108
system design, 106–107
TCP transport protocol
arrival time density, 103–104
1-bit flags, 101
congestion management, 102
data exchanges, 100
main ports, 100
structure, 101
TPDU, 98–99
transport and network, 96–97
TSAP and NSAP, 97–98
UDP transport protocol, 99
Transport Protocol Data Unit (TPDU), 97–98, 480
Transport service access points (TSAPs), 97
Transposition ciphers, 141–142
Triple encryption, 197

Trithemius, Johannes, 234
Trojan horses, 406
Trusted Information System (TIS)/PEM, 228
Twisted pair, 46–47

U
UDP transport protocol, 99
Ultrasonic bugging devices, 599
Ultra-wide band (UWB), 474
Unicast mode, 4
Uniform Resource Locator (URL), 114, 118
Unipolar encodings, 33
Universal Mobile Telecommunications System (UMTS), 478, 487–491, 596
Universal serial bus (USB) pen drives, 602
UNIX
  based firewalls, 318–319
  virus protection, 411
Unshielded twisted pair (UTP), 47
Unvoiced sounds, 566
URG (Urgent), 308
User Datagram Protocol (UDP), 14, 590
User data header length (UDHL), 482
User data length (UDL), 482
User equipment (UE), 488
US Federal Communications Commission (FCC), 457
US National Science Foundation NET (NSFNET), 16–17

V
Very low frequency (VLF) band, 598
Video, 129–131
  chat programs, 616–617
  communications systems, steganography in, 246
  compression
    coding and decoding, 131
    data compression, 131
    JPEG algorithm, 131–133
    MPEG algorithm, 133–136
    redundancies, in film, 134
  images, digital watermarking in, 285
  recorders, 602
The Vietnam War, steganography in, 237
Vigenére cipher, 142
Virtual Local Area Network (VLANs), 79, 463
Virtual private network (VPN), 520, 590
  authentication, 363
  compliance, 364
  configurations, 361
  cryptography, 363–364
  design, 361
  disadvantages, 362
  example, 360
  firewall-based, 364
  network diagram, 362
  policies, 556
  remote access modem, 361–362
  risk factors, 363
  router-based, 364
  settings, 365
  software- and hardware-based, 364–365
  technologies, 553–554
  wireless architectures, 554–556
Visitor location register (VLR), 480
Voice configuration
  classic source-filter model, 567–568
  general source-filter model, 568–569
  linear prediction modelling, 569–570
Voice cryptanalysis
  analogue methods, 587
  digital ciphers, 588
  linear prediction vocoder cryptanalysis, 588
  spectrograph, 586–587
  tools and parameters, 586
Voiced sounds, 566
Voice organs, 565, 566
Voice over IP (VoIP) systems, 127–129
  authentication mechanism, 591
  availability aspects, 590–591
  communication vulnerabilities, 590
  confidentiality aspects, 590
  DoS, 591
  firewalls, 591
  fixed networks, 590
  hardware vulnerabilities, 590
  human vulnerabilities, 590
  integrity aspects, 590–591
  Internet, 588–599
  IP datagram packets, 591
  natural vulnerabilities, 590
  physical vulnerabilities, 590
  public switch telephone network, 589
  sniffers, 591
  software vulnerabilities, 590
  TCP/IP, 589, 590
  telephone networks, 589
  traffic and features, 589
  virtual private network, 590
  wireless network, 590
Voice security
  cryptanalysis
    analogue methods, 587
    digital ciphers, 588
    linear prediction vocoder cryptanalysis, 588
    spectrograph, 586–587
    tools and parameters, 586
  digital encryption, 580
  signal analogue encryption

frequency domain, 573–576
time and frequency domain, 579–580
time domain/time domain scrambler, 577–579
transformation, 576–577
signal transmission
continuous synchronisation, 571–572
initial synchronisation, 570–571
source encoding
channel vocoder, 581–582
formant vocoder, 581
sinusoidal model, 585
standards, 585
vocoder, linear prediction, 582–584
spoken language
language structure, 567
phonemes and phones, 565, 567
voice organs, 565, 566
voice configuration
classic source-filter model, 567–568
general source-filter model, 568–569
linear prediction modelling, 569–570
VoIP systems (*See* Voice over IP (VoIP) systems)
Voice signal analogue encryption
frequency domain, 573–576
time and frequency domain, 579–580
time domain/time domain scrambler, 577–579
transformation, 576–577
Voice signal transmission
continuous synchronisation, 571–572
initial synchronisation, 570–571
Voice source encoding
channel vocoder, 581–582
formant vocoder, 581
sinusoidal model, 585
standards, 585
vocoder, linear prediction, 582–584

W

WAN-type wireless network technology
cellular phone technology
code division multiple access, 476
first generation, 475
GSM standard, 476–478
MMS service, 485–487
second generation, 476
SMS service, 478–485
UMTS standard, 487–491
GPS technology, 491–492
TETRA technology, 492–495
wireless application protocol
effectiveness, 496–497
ISO/OSI, TCP/IP and WAP models, 495–496
security architecture, 499
security level, 497–499
War chalking, 454
War dialing, 455
War drivers, 453, 455–456
War flying, 454
War walker, 454
Washington, George, 235
Watermarking, digital. *See* Digital watermarking
Wavelength division multiplexing (WDM), 42, 45–46
Web browser
ActiveX components, 389–391
attacks, 389
cookies, 391–392
HTML tags, 388
Web page information, 388
White hat hacker, 293
Whitening, 197
Wide area networks (WANs), 6–7
Wi-Fi phones, 513
Wi-Fi Protected Access (WPA), 67, 528–529, 543
WiGig, 73
Wilkins, John, 235
Window hopping, 578
Wired Equivalent Privacy (WEP), 458–459, 516–518, 538–540, 590
Wired networks security
attacks
analysis, 411–416
execution of, 416–419
prevention, 420
audit trails, 382–383
authentication, 358–360
commercial transactions, on Internet
authentication, 376
buyer additional information, 379–380
confidentiality, 377
credit cards, 380–381
digital signatures, 376–377
electronic cash, 377–378
plaintext, 376
Secure Electronic Transmission, 381–382
computer viruses
bomb, 404–405
concealment, 402–404
considerations, 400
developments, 400
prevention, 406–409
protection, 409–411
replication, 400–402
trojan horses, 406
worm viruse, 405–406
disaster prevention and recovery
division of, 421
network disasters, 421–426

server disasters, 427–432
simulation, 432
firewall (*See* Firewall)
hackers, 293
intrusion detection
advantages, 339
configuration, 344–346
data packet, 340
and firewall, 341
fusion of, 343–344
installation on host, 342–343
interruption of session, 341
limitations, 339–340
non-integral protection, 342
Java language
applets, 385
application programming interfaces (APIs), 387
developments, 384–385
Java Archive (JAR), 387
persistent objects, 386
sandbox security model, 386
strings, 387
Kerberos keys exchange
access an archive, 374–375
authentication, 368
distributed system, 365–366
inter-realm key, 370–371
operating mechanism, 366
plaintext request, 368–370
realm key, 370–371
ticket flags, 372–374
tickets comparision, 367
transmission mode, 368
trusted server, 365–367
vulnerability, 375–376
network attacks
active desynchronisation attack, 349–353
denial-of-service (DoS) attack, 346
hyperlink *spoofing,* 355
number sequence anticipation attack, 346–348
sniffer attacks, 348–349
spoofing attack, 353–355
TCP protocol hijack, 348
web *spoofing,* 355–358
policies and risk analysis
acceptance and applicability, 296
confidentiality, 297
cost of attack, 295
good security policy, 296
intellectual resources, 294
network attacks, 295
physical resources, 294
resources of perception, 294
time resources, 294
scripts and security issues
CGI standard, 392–395, 397–399
languages, 395–396
Perl language, 396–397
secure-HTTP (S-HTTP) protocol
applications, 330–331
bandwidth exchange, 329
characterisation, 327
creation of, 328
designs, 330
digital signatures, 331–332
encryption, 328–329
key exchange, 328–329
messages, 328
signature, 328
verification, 329–330
Secure Socket Layer
authentication, 334
browsers and servers, 336–337
characterisation, 333
connections, 335–336
disadvantages, 337–338
encryption, 334–335
position in IP, 333
S/MIME, 338–339
security policy
adequate education, 441
assets to be protected, 442
confidential information, 437
corporate management, 432–433
email account, 438–439
implementation, 433
organisation resources, 436
privacy, 441
privileges and administration password, 440–441
protection and process, 433, 434, 442–443
risk elements, 436–437
users access, 440
users information, 434–435
vulnerabilities, 438
virtual private networks
authentication, 363
compliance, 364
configurations, 361
cryptography, 363–364
design, 361
disadvantages, 362
example, 360
firewall-based, 364
network diagram, 362
remote access modem, 361–362
risk factors, 363
router-based, 364

settings, 365
software- and hardware-based, 364–365
Web browser
ActiveX components, 389–391
attacks, 389
cookies, 391–392
HTML tags, 388
Web page information, 388
Wireless application protocol (WAP)
effectiveness, 496–497
ISO/OSI, TCP/IP and WAP models, 495–496
security architecture, 499
security level, 497–499
Wireless bridging, 465, 466
Wireless fidelity (Wi-Fi) technology, 595
WirelessHD, 73–74
Wireless high-definition interface (WHDI), 74
Wireless identity module (WIM), 499
Wireless LAN (WLAN)
802.11 a standard, 65–66
802.11 b standard, 66
802.11 g standard, 66
802.11 i standard, 66–67
802.11 multiple cell network, 22–23
802.11 n standard, 67–69
Bluetooth, 70–72
broadband wireless, 69–70
distribution services, 64
ethernet standard, 20
exposed station problem, 64
FHSS, 63
hardwares, 7–8
hidden station problem, 64
HIPERLAN, 72–73
HR-DSSS, 63
individual units, 20, 21
infrared, 63
OFDM, 63
periodic signals, 21
radio base station/AP, 20
services stations, 64–65
sinusoidal signals, 22, 23
transmission techniques, 62–63
WiGig, 73
WirelessHD, 73–74
wireless high-definition interface (WHDI), 74
Wireless LANs (WLANs), 422–423
Wireless networks, 7–8. *See also* Wireless LAN (WLAN)
access points attack tools, 563
access points device, 510–511
ad hoc mode, 464
analysis, 450
antenna
beam width, 501
diversity of, 502
Fresnel zone, 502–503
gain of, 500–501
multiple paths, 501–502
path loss, 501
polarisation, 500
types, 503–504
bridging, 465, 466
cellular telephone security, 452–453
cracking tools, 563
denial-of-service, 451
DoS tools, 563
electromagnetic wave propagation, 446–448
frame management
association request, 460
association response, 460
authentication, 459–460
Beacon, 459
carrier sense multiple access/collision avoidance, 460–462
disassociation and de-authentication, 460
distributed coordination function, 462–463
distribution system, 458
fragment and retry, 458
fragmentation, 462
interframe spacing, 463
point coordination function, 463
power management, 458
probe request, 459
probe response, 459
protocol version, 458
service set identifier, 463
type, 458
wired equivalent privacy, 458–459
hacking and hackers
bluejacking, 454
cordless phone driving, 455
motivation of, 453
war chalking, 454
war dialing, 455
war drivers, 453, 455–456
war flying, 454
war walker, 454
X10 driving, 455
hybrid tools, 562
implementation
audit phase, 510
certification, 510
cost estimation, 505
development and installation, 509–510
device verification, 509
investment evaluation, 505–506

network design, 509
requirement acquisition, 504–505
site analysis, 506–509
industrial, scientific and medical band, 457
information theory
availability, 450
confidentiality, 449–450
integrity, 450
infrastructure mode, 464–465
LAN standards, 466–467
malicious codes, 451
mesh networks, 466
mobile user devices
laptops, 511
PDA devices, 512
portable scanners, 512
smart phone, 512–513
tablet computers, 511–512
Wi-Fi phones, 513
modulation techniques, 457–458
personal area networks
bluetooth, 467–474
infrareds, 474
ultra-wide band, 474
Zigbee, 474–475
physical layer, 456–457
radio frequency identification, 456
repeater, 465, 466
Rogue access points, 452
scanning tools, 562
security (*see* Wireless security)
security tools, 563
signal-to-noise ratio, 448–449
sniffing tools, 562
social engineering, 451–452
spoofing, 450–451
WAN technology
cellular phone technology, 475–491
GPS technology, 491–492
TETRA technology, 492–495
wireless application protocol, 495–499
Wireless remote camera detectors, 614–615
Wireless security
802.11i
advanced encryption standard, 533
features, 533–534
robust security network, 530–532
temporal key integrity protocol, 532–533
EAP
EAP-FAST, 526–528
EAP-MD5, 522–523
EAP-TLS, 523–524
EAP-TTLS, 524–525
LEAP, 526
PEAP, 526, 527
false access point detection, 535
foundations, 516
history of, 514
open key authentication, 515–516
RADIUS, 520–521
security policy
access policies, 549–550
drafting of, 546
guidelines, 546
impact analysis, 548
password management policies, 548–549
physical security, 551
procedures, 546
public access management policies, 550
risk assessment, 547–548
standards, 545–546
shared key authentication, 514–515
SSID, 516
violation
breach technologies, 538–544
process of attack, 536–538
WAPI, 534–535
wired equivalent privacy, 516–518
wireless architecture
gateway, 556–558, 561
VPN, 553–556, 560–561
WEP, 551–552, 560
802.1x, 558–559, 561
WPA, 528–529
WPA2, 534
802.1x
applicant device, 519
authentication server, 518
authenticator, 519
EAPOL, 519–520
Wireless transmission
electromagnetic spectrum, 48–49
infrared and millimetre wave transmissions, 50
light wave, 50
microwave transmissions, 49–50
radio waves, 49
WLAN Authentication and Privacy Infrastructure (WAPI), 534–535
World Wide Web (WWW)
architecture, 114, 115
client side, 114–116
dynamic web documents, 120–122
history, 113
Hypertext Transfer Protocol, 122
Javascript, 121
performance improvement, 122–124

server side
cookies, 119
multi-thread type, 116–117
protocols, 117–118
server farm, 117
URLs, 118
static web documents, 120
WPA2, 66–67, 534
Written text, steganography in, 245–246

X
X. 25, 18
X.509, 226
X10 driving, 455
XOR operation/exclusive OR, 142–143

Z
Zigbee, 474–475

WITPRESS ...for scientists by scientists

Critical Infrastructure Security

Assessment, Prevention, Detection, Response

Edited by: ***F. FLAMMINI****, Italy*

Critical Infrastructure Security: Assessment, Prevention, Detection, Response provides the most comprehensive survey yet of state-of-the-art techniques for the security of critical infrastructures (CI). It addresses both logical and physical aspects of security from an engineering point of view, and considers both theoretical aspects and practical applications for each topic. The book emphasises model-based holistic evaluation approaches as well as emerging protection technologies, including smart surveillance through networks of intelligent sensing devices.

Chapters investigate recently developed methodologies and tools for CI analysis as well as strategies and technologies for CI protection in the following strongly interrelated and multidisciplinary main fields: Vulnerability analysis and risk assessment; Threat prevention, detection and response; Emergency planning and management. Chapters are written by experts in the field, invited by the editors to contribute to the book. Researchers who participated are based at such institutions as Naval Postgraduate School, Argonne National Laboratory, Johns Hopkins University Applied Physics Laboratory, Pennsylvania State University, the University of Wisconsin, and SAIC.

The book can serve as a self-contained reference handbook for both practitioners and researchers or even as a textbook for master/doctoral degree students in engineering or related disciplines.

ISBN: 978-1-84564-562-5 eISBN: 978-1-84564-563-2
Published 2012 / 326pp / £132.00

WIT Press is a major publisher of engineering research. The company prides itself on producing books by leading researchers and scientists at the cutting edge of their specialities, thus enabling readers to remain at the forefront of scientific developments. Our list presently includes monographs, edited volumes, books on disk, and software in areas such as: Acoustics, Advanced Computing, Architecture and Structures, Biomedicine, Boundary Elements, Earthquake Engineering, Environmental Engineering, Fluid Mechanics, Fracture Mechanics, Heat Transfer, Marine and Offshore Engineering and Transport Engineering.

This page intentionally left blank