

UNIVERSITÀ DEGLI STUDI DI PADOVA



CORSO DI LAUREA IN
INGEGNERIA DELLE TELECOMUNICAZIONI

TESI DI LAUREA

Design, Implementation and Configuration
of an IPSec based Industrial WLAN

Renzi Marco Maria, Laureando

Prof. Sergio Congiu, Relatore

Prof. Alexandru Soceanu, Correlatore

Hr. Albert Link, Relatore in azienda

Hr. Ulrich Schwarz, Coordinatore in azienda

Padova, 20 Aprile 2004



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Alla mia famiglia.

Abstract

In this thesis we are concerned with the use of Wireless LAN for stacker control systems in industrial environments. Due to the insufficient degree of security allowed by the IEEE 802.11 standard (WEP), the use of IPSec encoding is taken into consideration.

This activity is part of a broader internship between the University of Padua (Telecommunication Engineering Faculty) in Italy, Regensburg Fachhochschule and Syskron GmbH in Germany.

The developed project is divided into two parts; the first one concerns with design and configuration of an IPSec industrial WLAN and it's developed in this thesis, while the second one, concerning with analysis and monitoring, is fully developed in the companion thesis.

This study observes how the IPSec use alters the behavioral course of events during handover and in cases of connection interruptions. An optimal network design is proposed, regarding which devices must take part to the VPN and from which participant is the IPSec encoding to be started.

The products currently available on the market are analyzed and the ones, that are most suited to the planned usage, are therefore deeply tested.

The monitoring of the entire Wireless LAN system is taken into consideration, concerning central management of configurations and updates, the avoidance of non-configured access points and clients, fast replacement of redundant versions for critical components.

A test system composed of the selected components is developed and fully functional; this project is now ready to be employed in a real industrial environment and to be sold on the market.

Sommario

Questa tesi riguarda l'uso di reti wireless WLAN per un sistema di controllo di mezzi mobili per le operazioni di carico e scarico merci in un ambiente industriale.

A causa dell'insufficiente grado di sicurezza introdotto dallo standard IEEE 802.11 con il WEP, è stato preso in considerazione l'utilizzo di VPN criptate tramite l'impiego del protocollo IPSec.

Questa attività è stata frutto di una collaborazione tra l'Università degli Studi di Padova (Facoltà di Ingegneria delle Telecomunicazioni), la Fachhochschule di Regensburg e la Syskron GmbH (Germania).

Il progetto svolto è stato diviso in due parti; la prima riguarda il design e la configurazione di una rete WLAN con crittografia IPSec ed è sviluppata in questa tesi, mentre la seconda parte, riguardante l'analisi ed il monitoraggio, è inclusa nella tesi "gemella".

Questo studio osserva come l'uso dell'IPSec possa interagire con il normale funzionamento di una rete wireless specialmente nel caso di handover e perdita di segnale. E' stato proposto un design ottimale di rete, riguardante quali device dovessero prender parte alla VPN e da quali partecipanti dovesse essere stabilito il tunnel IPSec.

Sono stati valutati i prodotti attualmente disponibili sul mercato e, quelli risultati maggiormente adatti alle esigenze richieste dal progetto, sono stati in seguito analizzati e testati.

Inoltre è stato preso in considerazione un sistema di monitoraggio dell'intera rete, comprendente un sistema di controllo per la configurazione e l'update del software, in modo da evitare la presenza di AP e client non correttamente configurati.

Infine è stato sviluppato un modello di rete completamente funzionante

usando le apparecchiature testate; questo progetto è quindi pronto per trovare applicazione in ambito industriale.

Acknowledgements

We would like to mention CMS GmbH, APE GmbH and PSP GmbH companies that supplied the test equipment we used to develop our project.

We would like to express our gratitude to the following people for their support and assistance in developing this thesis:

Albert Link and Ulrich Schwarz for all assistance and help during our stage in the company

Alexandru and Marion Soceanu for the chance they gave us for coming here in Germany

Prof. Sergio Congiu for the mail support during all the time we stayed in Germany

and also thanks to:

all Erasmus students in Fachhochschule Regensburg for all the good time we had, especially my danish friend Jeppe

all the students living in Thomaheim, especially: Gao Ya, Matthias, Dominique, Michael and Genoveva

all the friends from Verona, especially: Francisco (*tent*), Simone (*nicuzn*), Sergio (*sè*)

all the friends from the university of Padua, especially: Aronne, Massimo, Matteo, Federico, Nicola, Daniele

all the friends from the room in Padua: Claudio (*cmutinel*), Giordano (*Fox*) and Luigi (*gigi*)

all my parents and relatives: father Alberto, mother Lucia, brother Andrea, aunt Liana, uncle Tiziano (*zio_tecnologico*), cousin Silvia, grandfather Decimo and grandmother Maria

Ringraziamenti

Nel fare questo tipo di cose c'è sempre il rischio di dimenticare qualcuno... spero non sia questo il caso!

Innanzitutto un doveroso ringraziamento va ai miei genitori, papà Alberto e mamma Lucia, per avermi sostenuto in tutti questi anni, condividendo i periodi più difficili ma anche quelli più gioiosi. Senza il loro sostegno non sarei mai arrivato alla fine di questa lunga impresa.

Non posso non ricordare mio fratello Andrea, la zia Liana, lo zio Tiziano, mia cugina Silvia, il nonno Decimo e la nonna Maria per l'affetto e la comprensione dimostrata; avrete sempre un posto speciale nel mio cuore.

Inoltre

ringrazio Francisco per essermi sempre stato vicino e per la sincera amicizia che oramai ci lega da molto tempo.

Ringrazio Simone e Sergio per essersi dimostrati amici soprattutto nelle situazioni più difficili.

Ringrazio Claudio, Giordano e Luigi per i bei momenti passati assieme nell'appartamento di Padova.

Ringrazio tutti gli amici universitari di Padova che mi hanno accompagnato in questo lungo e faticoso viaggio, condividendo gioie, speranze e delusioni.

Ringrazio tutti i nuovi amici che ho trovato a Regensburg per le magnifiche serate trascorse assieme, in special modo Jeppe e Gao Ya.

Mario Mario Renzi

List of Abbreviations

WLAN	<i>Wireless Local Area Network</i>
ACK	<i>Acknowledgment</i>
AH	<i>Authentication Header</i>
AP	<i>Access Point</i>
ARP	<i>Address Resolution Protocol</i>
BPSK	<i>Binary Phase Shift Keying</i>
BSS	<i>Basic Service Set</i>
CRL	<i>Certificate Revocation List</i>
CTS	<i>Clear To Send</i>
DCF	<i>Distribution Coordination Function</i>
DES	<i>Data Encryption Standard</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DIFS	<i>Distributed Inter Frame Space</i>
DoS Attack	<i>Denial of Service Attack</i>
DS	<i>Distribution System</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
ESP	<i>Encapsulated Security Payload</i>
ESS	<i>Extended Service Set</i>
FHSS	<i>Frequency Hopping Spread Spectrum</i>
HMAC	<i>Hash Message Authentication Code</i>
ICV	<i>Integrity Check Value</i>

IETF *Internet Engineering Task Force*

IKE *Internet Key Exchange*

IP *Internet Protocol*

IPSec *IP Security*

ISAKMP *Internet Security Association and Key Management Protocol*

ISP *Internet Service Provider*

IV *Initialization Vector*

L2TP *Layer 2 Tunnel Protocol*

LAC *L2TP Access Concentrator*

LAN *Local Area Network*

LLC *Logical Link Control*

LNS *L2TP Network Server*

MAC *Medium Access Control*

MTU *Maximum Transmission Unit*

NAS *Network Access Server*

NAT *Network Address Translation*

PCF *Point Coordination Function*

PKI *Public Key Infrastructure*

PCMCIA *Personal Computer Memory Card International Association*

PDU *Protocol Data Unit*

POP *Point Of Presence*

PPP *Point-to-Point Protocol*

PPTP *Point-to-Point Tunneling Protocol*

QPSK *Quadrature Phase Shift Keying*

RF *Radio Frequency*

RTS *Request To Send*

SA *Security Association*

List of Abbreviations

SAP *Service Access Point*

SIFS *Short Inter Frame Space*

SPI *Security Parameter Index*

TCP *Transmission Control Protocol*

TOS *Type Of Service*

TTL *Time To Live*

UDP *User Datagram Protocol*

VPN *Virtual Private Network*

WEP *Wired Equivalent Privacy*

WLAN *Wireless Local Area Network*

List of Abbreviations

Table of Contents

Abstract	i
Acknowledgements	iv
List of Abbreviations	vi
1 The Company & the Project	1
1.1 The Company	1
1.1.1 Kronos	1
1.1.2 Syskron	3
1.2 The Project	4
2 IEEE 802 Protocols	7
2.1 IEEE 802	7
2.1.1 Protocol Architecture	7
2.1.2 LLC sublayer	9
2.1.3 MAC sublayer	11
2.2 IEEE 802.11	11
2.2.1 Protocol Architecture	11
2.2.2 Services	12
2.2.3 Medium Access Control - MAC	17
2.2.4 Physical Layer	31
2.3 IEEE 802.11b	31
2.3.1 Introduction	31
2.3.2 Overview	34
2.3.3 CCK	35
2.3.4 Walsh and Complementary Codes	38
2.3.5 Fast Transform Structure	39
3 The WEP Vulnerabilities	43
3.1 Introduction	43

TABLE OF CONTENTS

3.2	Attack Practicality	43
3.3	The Risks of Keystream Reuse	45
3.3.1	Decryption Dictionaries	49
3.3.2	Key Management	50
3.4	Message Authentication	50
3.4.1	Message Modification	51
3.4.2	Message Injection	52
3.4.3	Message Decryption	53
3.5	Fluhrer, Mantin, and Shamir Attack	59
3.6	Conclusions	60
4	IPSec and L2TP	61
4.1	Introduction on IPSec	61
4.2	IPSec modes	62
4.3	Authentication, Integrity and Confidentiality Services	64
4.3.1	ESP	64
4.3.2	AH	67
4.3.3	AH and ESP	69
4.4	Protocol Negotiation and Key Management	69
4.4.1	The Security Association - SA	69
4.4.2	Security Parameter Index - SPI	70
4.4.3	Internet Key Exchange - IKE	71
4.4.4	Negotiating the SA	78
4.5	IPSec Processing	79
4.5.1	Outbound Packet Processing	79
4.5.2	Inbound Packet Processing	80
4.6	IPSec Implementation	81
4.7	Performance Consideration	83
4.7.1	The Impact of IPSec on WLAN Performance	83
4.8	Vulnerabilities Affecting IPSec-protected WLANs	84
4.8.1	Confidentiality	85
4.8.2	Integrity	86
4.8.3	Resistance to Denial of Service Attacks	86
4.8.4	Traffic Flow Analysis	87
4.9	Conclusions on IPSec	89
4.10	Introduction on L2TP	91
4.11	Overview on L2TP	92
4.12	L2TP Frame Structure	93

TABLE OF CONTENTS

5	PKI and Digital Certificates	97
5.1	Overview	97
5.2	Understanding the Function of Components	98
5.3	Recognizing the vulnerabilities	100
5.4	Conclusions	100
6	Demo Brewery Network Design	101
6.1	Introduction	101
6.2	Network Equipment	101
6.3	Network Operation	103
6.4	Network Design	103
6.4.1	Channel Frequencies Allocation	104
6.5	Security Implementation	106
6.6	Products Selection	108
6.6.1	Available products on the market	109
7	Orinoco AP-500 test	113
7.1	Introduction on Orinoco AP Manager	113
7.2	Remote Monitoring	113
7.2.1	System tab	115
7.2.2	Remote tab	118
7.2.3	Wireless tab	119
7.3	Remote Configuring	120
7.3.1	Wireless Interfaces tab	120
7.3.2	SNMP tab	122
7.3.3	Remote update	123
7.4	Introduction on Orinoco AP-500 MIB	123
7.5	MIB Groups	124
7.5.1	The System group	124
7.5.2	The Interfaces group	125
7.5.3	The Address Translation group	127
7.5.4	The IP group	128
7.5.5	The ICMP group	132
7.5.6	The TCP group	135
7.5.7	The UDP group	137
7.5.8	The EGP group	139
7.5.9	The Transmission group	141
7.5.10	The SNMP group	143
7.5.11	The PRIVATE MIB group	148

8 Nortel Networks Contivity 1010 test	151
8.1 Setup and Configuration	151
8.1.1 Overview	151
8.1.2 Devices	151
8.1.3 Configuration with a Wireless Client	153
8.1.4 Configuration with a Pocket PC	160
8.1.5 Configuration with a Wireless Printer	162
8.2 Test and Analysis	162
9 NCP Secure Communications test	163
9.1 Setup and Configuration	163
9.1.1 Overview	163
9.1.2 Devices	164
9.1.3 NCP Secure Server Installation	165
9.1.4 NCP Secure Server Configuration	166
9.1.5 NCP Secure Client Configuration	175
9.1.6 NCP Secure Client Manager	179
9.1.7 NCP Secure CE Client	183
9.2 Test and Analysis	184
10 SonicWALL TZ 170 test	185
10.1 Setup and Configuration	185
10.1.1 Overview	185
10.1.2 Devices	185
10.1.3 Configuration with a Wireless Client	187
10.1.4 Configuration with a Pocket PC	195
10.1.5 Configuration with a Wireless Printer	195
10.2 Test and Analysis	195
11 Conclusions	197
11.1 Components	197
11.2 Best Product Comparison	198
11.3 Final Considerations	201
Bibliography	203

List of Figures

1.1	The Krones Company distribution	2
1.2	The Krones Company structure	2
1.3	An example of filling production chain	3
2.1	IEEE 802 Protocol Layers Compared to OSI Model	8
2.2	IEEE 802 Protocols in context	9
2.3	LLC PDU in a generic frame format	10
2.4	IEEE 802 Architecture components	12
2.5	ESS structure example	13
2.6	Authentication & Association procedure	17
2.7	IEEE 802.11 Protocol architecture	18
2.8	Basic access mechanism logic used in DCF	19
2.9	Basic access mechanism used in DCF	21
2.10	MAC timing	22
2.11	MAC frame	23
2.12	WEP encryption	27
2.13	Block diagrams for WEP encryption and decryption	29
2.14	Shared key authentication	30
2.15	IEEE 802.11 PHY layer	31
2.16	Digital modulation of data with PRN sequence	34
2.17	Forming Walsh Codes by successive folding	38
2.18	Basic Fast Walsh Transform Block (BFWB)	40
2.19	Modified Walsh Transform	41
3.1	Hacker's targets	45
3.2	Forcing a known plaintext to be transmitted	48
3.3	Message modification	52
3.4	IP redirection	54
3.5	Reaction attack	57
3.6	Working of airsnort	60

4.1	Authentication Header in Transport and Tunnel Modes	63
4.2	Encapsulating Security Payload in the 2 Modes	64
4.3	The Encapsulating Security Payload format	65
4.4	The Authentication Header format	68
4.5	IKE Main mode	76
4.6	IKE Aggressive mode	77
4.7	IKE Quick mode	78
4.8	Outbound Packet Processing	80
4.9	Inbound Packet Processing	81
4.10	Host to host implementation of IPSec	82
4.11	Gateway to gateway implementation of IPSec	82
4.12	Host to gateway implementation of IPSec	83
4.13	L2TP tunnel details	92
4.14	L2TP packet frame	93
4.15	L2TP additional fields	95
4.16	L2TP AVP structure	95
5.1	The PKI major component	98
5.2	The PKI processing request	99
6.1	The Access Point special box	102
6.2	The mobile PC mounted on an forklift	102
6.3	The bar-code reader	102
6.4	The planimetry	103
6.5	The Access Points roaming	104
6.6	The optimal channel separation	105
6.7	The insufficient channel separation	105
6.8	The Channel Frequencies Allocation plan	106
6.9	The security concept	108
7.1	The main AP Manager window	114
7.2	The password request window	114
7.3	The system information window	115
7.4	The Intervals window	116
7.5	The Select a Remote Link Partner for 129.123.20.200 window .	117
7.6	The Remote Link Test window	117
7.7	The Remote statistics information window	118
7.8	The Wireless information window	119
7.9	The Wireless Security Setup: WEP option window	121
7.10	The Edit Access Point SNMP window	122
7.11	The Remote update option window	123

LIST OF FIGURES

7.12	The MIB tree	125
7.13	The System group	126
7.14	The Interfaces group	128
7.15	The at group	129
7.16	The ip group	133
7.17	The icmp group	135
7.18	The tcp group	137
7.19	The udp group	138
7.20	The egp group	140
7.21	The Transmission group	142
7.22	The snmp group	145
7.23	The dot1dBridge group	148
7.24	The PRIVATE MIB group	149
8.1	The gateway front view	153
8.2	The test configuration	153
8.3	The HyperTerminal program manage the gateway via serial	154
8.4	The Web Management interface	154
8.5	The Connectivity property of the <i>\Base\Control Tunnel</i> group	155
8.6	The IPSec property of the <i>\Base\Control Tunnel</i> group	156
8.7	The Contivity VPN Client	157
8.8	The Contivity VPN Client Monitor	157
8.9	The Address Pool assignment window	158
8.10	The Automatic Backup File Servers window	158
8.11	The interface LAN1 filters	159
8.12	The wireless test configuration with Pocket PC	160
8.13	The movianVPN client welcome and connection screen	160
8.14	The movianVPN client basic settings screens	161
8.15	The movianVPN client IKE and IPSec settings screens	161
8.16	The wireless printer configuration test	162
9.1	The test configuration	163
9.2	The SNMP Configuration window	166
9.3	The "General" folder	167
9.4	The "VPN" folder	168
9.5	The Secure Server Adapter "General" folder	169
9.6	The Secure Server Adapter "Pools" folder	169
9.7	The LAN Adapter 2 "General" folder	170
9.8	The LAN Adapter 3 "General" folder	170
9.9	The IPsec User "Basic Setting" folder	171
9.10	The IPsec User "Authentication" folder	172

9.11	The IPsec User "Security" folder	173
9.12	The IKE Policy	174
9.13	The IPSec Policy	174
9.14	The Client Monitor window	175
9.15	The Line Management window	176
9.16	The Incoming Calls window	176
9.17	The Security window	177
9.18	The Advanced window	177
9.19	The VPN Tunnelling window	178
9.20	The Firewall Settings window	178
9.21	The Connection established window	179
9.22	The user profile configuration	180
9.23	The Remote Administration (client side)	181
9.24	The Remote Administration (administrator side) example 1	182
9.25	The Remote Administration (administrator side) example 2	183
9.26	The NCP Secure CE Client	183
9.27	The NCP Secure CE Client Monitor	184
10.1	The SonicWALL front and rear view	187
10.2	The test configuration	187
10.3	The Network settings	188
10.4	The DHCP Server settings	188
10.5	The Intranet settings	189
10.6	The Firewall rules	190
10.7	The Firewall configuration	190
10.8	The VPN configuration	191
10.9	The Group VPN tab setting windows	192
10.10	The User setting windows	193
10.11	The Client "Status" and "Connection Status Details"	194
10.12	The Backup/Restore feature	194
10.13	The wireless printer configuration test	195

List of Tables

3.1	A decryption dictionary	49
6.1	The radio characteristics for 2.4GHz Frequency Band	105
7.1	The Errors to Bridge Packets ratio	119
11.1	The necessary components	197

LIST OF TABLES

Chapter 1

The Company & the Project

1.1 The Company

1.1.1 Krones

The Krones Group, headquartered in Neutraubling, Germany, develops and manufactures machines and complete lines for all categories of filling and packaging technology. Specialised technical knowledge, evidenced by more than 1,300 existing patents, a generous level of expenditure on research and development, high-precision manufacturing to stringent standards of quality, plus round the clock service support through more than 30 branch operations worldwide, coordinated by a corporate culture of continuous innovation - these are just some of the factors which have contributed to the company's success. The group comprises Krones AG itself and its subsidiaries Steinecker (brewing technology), Sander Hansen (pasteurising technology) and Syskron (process automation).

Krones started in 1951 with the manufacture of labelling machines, and nowadays operates on a global scale. Krones AG's product range includes not only labellers, but also stretch blow-moulding machines, fillers, mixers, inspectors, monitoring equipment, conveyors and control systems. The Contiform stretch blow-moulding machine developed in 1997 for making PET bottles marked the firm's debut in the dynamically expanding field of plastics technology. Increasingly, the Krones Group is entrusted with jobs involving the planning and installation of complete filling and packaging lines for glass and PET bottles, as well as for cans. The group's capabilities also include turnkey breweries and production facilities for the soft drinks industry. Its customers are breweries and soft drinks companies, wine, champagne, spirits and food producers, and the chemical, pharmaceutical and cosmetic industries as well.



Figure 1.1: The Krones Company distribution

The group has production facilities in Neutraubling, Nittenau, Flensburg, Freising and Rosenheim (all of them certified under DIN ISO 9001). Worldwide, Krones employs about 8,500 people. Exports account on average for approximately 80% of the lines delivered. Consolidated turnover totalled 1,305 million euros in 2002.

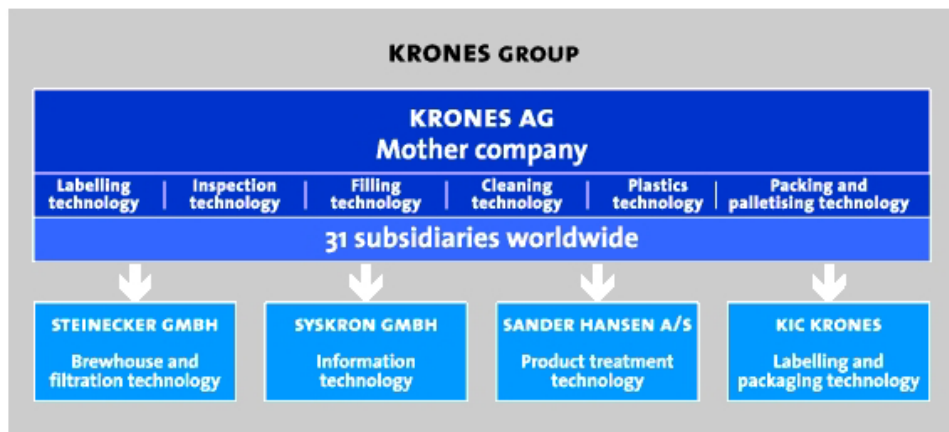


Figure 1.2: The Krones Company structure

1.1.2 Syskron

Syskron is a newly founded IT enterprise with a past rich in tradition. Originating from a combination of the former software departments of the firm Kronos and the Steinecker subsidiary BC Automation, Syskron is among the leading worldwide suppliers of system components for process technology and automation techniques.

The use of information technology presents a reliable instrument for the planning, analysis and optimization of production lines. If it is accompanied by a founded knowledge in plant construction, process technology and production logistics, the optimum basis is achieved for providing comprehensive advice.

Syskron GmbH offers such advice and has even created its own special field which uses intelligent software programmes and years of experience in the construction and running of production machines in order to analyze, assess and optimize planned and existing machines.



Figure 1.3: An example of filling production chain

The production programme contains special service packages which persistently increase the quality of decisions made by line proprietors, both in the planning stage and during production.

Syskron plans, develops and realizes control systems for production, filling, packaging and logistics in the foodstuffs and luxury foods industry. In the brewery and beverage field, their plant concepts and process analyzes, adapted to the respective requirements, have been setting standards for a quarter of a century. And for several years now, Syskron has also been offering its innovative products and services successfully for the pharmaceutical and chemical industry.

1.2 The Project

Our project was developed in Freising by Syskron GmbH from October 2003 to February 2004; the dissertation was about:

The Use of Wireless LAN for Stacker Control Systems in Industrial Environments.

The following themes will be taken into consideration:

The Use of IPSec Encoding for Stacker Control Systems in Industrial Environments

Due to the large warehouse surface areas covered, there is an high risk that data communication may be tapped. However, the available WEP encoding permits an insufficient degree of security. Yet because of the mobile devices employed, the performance that the IPSec client requires is strictly limited.

The special conditions that concern the use of wireless LAN for stacker control systems should hereby be taken into consideration.

On one hand, wireless LAN was not initially designed for usage on vehicles. Therefore, one should observe how the IPSec implementation alters the behavioral course of events during handover and in cases of connection interruptions.

On the other hand, with 30 access points, only approximately 30 clients can be serviced and, for this reason, solutions for office usage are often inappropriate.

The following points must be examined:

- *Which products are available on the market? Are they suited to the planned usage?*

A list of the available solutions and their corresponding characteristics are described in chapter 6 *Demo Brewery Network Design*, section 6 *Product Selection*.

- *From which participant is the IPSec encoding to be started (server, switch or access point)?*

The IPSec tunnel is established between a mobile device and the gateway; more detailed informations are available in chapter 6 *Demo Brewery Network Design*, section 5 *Security implementation*.

- *Is the Pocket PC devices integration with IPSec possible?*
Yes; more detailed information are available in the tests of the devices taken into consideration (chapter 8 *Nortel Network Contivity 1010 test*, chapter 9 *NCP Secure Communications test* and chapter 10 *SonicWALL TZ 170 test*).
- *How will the simultaneous use of devices that are not IPSec compatible, such as wireless bar code readers, WLAN printers and APs, still be possible?*
For a complete explanation, see chapter 6 *Demo Brewery Network Design*, section 5 *Security implementation* and the chapters of the tests (chapter 8 *Nortel Network Contivity 1010 test*, chapter 9 *NCP Secure Communications test* and chapter 10 *SonicWALL TZ 170 test*).

Monitoring of Wireless LAN Systems

For the monitoring of the entire wireless LAN system, a tool that enables permanent monitoring of the system is required. Error messages must thereby be drawn up in such a manner that they are useful both for the manager of the plant as well as for subsequent diagnosis by network specialists.

The following points must be examined:

- *Can the currently used network and server monitoring software provide the necessary information?*
The currently used network and server monitoring software provide information about one device (AP) at time; it doesn't allow a complete overview of the entire network. More detailed informations are available in chapter 7 *Orinoco AP-500 test*.
- *Are there more suitable products available on the market?*
No.
- *What information is supplied by the access point and the clients?*
More detailed informations are available in chapter 7 *Orinoco AP-500 test*.
- *Can information be collected concerning handover and connection interruptions?*
No.

- *What information is supplied by the IPSec gateway?*
The IPSec gateway supplies information about the device status (DHCP leases, current active VPN tunnels, etc.) and eventual configuration problems.

Management of the Configuration and Fail-Safety of the Wireless LAN System

A type of central configuration management should enable the user to replace defective devices in case of failures and then to adjust the configuration. Here, particular care must be taken to assure the aspects of security.

The following points must be examined:

- *Central management of configurations and updates*
- *The avoidance of non-configured access points and clients*
- *Fast replacement or redundant versions for critical components*

These three points will be examined in the following chapters.

A test system is planned to be assembled, composed of selected components. It is thereby to be determined whether the system can be used in practical situations, to what extent it corresponds to the desired intention and which are the areas that must still be improved.

Chapter 2

IEEE 802 Protocols

2.1 IEEE 802

2.1.1 Protocol Architecture

This architecture was developed by the IEEE 802 committee and has been adopted by all organizations working on the specification of LAN standards. It is generally referred to as the IEEE 802 reference model. Working from the bottom up, the lowest layer of the IEEE 802 reference model corresponds to the physical layer of the OSI model and includes such function as

- Encoding/decoding of signals
- Preamble generation/removal
- Bit transmission/reception

In addition, the physical layer of the 802 model includes a specification of the transmission medium and topology. Generally, this is considered "below" the lowest layer of the OSI model. However, the choice of transmission medium and topology is critical in LAN design and so a specification of the medium is included. Above the physical layer there are the functions associated with providing service to LAN users. These includes the following:

- On transmission, assemble data into a frame with address and error detection fields.
- On reception, disassemble frame, and perform address recognition and error detection.
- Govern access to the LAN transmission medium.

- Provide an interface to higher layers and perform flow and error control.

These are functions typically associated with OSI layer 2. The set of functions are grouped into a *logical link control (LLC)* and in a separate layer called *medium access control (MAC)*. The separation is done because the logic required to manage access to a shared-access medium is not found in traditional layer 2 data link control and also because for the same LLC, several MAC options may be provided.

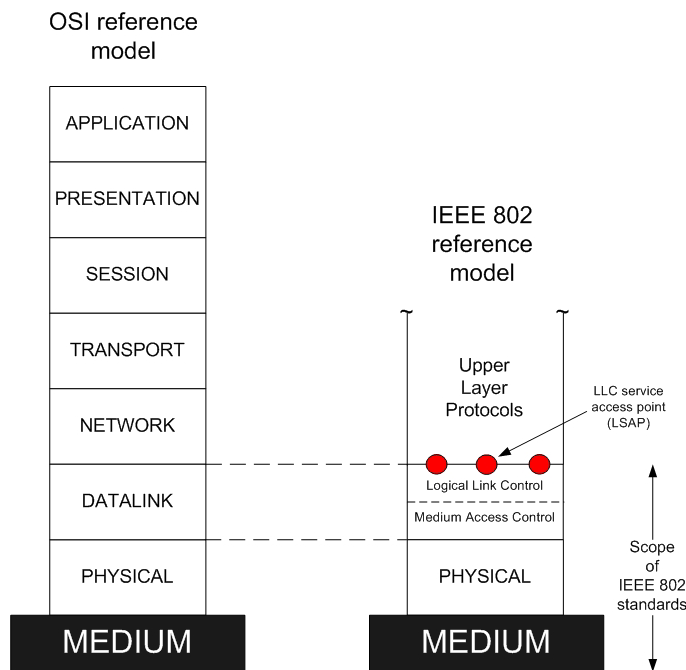


Figure 2.1: IEEE 802 Protocol Layers Compared to OSI Model

Service Access Points (SAPs)

Multiple *link service access points (LSAPs)* provide interface ports to support multiple higher layer users. The MAC sublayer provides a single interface port to the LLC sublayer. The Physical layer provides an interface port to a single MAC station. A user of LLC is identified by, at a minimum, the logical concatenation of the MAC address fields and the LLC address fields (LSAPs) in a frame. Figure 2.1 illustrates the relationship between the levels of the two architecture.

Higher level data are passed down to LLC, which appends control information as a header, creating an LLC *protocol data unit (PDU)*. This control

information is used in the operation of the LLC protocol. The entire LLC PDU is then passed down to the MAC layer, which appends control information at the front and back of the packet, forming a MAC frame as shown in figure 2.2. Again, the control information in the frame is needed for the operation of the MAC protocol.

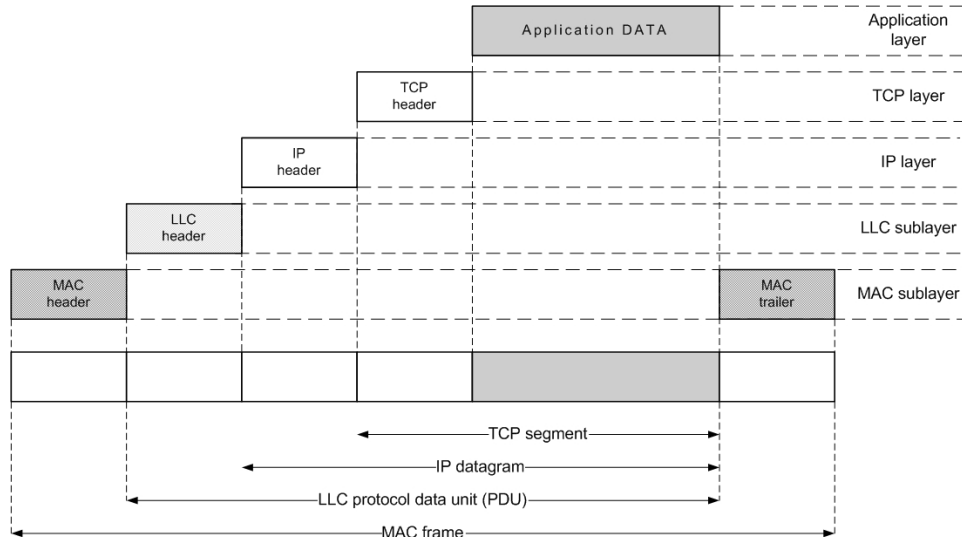


Figure 2.2: IEEE 802 Protocols in context

2.1.2 LLC sublayer

The LLC sublayer standard describes three types of operations for data communication between service access points:

- *unacknowledged connectionless service* - **type 1 operation**.
With type 1 operation, information frames are exchanged between LLC entities without the need for the prior establishment of a logical link between peers. These LLC frames are not acknowledged, nor are there any flow control or error recovery procedures.
- *connection-oriented service* - **type 2 operation**.
With type 2 operation, a logical link is established between pairs of LLC entities prior to any exchange of information frames. In the data transfer phase of operation, information frames are transmitted and delivered in sequence. Error recovery and flow control are provided.

- *acknowledged connectionless service - type 3 operation.*

With type 3 operation, information frames are exchanged between LLC entities without the need for the prior establishment of a logical link between peers. However, the frames are acknowledged to allow error recovery and proper ordering. Further, type 3 operation allows one station to poll another for data.

As shown in figure 2.3, all three LLC operations employ the same PDU format, which consists of four fields:

- *Destination SAP (DSAP)*: contains a 7-bit address and specify the destination user of LLC.
- *Source SAP (SSAP)*: contains a 7-bit address and specify the source user of LLC.
- **LLC control**: contains LLC control data (data flow control etc.); it's 1 byte in length when used connectionless services and 2 bytes in length when used connection-oriented services.
- **Information**: this field indicates which type of operation may be provided for that particular SSAP. For a SAP that supports type 2 operation, and for a particular connection, the information field also includes the receive window size used in the sliding-window mechanism. The field's size is variable, but it must be an integer multiple of 1 byte.

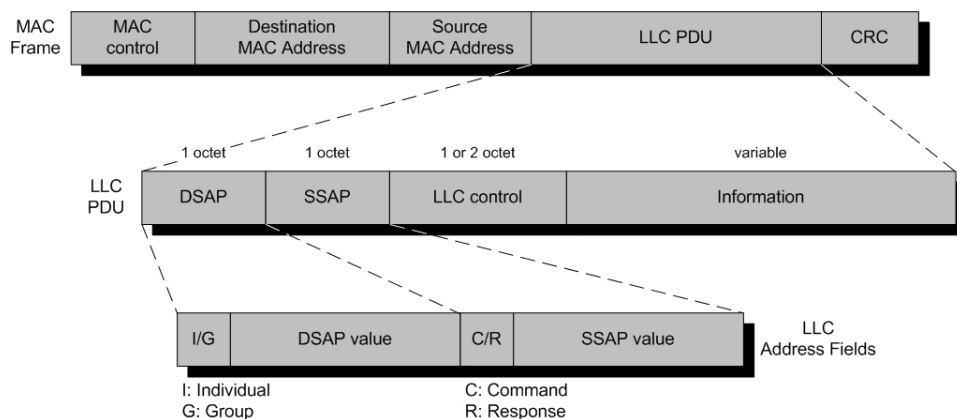


Figure 2.3: LLC PDU in a generic frame format

2.1.3 MAC sublayer

The MAC sublayer performs access control functions for the shared medium in support of the LLC sublayer. For different applications, different MAC options may be required. The MAC sublayer performs the addressing and recognition of frames in support of LLC. MAC also performs other functions, such as frame check sequence generation and checking, and LLC protocol data unit (PDU) delimiting.

The exact format of the MAC frame differs for the various MAC protocols in use. In general all the MAC frames have a format similar to the one of the figure 2.3. The fields of this frame are as follows:

- **MAC control:** this field contains any protocol control information needed for the functioning of the MAC protocol (e.g. priority level).
- **Destination MAC address:** the destination physical attachment point on the LAN for this frame.
- **Source MAC address:** the source physical attachment point on the LAN for this frame.
- **Data:** the body of the MAC frame. This may be LLC data from the next higher layer or control information for the MAC protocol.
- **CRC:** the cyclic redundancy check field is an error-detecting code.

The MAC sublayer is also responsible for detecting errors and discarding any frames that are in error.

2.2 IEEE 802.11

2.2.1 Protocol Architecture

802.11 was the first IEEE standard adopted (first release in 1997, current version in 1999) for *wireless LANs* (**WLAN**). Figure 2.4 illustrates the model developed by the 802.11 working group.

The smallest building block of a wireless LAN is a *basic service set* (**BSS**), which consists of some number of stations executing the same MAC protocol and competing for access to the same shared wireless medium. A BSS may be isolated or it may connect to a backbone *distribution system* (**DS**) through an *access point* (**AP**). The access point functions as a bridge. The

MAC protocol may be fully distributed or controlled by a central coordinator function housed in the access point. The DS can be a switch, a wired network or a wireless network.

The simplest configuration is the one in which each station belongs to a single BSS; in this case, each station is within wireless range only of other stations within the same BSS. There's the opportunity to overlap geographically two BSS, so that a single station could participate in more than one BSS. The association between a station and a BSS is dynamic, so the station is allowed to turn off, come with range and go out of range.

An *extended service set (ESS)* consists of two or more BSS interconnected by a distribution system (an example is shown in figure 2.5). Consequently the extended service set appears as a single logical LAN to the logical link control (LLC) level. A portal is used when it's needed to integrate the IEEE 802.11 architecture with a traditional wired LAN. This portal is usually implemented in a device such a bridge or a router belonging to the wired LAN and attached to the DS.

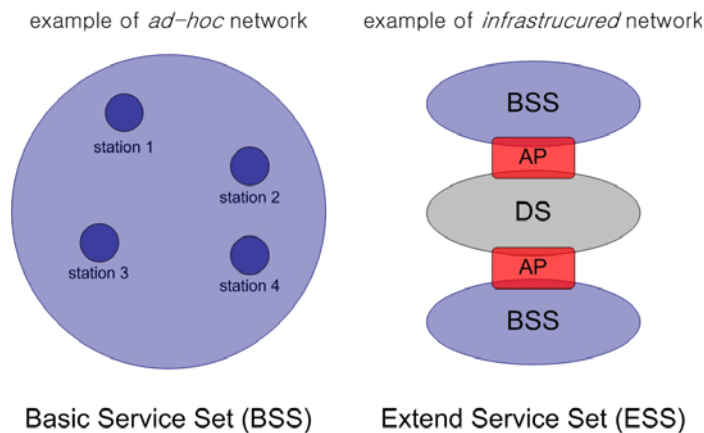


Figure 2.4: IEEE 802 Architecture components

2.2.2 Services

IEEE 802.11 defines nine services that need to be provided by the wireless LAN to provide functionality equivalent to that which is inherent to wired LANs. The services provider can be either the station or the distribution system.

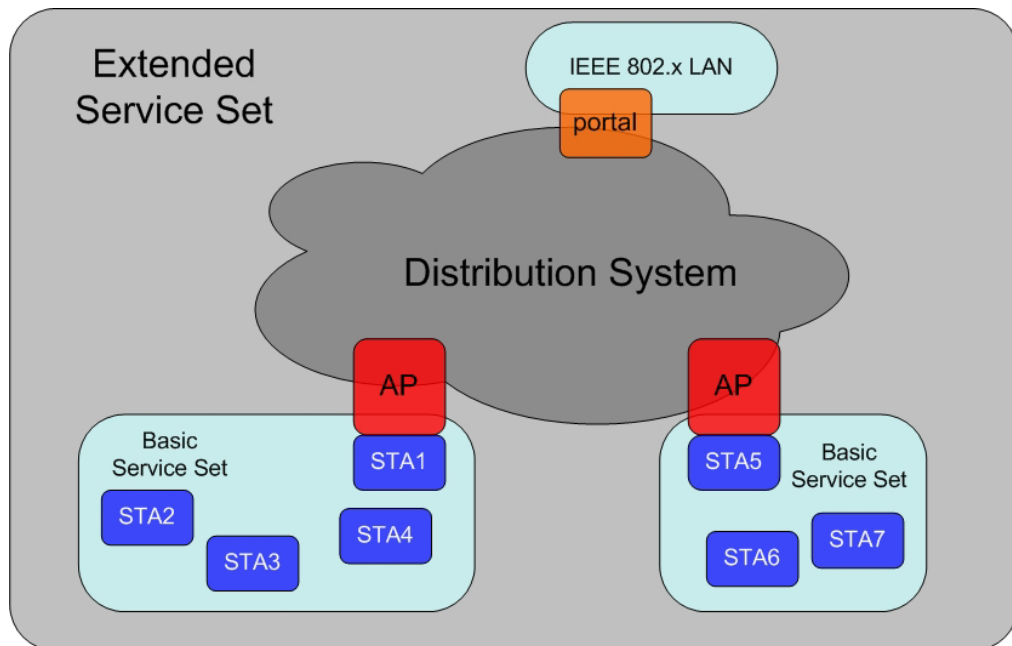


Figure 2.5: ESS structure example

Station Services

The 802.11 standard defines services for providing functions among stations. Station services are implemented within all stations on an 802.11 WLAN (including access points). The main thrust behind station services is to provide security and data delivery services for the WLAN. The four station services and functions detailed below include: authentication, de-authentication, privacy and data delivery.

1) Authentication

Because wireless LANs have limited physical security to prevent unauthorized access, 802.11 defines authentication services to control access to the WLAN. The goal of authentication service is to provide access control equal to a wired LAN. The authentication service provides a mechanism for one station to identify another station. Without this proof of identity, the station is not allowed to use the WLAN for data delivery. All 802.11 stations, whether they are part of an independent BSS or ESS network, must use the authentication service prior to communicating with another station. IEEE 802.11 defines two types of authentication services: **Open System Authentication** and **Shared Key Authentication**.

Open System Authentication is the default authentication method, which is a very simple, two-step process. First the station wanting to authenticate with another station sends an authentication management frame containing the sending station's identity. The receiving station then sends back a frame alerting whether it recognizes the identity of the authenticating station.

Shared Key Authentication assumes that each station has received a secret shared key through a secure channel independent of the 802.11 network. Stations authenticate through shared knowledge of the secret key. Use of shared key authentication requires implementation of encryption via the *Wired Equivalent Privacy* (**WEP**) algorithm.

2) De-Authentication

The de-authentication service is used to eliminate a previously authorized user from any further use of the network. Once a station is de-authenticated, that station is no longer able to access the WLAN without performing the authentication function again. De-authentication is a notification and cannot be refused. For example, when a station wishes to be removed from a BSS, it can send a de-authentication management frame to the associated access point to notify the access point of the removal from the network. An access point could also de-authenticate a station by sending a de-authentication frame to the station.

3) Privacy

The privacy service of IEEE 802.11 is designed to provide an equivalent level of protection for data on the WLAN as that provided by a wired network with restricted physical access. This service protects that data only as it traverses the wireless medium. It is not designed to provide complete protection of data between applications running over a mixed network.

With a wireless network, all stations and other devices can "hear" data traffic taking place within range on the network, seriously impacting the security level of a wireless link. IEEE 802.11 counters this problem by offering a privacy service option that raises the security of the 802.11 network to that of a wired network. The privacy service, applying to all data frames and some authentication management frames, is an encryption algorithm based on the 802.11 WEP algorithm.

4) Data Delivery

Data delivery service is similar to that provided by all other IEEE 802 LANs. The data delivery service provides reliable delivery of data frames from the MAC in one station to the MAC in one or more other stations, with minimal duplication and reordering of frames.

Distribution Services

Distribution services provide functionality across a distribution system. Typically, access points provide distribution services. The five distribution services and functions detailed below include: association, disassociation, re-association, distribution, and integration.

1) Association

The association service is used to make a logical connection between a mobile station and an access point. Each station must become associated with an access point before it is allowed to send data through the access point onto the distribution system. The connection is necessary in order for the distribution system to know where and how to deliver data to the mobile station. The mobile station invokes the association service once and only once, typically when the station enters the BSS. Each station can associate with one access point though an access point can associate with multiple stations.

2) Disassociation

The disassociation service is used either to force a mobile station to eliminate an association with an access point or for a mobile station to inform an access point that it no longer requires the services of the distribution system. When a station becomes disassociated, it must begin a new association to communicate with an access point again.

An access point may force a station or stations to disassociate because of resource restraints, the access point is shutting down or being removed from the network for a variety of reasons. When a mobile station is aware that it will no longer require the services of an access point, it may invoke the disassociation service to notify the access point that the logical connection to the services of the access point from this mobile station is no longer required. Stations should disassociate when they leave a network, though there is nothing in the architecture to assure this happens. Disassociation is a notification and can be invoked by either associated party. Neither party can refuse termination of the association.

3) Re-Association

Re-association enables a station to change its current association with an access point. The re-association service is similar to the association service, with the exception that it includes information about the access point with which a mobile station has been previously associated. A mobile station will use the re-association service repeatedly as it moves through out the ESS, loses contact with the access point with which it is associated, and needs

to become associated with a new access point. By using the re-association service, a mobile station provides information to the access point to which it will be associated and information pertaining to the access point which it will be disassociated. This allows the newly associated access point to contact the previously associated access point to obtain frames that may be waiting there for delivery to the mobile station as well as other information that may be relevant to the new association. The mobile station always initiates re-association.

4) Distribution

Distribution is the primary service used by an 802.11 station. A station uses the distribution service every time it sends MAC frames across the distribution system. The distribution service provides the distribution with only enough information to determine the proper destination BSS for the MAC frame. The three association services (association, re-association, and disassociation) provide the necessary information for the distribution service to operate. Distribution within the distribution system does not necessarily involve any additional features outside of the association services, though a station must be associated with an access point for the distribution service to forward frames properly.

5) Integration

The integration service connects the 802.11 WLAN to other LANs, including one or more wired LANs or 802.11 WLANs. A portal performs the integration service as shown in figure 2.5 on page 13. The portal is an abstract architectural concept that typically resides in an access point though it could be part of a separate network component entirely. The integration service translates 802.11 frames to frames that may traverse another network, and vice versa as well as translates frames from other networks to frames that may be delivered by an 802.11 WLAN.

Authentication & Association

As shown before, the need of a client to be mobile brought in the separation of authentication and association processes. Since a client frequently changes AP boundaries, it can be authenticated to various AP at a given point, yet remains associated to its chosen one. Before a client gets associated to other, it must be first authenticated.

This procedure is shown in figure 2.6.

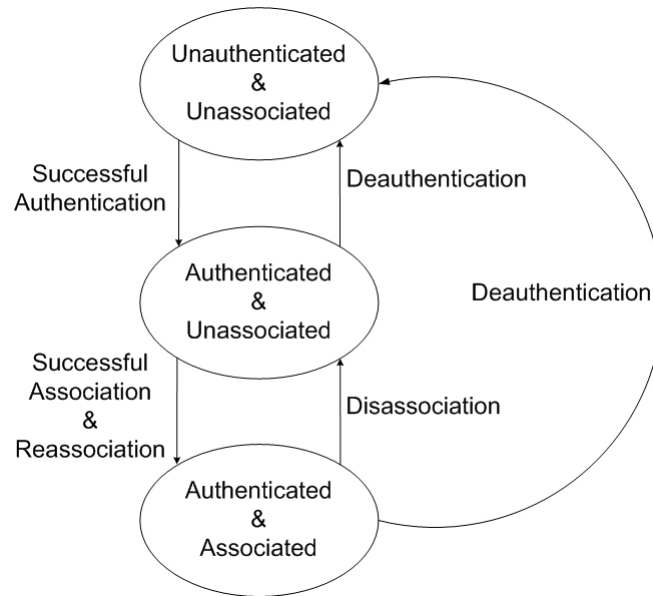


Figure 2.6: Authentication & Association procedure

2.2.3 Medium Access Control - MAC

Reliable Data Delivery

The 802.11 MAC layer provides functionality to allow reliable data delivery for the upper layers over the wireless PHY media. The data delivery itself is based on an asynchronous, best-effort, connectionless delivery of MAC layer data. There is no guarantee that the frames will be delivered successfully. For this reason, IEEE 802.11 includes a frame exchange protocol. When a station receives a data frame from another station it returns an *acknowledgment frame* (**ACK**) to the source station. This exchange can not to be interrupted by a transmission from any other station. If the source does not receive an ACK within a short period of time, it retransmits the frame.

To further enhance reliability, a four frame exchange may be used. A source first issues a *Request To Send* (**RTS**) frame to the destination; the destination then responds with a *Clear To Send* (**CTS**). After receiving the CTS, the source transmits the data frame and wait for the corresponding ACK from the destination. The RTS and CTS alert all stations that are within reception range of the source that an exchange is under way; these

stations renounce to transmit in order to prevent collisions. The RTS/CTS portion of the exchange is a required function of the MAC, but may be disabled.

Access Control

The 802.11 MAC algorithm is called *Distributed Foundation Wireless MAC (DFWMAC)* and provides a distributed access control mechanism with an optional centralized control built on top of that. Figure 2.7 illustrates this architecture.

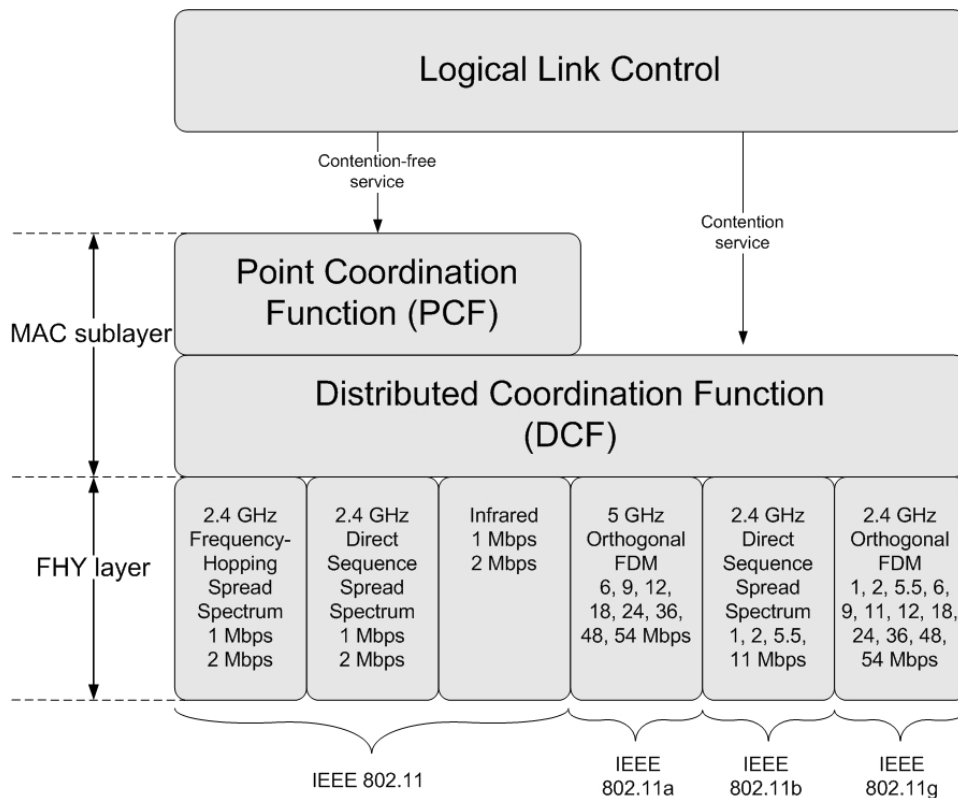


Figure 2.7: IEEE 802.11 Protocol architecture

The lower sublayer of the MAC is the *Distributed Coordinator Function (DCF)*. DCF uses a contention algorithm to provide access to all traffic. Ordinary asynchronous traffic directly uses DCF.

The *Point Coordinator Function (PCF)* is a centralized MAC algorithm used to provide contention-free service. PCF is built on top of DCF and

exploits features of DCF to assure access for its users.

Distributed Coordinator Function (DCF)

The DCF sublayer makes use of a simple CSMA algorithm. DCF defines a basic two-way handshaking access mechanism and an optional four-way handshaking mechanism.

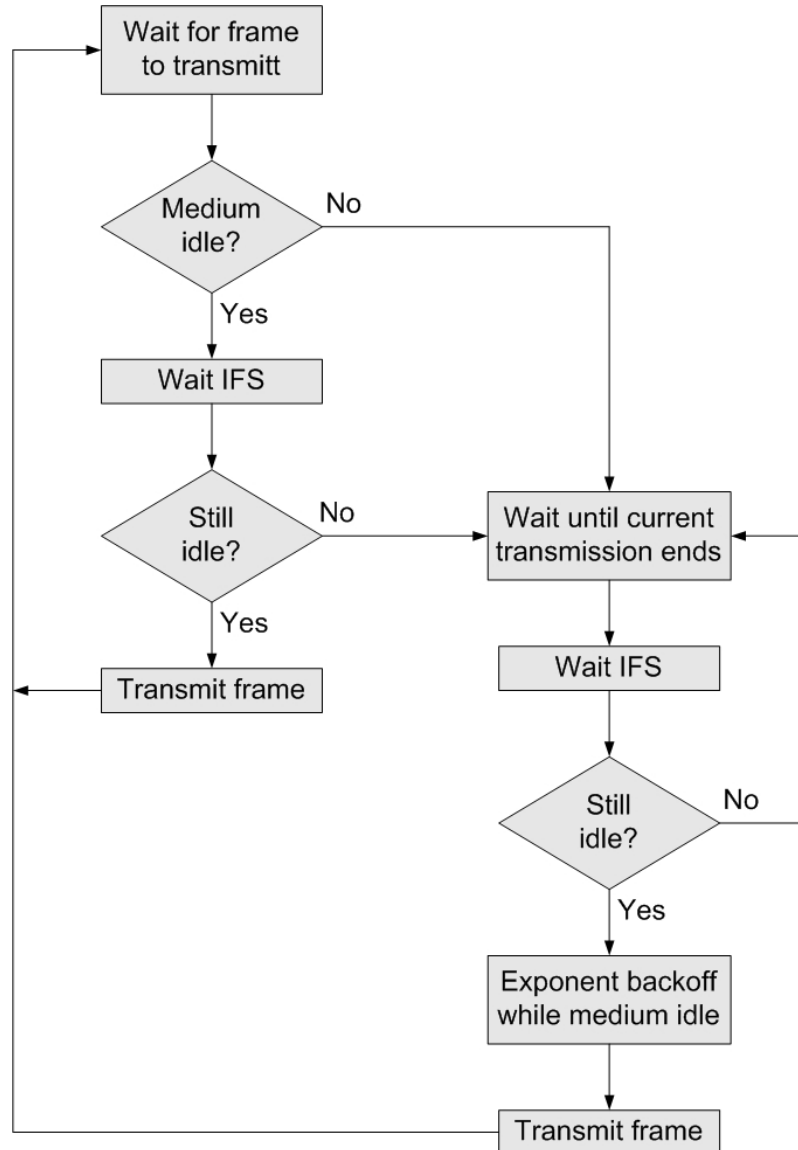


Figure 2.8: Basic access mechanism logic used in DCF

A station with a new packet to transmit monitors the channel activity.

If the channel is idle for a period of time equal to a *Distributed Inter-Frame Space* (**DIFS**), the station transmits. Otherwise, if the channel is sensed busy (either immediately or during the DIFS), the station persists to monitor the channel until it is measured idle for a DIFS. At this point, the station generates a random backoff interval before transmitting (this is the "Collision Avoidance" feature of the protocol) to minimize the probability of collision with packets being transmitted by other stations. In addition, to avoid channel capture, a station must wait a random backoff time between two consecutive new packet transmissions, even if the medium is sensed idle in the DIFS time. The logic flow chart of this mechanism is shown in figure 2.8.

For efficiency reasons, DCF employs a discrete-time backoff scale. The time immediately following an idle DIFS is slotted, and a station is allowed to transmit only at the beginning of each slot time. The slot time size is set equal to the time needed at any station to detect the transmission of a packet from any other station. The value depends on the physical layer (specified in IEEE802.11 specification), and it accounts for the propagation delay, for the time needed to switch from the receiving to the transmitting state (**RX_TX_Turnaround_Time**), and for the time to signal to the MAC layer the state of the channel (busy detect time).

DCF adopts an exponential backoff scheme. At each packet transmission, the backoff time is uniformly chosen in the range $(0, w)$. The value w is called *Contention Window*, and depends on the number of transmissions failed for the packet. At the first transmission attempt, is set equal to a value CW_{min} called minimum contention window. After each unsuccessful transmission, w is doubled, up to a maximum value CW_{max} . The backoff time counter is decremented as long as the channel is sensed idle, "frozen" when a transmission is detected on the channel, and reactivated when the channel is sensed idle again for more than a DIFS. The station transmits when the backoff time reaches zero.

Since the CSMA/CA does not rely on the capability of the stations to detect a collision by hearing their own transmission, an ACK is transmitted by the destination station to signal the successful packet reception. The ACK is immediately transmitted at the end of the packet, after a period of time called *Short Inter-Frame Space* (**SIFS**). As the SIFS (plus the propagation delay) is shorter than a DIFS, no other station is able to detect the channel idle for a DIFS until the end of the ACK. If the transmitting station does not receive the ACK within a specified **ACK_Timeout**, or it detects the transmission of a different packet on the channel, it reschedules the packet transmission according to the given backoff rules.

The above described two-way handshaking technique for the packet trans-

mission is called **basic access mechanism** and is shown in figure 2.9.

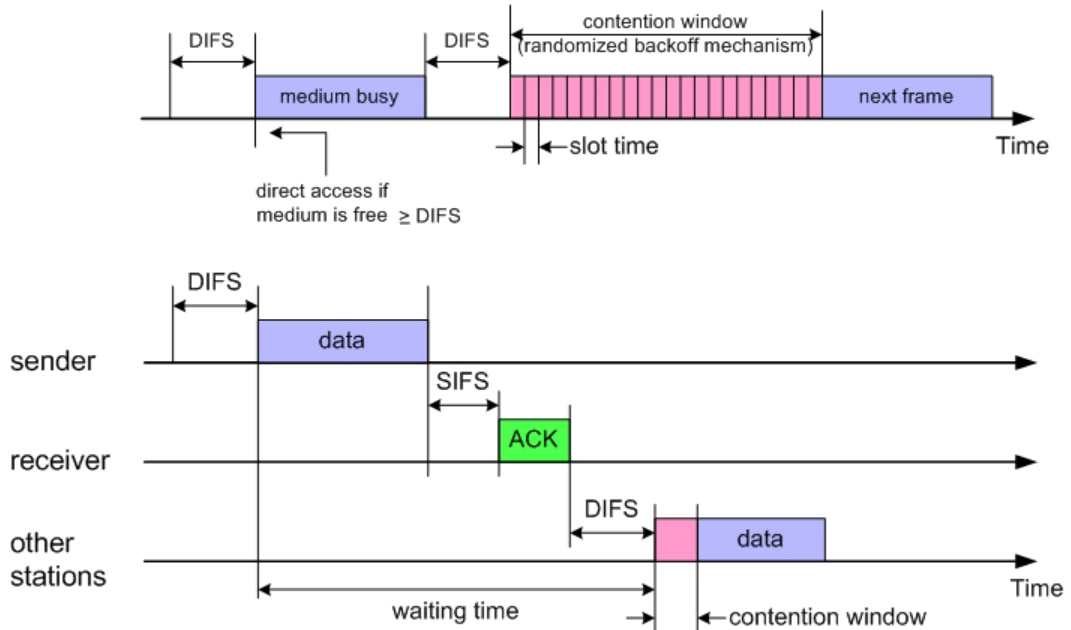


Figure 2.9: Basic access mechanism used in DCF

DCF defines an additional four-way handshaking technique to be optionally used for a packet transmission. This mechanism is also known with the name **RTS/CTS**. This optional mechanism is very useful for solving hidden/exposed terminal problems that are common in ad hoc networks.

A station that wants to transmit a packet, waits until the channel is sensed idle for a DIFS, follows the backoff rules explained above, and then, instead of the packet, preliminarily transmits a special short frame called request to send (RTS). When the receiving station detects an RTS frame, it responds, after a SIFS, with a clear to send (CTS) frame. The transmitting station is allowed to transmit its packet only if the CTS frame is correctly received.

The frames RTS and CTS carry the information of the length of the packet to be transmitted. This information can be read by any listening station, which is then able to update a *Network Allocation Vector (NAV)* containing the information of the period of time in which the channel will remain busy. Therefore, when a station is hidden from either the transmitting or the receiving station, by detecting just one frame among the RTS and CTS frames, it can suitably delay further transmission, and thus avoid collision.

Point Coordination Function (PCF)

PCF is an alternative access method; it uses a *Point Coordinator (PC)* and it's built on top of the DCF. It is a polling-based method.

The point coordinator makes use of *Point coordinator function IFS (PIFS)* when issuing polls. Because PIFS is smaller than DIFS, the point coordinator can seize the medium and lock out all asynchronous traffic while it issues polls and receives responses. If the discipline of the preceding paragraph were implemented, the point coordinator would lock out all asynchronous traffic by repeatedly issuing polls. To prevent this, an interval known as the superframe is defined. During the first part of this interval, the point coordinator issues polls in a round robin fashion to all stations configured for polling. Then it idles for the remainder of the superframe, allowing a contention period for asynchronous access. At the end of the superframe interval, the point coordinator contends for accessing the medium using PIFS. If the medium is idle, the point coordinator gains immediate access and a full superframe period follows. Otherways the point coordinator must wait until the medium is idle to gain access.

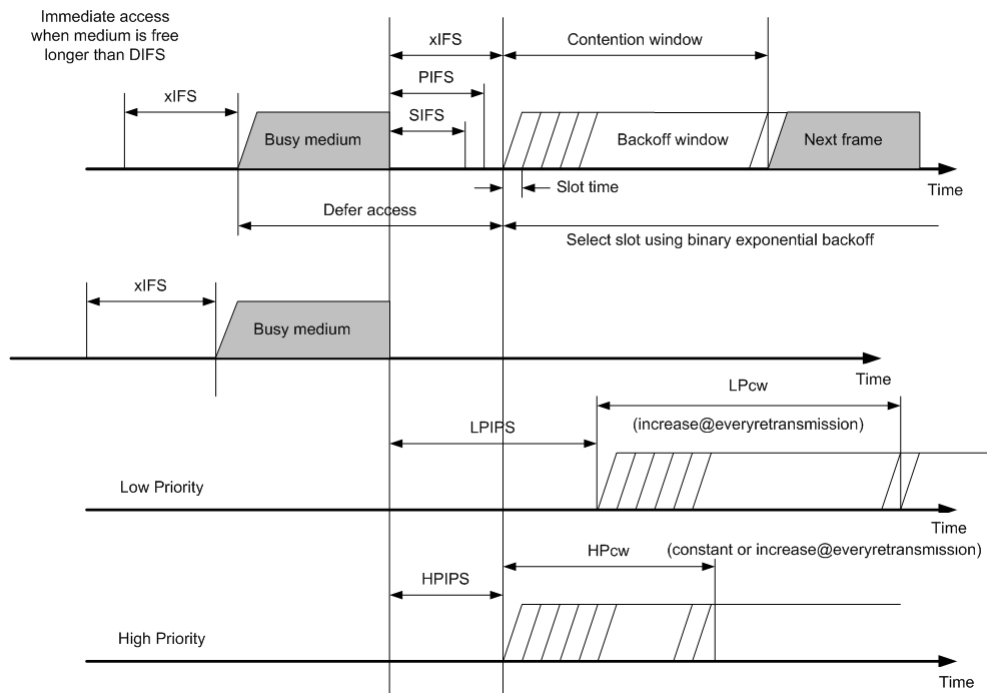


Figure 2.10: MAC timing

MAC frame

Figure 2.11 shows the 802.11 frame format. This is the format used for all data and control frames but not all fields are used in all contexts.

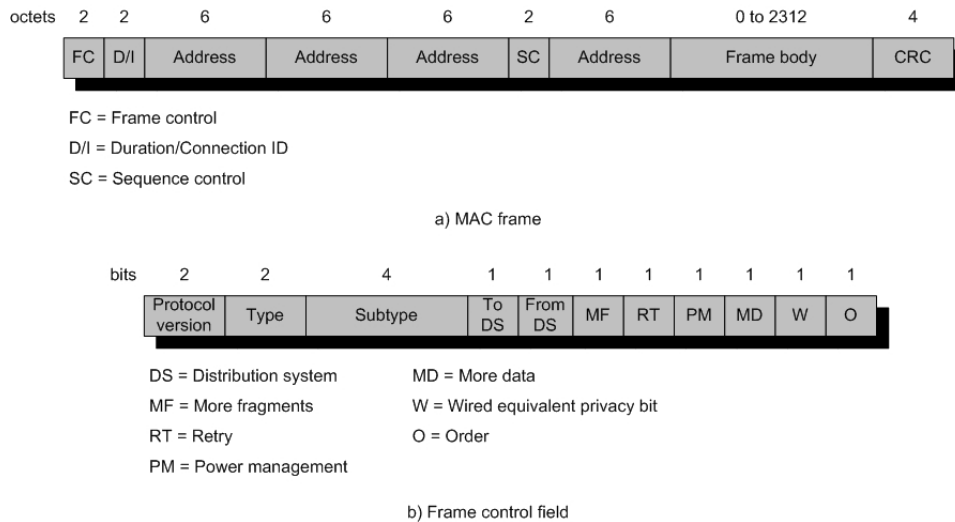


Figure 2.11: MAC frame

- **Frame control:** indicates the type of frame and provides control information.
- **Duration/connection ID:** if used as a duration field, indicates the time the channel will be allocated for successful transmission of a MAC frame. In some control frames, this field contains an association or connection identifier.
- **Addresses:** the number and meaning of the address fields depend on the context. Address types include source, destination, transmitting station and receiving station.
- **Sequence control:** contains 4-bit fragment number subfield used for fragmentation and reassembly and a 12-bit sequence number used to number frames sent between a given transmitter and receiver.
- **Frame body:** contains an MSDU or a fragment of an MSDU. The MSDU is a LLC protocol data unit or MAC control information.
- **Frame check sequence:** a 32-bit cyclic redundancy check.

The frame control field is built with the next fields.

- **Protocol version:** 802.11 version.
- **Type:** identifies the frame as control, management or data.
- **Subtype:** identifies the function of the frame.
- **To DS:** the MAC coordinator sets this bit to 1 in a frame destined to the distribution system.
- **From DS:** the MAC coordinator sets this bit to 1 in a frame leaving the distribution system.
- **More fragments:** set to 1 if more fragments follow this one.
- **Retry:** set to 1 if this is a retransmission of a previous frame
- **Power management:** set to 1 if the transmitting station is in a sleep mode.
- **More data:** indicates that a station has additional data to send.
- **WEP:** set to 1 if the optional wired equivalent protocol is implemented.
- **Order:** set to 1 in any data frame sent using Strictly Ordered service, which tells the receiver that the frames must be processed in order.

MAC frames can be divided in three groups: control frames, management frames and data frames.

1) Control Frames

Control frames assist in the reliable delivery of data frames.

- **Power save-poll (PS-Poll):** this frame is sent by any station to the station that includes the AP. Its goal is to request that the AP transmit a frame that has been buffered for this station while the station was in power-saving mode.
- **Request To Send (RTS):** the station sending this message alerts all other stations that it intends sending a data frame to a certain destination.
- **Clear To Send (CTS):** it is sent by the destination station to the source station to grant permission to send a data frame.

- **Acknowledgement:** provides an acknowledgement from the destination to the source.
- **Contention-free (CF)-end:** announces the end of a contention free period.
- **CF-end + CF-ack:** this frame ends the contention-free period and releases stations from the restrictions associated with that period.

2) Data Frames

- **Data:** the simplest data frame.
- **Data + CF-Ack:** may only be sent during a contention-free period. In addition to carry data, this frame acknowledges previously received data.
- **Data + CF-Poll:** used by a point coordinator to deliver data to a station and also to request that the station send a data frame that it may have buffered.
- **Data + CF-Ack + CF-Poll:** combines the functions of the two previous frames.
- **Null Function:** it carries no data, no acknowledgements and no polls. It is used only to indicate that a station is changing to a low-power operating state.
- **CF-Ack:** it has the same functionality as the corresponding data frame but without the data.
- **CF-Poll:** it has the same functionality as the corresponding data frame but without the data.
- **CF-Ack + CF-Poll:** it has the same functionality as the corresponding data frame but without the data.

3) Management Frames

Management Frames are used to manage communications between stations and AP.

- **Association request:** sent by a station to an AP to request an association with this BSS; in it its included information about whether encryption is to be used and whether this station is pollable.

- **Association response:** returned by the AP to the station to indicate whether it is accepting this association request.
- **Reassociation request:** sent by a station when it moves from one BSS to another and needs to make an association with the AP in the new BSS.
- **Reassociation response:** returned by the AP to the station to indicate whether it is accepting this reassociation request.
- **Probe request:** used by a station to obtain information from another station or AP; it is usually used to locate an IEEE 802.11 BSS.
- **Probe response:** response to a probe request.
- **Beacon:** transmitted periodically to allow mobile stations to locate and identify a BSS.
- **Announcement traffic indication message:** sent by a station to alert other stations that may have been in low power mode that this station has frame buffered and waiting to be delivered.
- **Disassociation:** used by a station to terminate an association.
- **Authentication:** multiple authentication frames are used in an exchange to authenticate one station to another.
- **Deauthentication:** sent by a station to another station or AP to indicate that is terminating secure communications.

Security: the WEP Protocol

Because wireless is a shared medium, everything that is transmitted or received over a wireless network can be intercepted. Encryption and authentication are always considered when developing a wireless networking system. The goal of adding these security features is to make wireless traffic as secure as wired traffic.

The IEEE 802.11 standard provides a mechanism to do this by encrypting the traffic and authenticating nodes via the *Wired Equivalent Privacy* (**WEP**) protocol. To provide privacy, as well as data integrity, WEP uses an encryption algorithm based on the **RC4** encryption algorithm. The WEP algorithm is used to protect wireless communication from eavesdropping. A secondary function of WEP is to prevent unauthorized access to a wireless

network; this function is not an explicit goal in the 802.11 standard, but it is frequently considered to be a feature of WEP.

WEP relies on a secret key that is shared between a mobile station (e.g. a laptop with a wireless ethernet card) and an access point (e.g. a base station). The secret key is used to encrypt packets before they are transmitted, and an integrity check is used to ensure that packets are not modified in transit. The standard does not discuss how the shared key is established. In practice, most installations use a single key that is shared between all mobile stations and access points. More sophisticated key management techniques can be used to help defend from the attacks.

Encryption of a frame

Two processes are applied to the plaintext data. One encrypts the plaintext; the other protects against unauthorized data modification.

The **secret key** (40-bits) is concatenated with an *initialization vector* (**IV**, 24-bits) resulting in a 64-bit total key size. The resulting key is input into the *PseudoRandom Number Generator* (**PRNG**). The PRNG, using RC4, outputs a pseudorandom key sequence based on the input key. The resulting sequence is used to encrypt the data by doing a bitwise XOR. This results in encrypted bytes equal in length to the number of data bytes that are to be transmitted in the expanded data plus 4 bytes. This is because the key sequence is used to protect the *Integrity Check Value* (**ICV**, 32-bits) as well as the data. To protect against unauthorized data modification, an integrity algorithm (CRC-32) operates on the plaintext to produce the ICV.

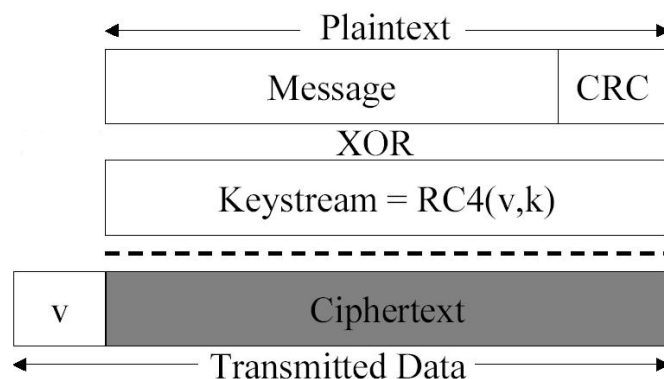


Figure 2.12: WEP encryption

The ciphertext is accomplished by doing the following steps as shown in figure 2.13.

1. Compute the ICV using CRC-32 over the message plaintext.
2. Concatenate the ICV to the plaintext.
3. Choose a random initialization vector (IV) and concatenate this to the secret key.
4. Input the secret key+IV into the RC4 algorithm to produce a pseudo-random key sequence.
5. Encrypt the plaintext+ICV by doing a bitwise XOR with the pseudo-random key sequence under RC4 to produce the ciphertext.
6. Communicate the IV to the peer by placing it in front of the ciphertext. The IV, plaintext, and ICV triplet forms the actual data sent in the data frame.

Decryption of a frame

In decryption, the IV of the incoming message is used to generate the key sequence necessary to decrypt the incoming message (see figure 2.13). Combining the ciphertext with the proper key sequence yields the original plaintext and ICV.

The decryption is verified by performing the integrity check algorithm on the recovered plaintext and comparing the output ICV' to the ICV transmitted with the message. If ICV' is not equal to ICV, the received message is in error, and an error indication is sent to the MAC management and back to the sending station.

Mobile units with erroneous messages (due to inability to decrypt) are not authenticated.

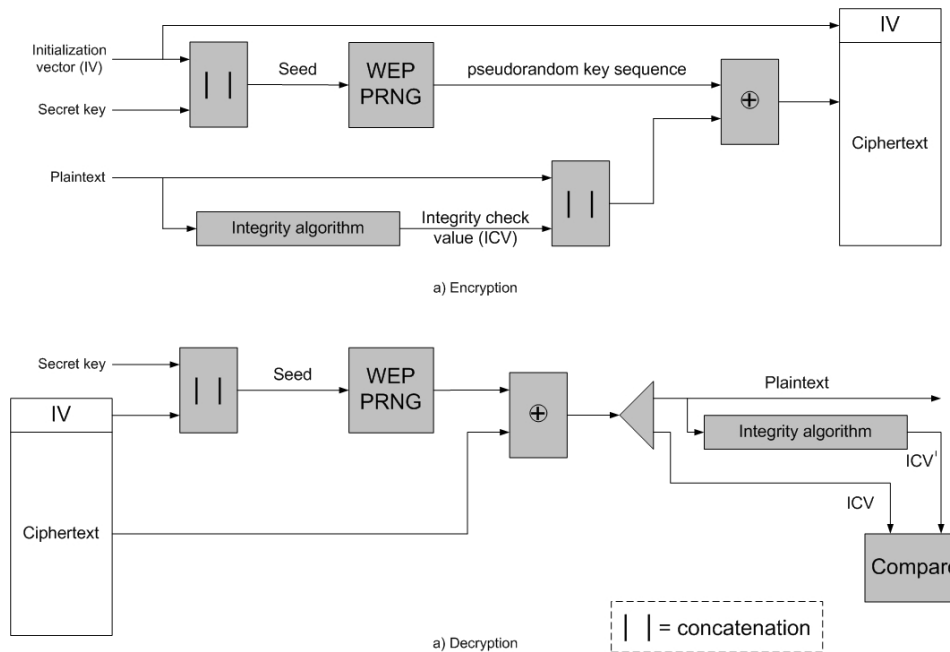


Figure 2.13: Block diagrams for WEP encryption and decryption

Authentication

The same shared key used to encrypt/decrypt the data frames is also used to authenticate the station. It is considered a security risk to have the encryption and authentication keys be the same. IEEE 802.11 provides two types of authentication: Open System Authentication and Shared Key System.

Open system authentication simply provides a way for two parties to agree to exchange data and provides no security benefits. In open system authentication, one party sends a MAC control frame, known as an authentication frame, to the other party. The frame indicates that this is an open system authentication type. The other party responds with its own authentication frame and the process is complete. Thus, open system authentication consists simply of the exchange of identities between the parties and provides "null" authentication. The station can associate with any access point and listen to all data that are sent plaintext.

Shared key authentication requires that two parties share a secret key not shared by any other party. This key is used to assure that both sides are authenticated to each other. Figure 2.14 illustrates the operation of shared-key authentication. The secret shared key resides in each station's MIB in a write-only form and is therefore only available to the MAC coordinator. The 802.11 standard does not specify how to distribute the keys to each station,

however.

The process is as follows:

1. A requesting station sends an Authentication Request frame to the access point.
2. When the AP receives an initial Authentication Request frame, it will reply with an Authentication frame containing 128 bytes of random "challenge text" generated by the WEP engine in standard form.
3. The requesting station will then copy the challenge text into an Authentication frame, encrypt it with a shared key, and then send the Challenge Response frame to the responding station.
4. The receiving AP will decrypt the value of the challenge text using the same shared key and compare it to the challenge text sent earlier. If a match occurs, the AP will reply with a frame indicating a successful authentication. If not, the responding AP will send a negative authentication.

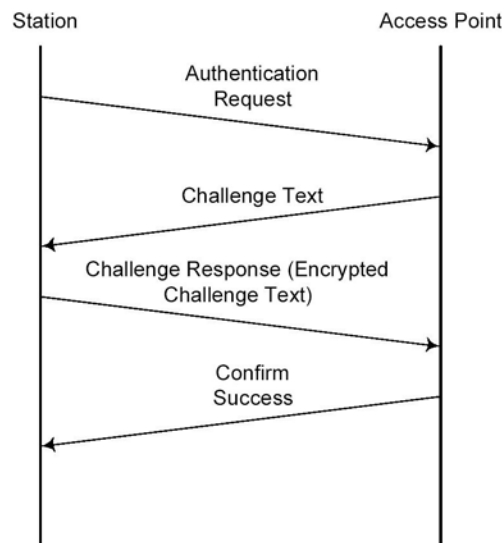


Figure 2.14: Shared key authentication

2.2.4 Physical Layer

The physical layer for IEEE 802.11 has been developed in three stages; the first part was issued in 1997 and the remaining two parts in 1999.

The first part, called *IEEE 802.11*, includes the MAC layer and three physical layer specifications:

- Direct-sequence spread spectrum operating in the 2.4 GHz ISM band, at data rates of 1 Mbps and 2 Mbps.
- Frequency-hopping spread spectrum operating in the 2.4 GHz ISM band, at data rates of 1 Mbps and 2 Mbps.
- Infrared at 1Mbps and 2 Mbps operating at a wavelength between 850 and 950 nm.

The other two layers developed in 1999 are *IEEE 802.11a* that operates in the 5 GHz band at data rates up to 54 Mbps and *IEEE 802.11b* that operates in the 2.4 GHz band at 5.5 and 11 Mbps. Further standards were developed in the following years but in our discussion we will focus only on the IEEE 802.11b physical layer.

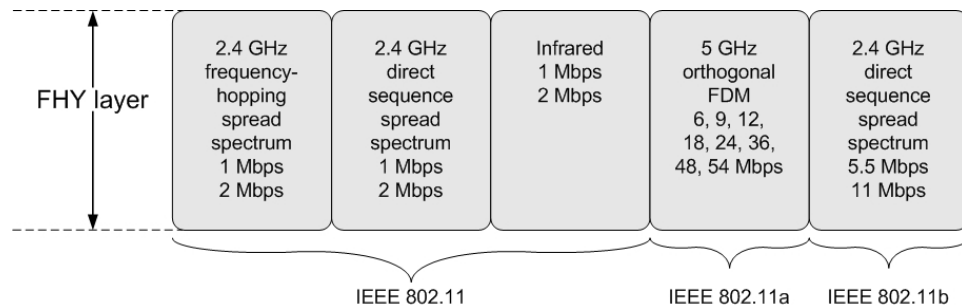


Figure 2.15: IEEE 802.11 PHY layer

2.3 IEEE 802.11b

2.3.1 Introduction

IEEE 802.11 specifies a 2.4 GHz operating frequency with data rates of 1 and 2 Mbps using either *Direct Sequence Spread Spectrum (DSSS)* or *Frequency Hopping Spread Spectrum (FHSS)*.

In IEEE 802.11b data is encoded using DSSS technology. DSSS works by taking a data stream of zeros and ones and modulating it with a second pattern, the chipping sequence. In 802.11, that sequence is known as the Barker code, which is an 11 bits sequence (10110111000) that has certain mathematical properties making it ideal for modulating radio waves.

A Barker sequence is a binary $\{-1,+1\}$ sequence $s(t)$ of length n with the property that its autocorrelation values $R(\tau)$ satisfy $|R(\tau)| \leq 1$ for all $|\tau| \leq (n - 1)$. Further, the Barker property is preserved under the following transformations.

$$s(t) \rightarrow -s(t) \quad s(t) \rightarrow (-1)^t s(t) \quad \text{and} \quad s(t) \rightarrow -s(n - 1 - t)$$

as well as under compositions of these transformations.

Only the following Barker sequence are known:

n =	2	++
n =	3	++-
n =	4	+++-
n =	5	++++-
n =	7	++++--+-
n =	11	+ - + + - + + + - - -
n =	13	+ + + + + - - + + - + - +

The 11-chip Barker sequence is used. Thus, each data binary 1 is mapped into the sequence $\{+ - + + - + + + - - -\}$, and each 0 is mapped into the sequence $\{- + - - + - - - + + +\}$.

Important characteristic of Barker sequences are their robustness against interference and their insensitivity to multipath propagation.

The basic data stream is XOR'd with the Barker code to generate a series of data objects called chips. Each bit is "encoded" by the 11bits Barker code, and each group of 11 chips encodes one bit of data.

IEEE 802.11b uses 64 **CCK** (*Complementary Code Keying*) chipping sequences to achieve 11 Mbps. Rather than using the Barker code, CCK uses a series of codes called Complementary Sequences. Because there are 64 unique code words that can be used to encode the signal, up to 6 bits can be represented by any one particular code word (instead of the 1 bit represented by a Barker symbol).

The wireless radio generates a 2.4 GHz carrier wave (2.4 to 2.483 GHz) and modulates that wave using a variety of techniques. For 1 Mbps transmission, **BPSK** (*Binary Phase Shift Keying*) is used (one phase shift for each bit). To accomplish 2 Mbps transmission, **QPSK** (*Quadrature Phase*

Shift Keying) is used. QPSK uses four rotations (0, 90, 180 and 270 degrees) to encode 2 bits of information in the same space as BPSK encodes 1. The trade-off is increase power or decrease range to maintain signal quality. Because the FCC regulates output power of portable radios to 1 watt **EIRP** (*Equivalent Isotropic Radiated Power*), range is the only remaining factor that can change. On 802.11 devices, as the transceiver moves away from the radio, the radio adapts and uses a less complex (and slower) encoding mechanism to send data.

The MAC layer communicates with the **PLCP** (*Physical Layer Convergence Protocol*) via specific primitives through a PHY (Physical Layer) service access point. When the MAC layer instructs, the PLCP prepares **MPDUs** (*MAC Protocol Data Units*) for transmission. The PLCP also delivers incoming frames from the wireless medium to the MAC layer. The PLCP sublayer minimizes the dependence of the MAC layer on the PMD sublayer by mapping MPDUs into a frame format suitable for transmission by the **PMD** (*Physical Medium Dependent*).

Under the direction of the PLCP, the PMD provides actual transmission and reception of PHY entities between two stations through the wireless medium. To provide this service, the PMD interfaces directly with the air medium and provides modulation and demodulation of the frame transmissions. The PLCP and PMD communicate using service primitives to govern the transmission and reception functions.

The CCK code word is modulated with the QPSK technology used in 2 Mbps wireless DSSS radios. This allows for an additional 2 bits of information to be encoded in each symbol. Eight chips are sent for each 6 bits, but each symbol encodes 8 bits because of the QPSK modulation. The spectrum math for 1 Mbps transmission works out as 11 Mchips per second times 2 MHz equals 22 MHz of spectrum. Likewise, at 2 Mbps, 2 bits per symbol are modulated with QPSK, 11 Mchips per second, and thus have 22 MHz of spectrum. To send 11 Mbps 22 MHz of frequency spectrum is needed.

It is much more difficult to discern which of the 64 code words is coming across the airwaves, because of the complex encoding. Furthermore, the radio receiver design is significantly more difficult. In fact, while a 1 Mbps or 2 Mbps radio has one correlator (the device responsible for lining up the various signals bouncing around and turning them into a bit stream), the 11 Mbps radio must have 64 such devices.

Figure 2.16 shows the digital modulation of data with the PRN sequence.

The wireless physical layer is split into two parts, called the PLCP and the PMD sublayer. The PMD takes care of the wireless encoding. The PLCP presents a common interface for higher-level drivers to write to and provides

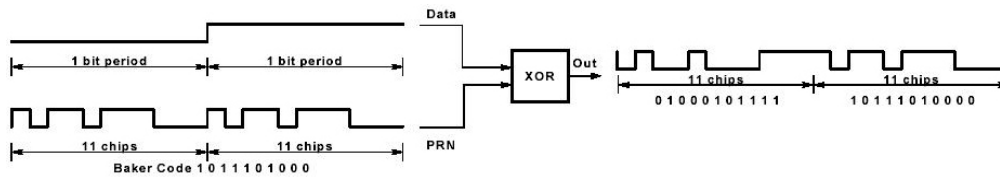


Figure 2.16: Digital modulation of data with PRN sequence

carrier sense and **CCA** (*Clear Channel Assessment*), which is the signal that the MAC layer needs so it can determine whether the medium is currently in use.

The PLCP consists of a 144 bits preamble that is used for synchronization to determine radio gain and to establish CCA. The preamble comprises 128 bits of synchronization, followed by a 16 bits field consisting of the pattern 1111001110100000. This sequence is used to mark the start of every frame and is called the SFD (Start Frame Delimiter).

The next 48 bits are collectively known as the PLCP header. The header contains four fields: signal, service, length and **HEC** (*Header Error Check*).

- The signal field indicates how fast the payload will be transmitted (1, 2, 5.5 or 11 Mbps).
- The service field is reserved for future use.
- The length field indicates the length of the ensuing payload.
- The HEC is a 16 bits CRC of the 48 bits header.

In a wireless environment, the PLCP is always transmitted at 1 Mbps. Thus, 24 bytes of each packet are sent at 1 Mbps. The PLCP introduces 24 bytes of overhead into each wireless Ethernet packet. Ethernet introduces only 8 bytes of data. Because the 192 bits header payload is transmitted at 1 Mbps, IEEE 802.11b is at best only 85 percent efficient at the physical layer.

2.3.2 Overview

The IEEE 802.11b is a Direct Sequence Spread Spectrum (DSSS) system very similar in concept to the CDMA Wireless, using a spread spectrum chip sequence.

In the IEEE 802.11b the transmission medium is wireless and the operating frequency band is 2.4 GHz. IEEE 802.11b provides 5.5 and 11 Mbps

payload data rates in addition to the 1 and 2 Mbps rates provided by 802.11. To provide the higher rates, 8 chip Complementary Code Keying (CCK) is employed as the modulation scheme. The CCK uses 6 bits to encode the code sent, this increase the speed of the 802.11 by 6. The chipping rate is 11 MHz, which is the same as the DSSS system as described in 802.11, thus providing the same occupied channel bandwidth.

IEEE 802.11b describes an optional mode replacing the CCK modulation with packet binary convolutional coding (HR/DSSS/PBCC).

Another optional mode of IEEE 802.11b allows data throughput at the higher rates (2, 5.5, and 11 Mbps) to be significantly increased by using a shorter PLCP preamble. This mode is called HR/DSSS/short. This Short Preamble mode can coexist with DSSS, HR/DSSS under limited circumstances, such as on different channels or with appropriate CCA mechanisms.

The High Rate PHY contains three functional entities: the PMD function, the physical layer convergence function, and the layer management function. For the purposes of MAC and MAC Management when channel agility is both present and enabled, the High Rate PHY shall be interpreted to be both a High Rate and a frequency hopping physical layer. The High Rate PHY service shall be provided to the MAC through the PHY service primitives.

To allow the MAC to operate with minimum dependence on the PMD sublayer, a physical layer convergence procedure (PLCP) sublayer is defined. This function simplifies the PHY service interface to the MAC services.

The PMD sublayer provides a means and method of transmitting and receiving data through a *wireless medium* (**WM**) between two or more STAs each using the High Rate system.

The PLME performs management of the local PHY functions in conjunction with the MAC management entity.

2.3.3 CCK

CCK is a variation on M-ary Orthogonal Keying modulation, which uses I/Q modulation architecture with complex symbol structures. CCK allows for multi-channel operation in the 2.4 GHz band using the existing 802.11 DSSS channel structure scheme. The spreading employs the same chipping rate and spectrum shape as the 802.11 Barker's code word. Spreading functions, allows three non-interfering channels in the 2.4 to 2.483 GHz band.

CCK is an M-ary Orthogonal Keying modulation where one of M unique (nearly orthogonal) signal codewords is chosen for transmission. The spread function for CCK is chosen from a set of M nearly orthogonal vectors by the data word. CCK uses one vector from a set of 64 complex (QPSK) vectors for the symbol and thereby modulates 6 bits (one of 64) on each 8 chips

spreading code symbol. Two more bits are sent by QPSK modulating the whole code symbol. This results in modulating 8 bits onto each symbol. The formula that defines the CCK codewords has 4 phase terms.

One of them modulates all of the chips (φ_1) and this is used for the QPSK rotation of the whole code vector.

The 3 others modulate every odd chip (φ_2), every odd pair of chips (φ_3) and every odd quad of chips (φ_4) respectively.

$$c = \left\{ \begin{array}{l} e^{j(\varphi_1+\varphi_2+\varphi_3+\varphi_4)}, e^{j(\varphi_1+\varphi_3+\varphi_4)}, e^{j(\varphi_1+\varphi_2+\varphi_4)}, e^{j(\varphi_1+\varphi_4)}, \\ e^{j(\varphi_1+\varphi_2+\varphi_3)}, e^{j(\varphi_1+\varphi_3)}, e^{j(\varphi_1+\varphi_2)}, e^{j(\varphi_1)} \end{array} \right\}$$

Walsh functions used for the M-ary Bi-Orthogonal keying (MBOK) modulation are the most well known orthogonal BPSK vector set. To transmit enough bits per symbol, the MBOK modulation is used independently on the I and Q channels of the waveform effectively doubling the data rate. CCK on the other hand uses a complex set of Walsh/Hadamard functions known as Complementary Codes.

Walsh/Hadamard properties are similar to Walsh functions but are complex, that is, more than two phase, while still being nearly orthogonal. With complex code symbols, it is not possible to independently transmit simultaneous code symbols without suffering amplitude modulation. Since the set of complementary codes is more extensive, however, we have a larger set of nearly orthogonal codes to pick from and can get the same number of bits transmitted per symbol without simultaneous transmission of symbols.

The multi-path performance of CCK is better than MBOK due to the lack of cross rail interference. For MBOK, there are 8 BPSK chips that have a maximum vector space of 256 code words of which it is possible to find sets of 8 that are orthogonal. Two independent BPSK vector sets are selected for the orthogonal I and Q channels which modulate 3 bits on each. Two additional bits are used to BPSK modulate each of the spreading code vectors. For CCK, there are 65536 possible code words, and sets of 64 that are nearly orthogonal. This is because it really takes 16 bits to define each code vector. To get a half data rate version, a subset of 4 of the 64 vectors having superior coding distance is used.

CCK suffers less from multi-path distortion in the form of cross coupling (of I and Q channel information) than MBOK. The information in CCK is encoded directly onto complex chips, which cannot be cross-couple corrupted by multi-path since each channel finger has an $Ae^{j\theta}$ distortion. A single channel path gain-scales and phase-rotates the signal. A gain scale and phase rotation of a complex chip still maintains I/Q orthogonal. This superior

encoding technique avoids the corruption resulting from encoding half the information on the I-channel and the other half on the Q-channel, as in MBOK, which easily cross-couple corrupts with the multipath's $Ae^{j\theta}$ phase rotation.

For 1 Mbps, the signal is modulated BPSK by one bit per symbol and then spread by BPSK modulating with the 11 chip Barker code at 11 Mbps. For 2 Mbps, the signal is QPSK modulated by two bits per symbol and then BPSK spread as before. For the 5.5 Mbps CCK mode, the incoming data is grouped into 4 bits nibbles where 2 of those bits select the spreading function out of the set of 4 while the remaining 2 bits QPSK modulate the symbol. The spreading sequence then DQPSK modulates the carrier by driving the I and Q modulators. To make 11 Mbps CCK modulation, the input data is grouped into 2 bits and 6 bits. The 6 bits are used to select one of 64 complex vectors of 8 chip length for the symbol and the other 2 bits DQPSK modulate the entire symbol. The chipping rate is maintained at 11 Mbps for all modes.

The signal acquisition scheme for 802.11 uses a specific preamble and header using the 1 Mbps modulation and has provision for sending the payload at different rates. The packet frame structure and protocol of 802.11 is much like 802.3 Ethernet, however it must operate wirelessly in a harsh RF environment. This means that the signal levels may become corrupted and subject to multi-path. Signal acquisition and synchronization of the preamble and header are critical. The preamble and header consists of six fields. They are: Preamble, SFD, Signal (rate), Service, Length and CRC. The header takes 48 bits, and the total length of the acquisition sequence is 192 μs . The preamble and header is modulated using the 1 Mbps modulation rate and is scrambled with a self-synchronizing scrambler. The high rate scheme will use this acquisition sequence, which already has a rate field that can be programmed for 1, 2, 5.5 or 11 Mbps.

The 802.11 packet transmission protocol is Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). This differs from "wired" Ethernet, which uses collision detection. Radios can't detect collisions, therefore they use collision avoidance using a listen before talk and random back off deferral mechanism. Since all stations use the same acquisition sequence at the lowest basic rate, all stations can see the traffic and process the signals at the appropriate rate. If legacy 1 and 2 Mbps stations receive the packet header, but are not capable of processing the higher rate, they can still defer the medium based on knowing that an 802.11 signal has been sensed and knowing the length of time it will be on the air.

To insure that the modulation has the same bandwidth as the existing 802.11 DS modulation, the chipping rate is kept at 11 Mbps while the symbol

rate is increased to 1.375 MSps. This accounts for the shorter symbols and makes the overall bit rate 11 Mbps. This approach makes system interoperability with the 802.11 preamble and header much easier. The spread rate remains constant and only the data rate changes and the spectrum of the CCK waveform is same as the legacy 802.11 waveform.

2.3.4 Walsh and Complementary Codes

Walsh codes can be obtained performing simple operations as it is illustrated in Figure 2.17. For the 2-ary case, taking a 2x2 matrix of 1s and inverting the lower right quadrant of the matrix form the basic symbols. To form the 4-ary case, take 4 of the 2x2 matrices and make a 4x4 matrix with the lower right hand quadrant again inverted. The procedure is repeated for the 8-ary case and beyond.

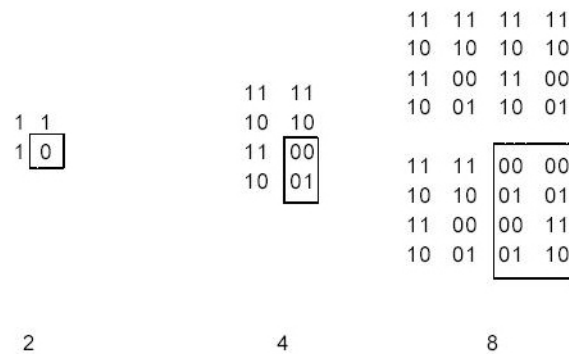


Figure 2.17: Forming Walsh Codes by successive folding

Walsh functions have a regular structure and at least one member that has a substantial DC bias. In this case it is the first row with all 1s. All the rest are half 1s and half 0s. The DC bias can be reduced on the worst member of the set by multiplying all members with a cover code. This, however, introduces a (smaller) bias in half of the members.

The main concern about MBOK is caused by the fact that it uses independent codes on the in phase and quadrature signals, which creates a significant amount of cross rail interference in the presence of multipath.

To avoid this, one would ideally transmit only symbols for which processing could be done on I and Q simultaneously, and use code words that all have good autocorrelation properties, such that there is minimal inter-symbol and

inter-chip interference. Such codes actually exist in the form of the complementary codes. For a code length of 8 chips, 256 possible sequences c can be constructed as follows, using 4 QPSK phases φ_1 to φ_4 . Note that φ_1 is presented in all 8 chips, so it simply rotates the entire code word. Hence, to decode these codes set, one would need 64 correlators plus an additional phase detection of the code that gave the largest correlation output. The correlation can be significantly simplified by using techniques like the Fast Walsh transform (analogous to an FFT butterfly circuit). In fact, when the 4 input phases φ_1 to φ_4 are binary, then the complementary code set reduces to a modified Walsh code set.

2.3.5 Fast Transform Structure

The four-phase variables each take on values of $[0, \pi/2, \pi, 3\pi/2]$, and there are 256 (4^4) possible 8 chip codes. These codes have an inherent "Walsh" type structure that allows a simple butterfly implementation of the decoder. Although it is possible to squeeze a few more complementary codes out of this 8 chips set, the rest of the codes cannot be decoded with the modified fast Walsh transform. Figure 2.18 shows the basic fast Walsh block which brings in 8 chips of soft decision data shown here by x_0, x_1, \dots, x_7 , and produces 16 possible correlation for given values of φ_1 and φ_2 . Figure 2.19 shows all 256 possible correlator outputs. The BFWB's are shown in detail in Figure 2.18. There are 28 butterflies needed for a length 8 transform. Each butterfly requires 4 additions (the phase rotations are trivial for 4-PSK), so the total number of operations is 112 complex additions. The direct calculation method with 64 separate correlators requires 512 complex additions, so the fast transform reduces the complexity by almost a factor of 5.

CCK is inherently a quadrature MOK signal. For the full data rate potential, DQPSK modulate the starting phase of the symbols to get 11 Mbps. To reduce the data rate for a more robust lower data rate, we can trim the signal set to one that has the greatest distance properties with a reduced number of vectors.

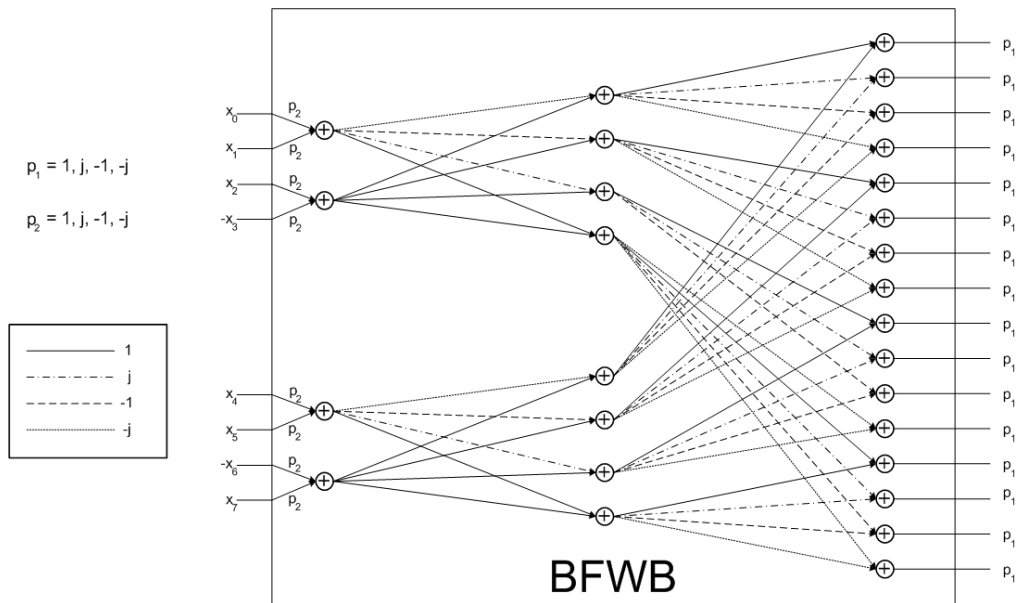


Figure 2.18: Basic Fast Walsh Transform Block (BFWB)

For 5.5 Mbps, there are two options

- First, trim the 64-ary set to 8-ary and BPSK modulate the symbols
- Second, trim the set to 4-ary and QPSK modulate the symbols.

Either scheme achieves 4 bits per symbol but simulations conclude that the latter is more robust in multipath.

The excellent range that the CCK modulation achieves is due to the fact that MOK has better E_b/N_0 performance than BPSK. This performance is due the embedded coding properties of the spreading modulation. The modulation basically ties several bits together so that the receiver makes a symbol decision. If a symbol is in error then all of the bits in that symbol are suspect, but not all will necessarily be in error. Thus, the symbol error rate and the bit error rates are related. While the SNR required making a symbol decision correctly is higher than required to make a one-bit decision, it is not as high as required to make all of the bit decisions of a symbol independently and correctly. Thus, some coding gain is embedded in the basic spreading waveform. Simulations conclude that the high rates are more susceptible to multi-path than the lower rates as would be expected from the higher required E_s/N_0 .

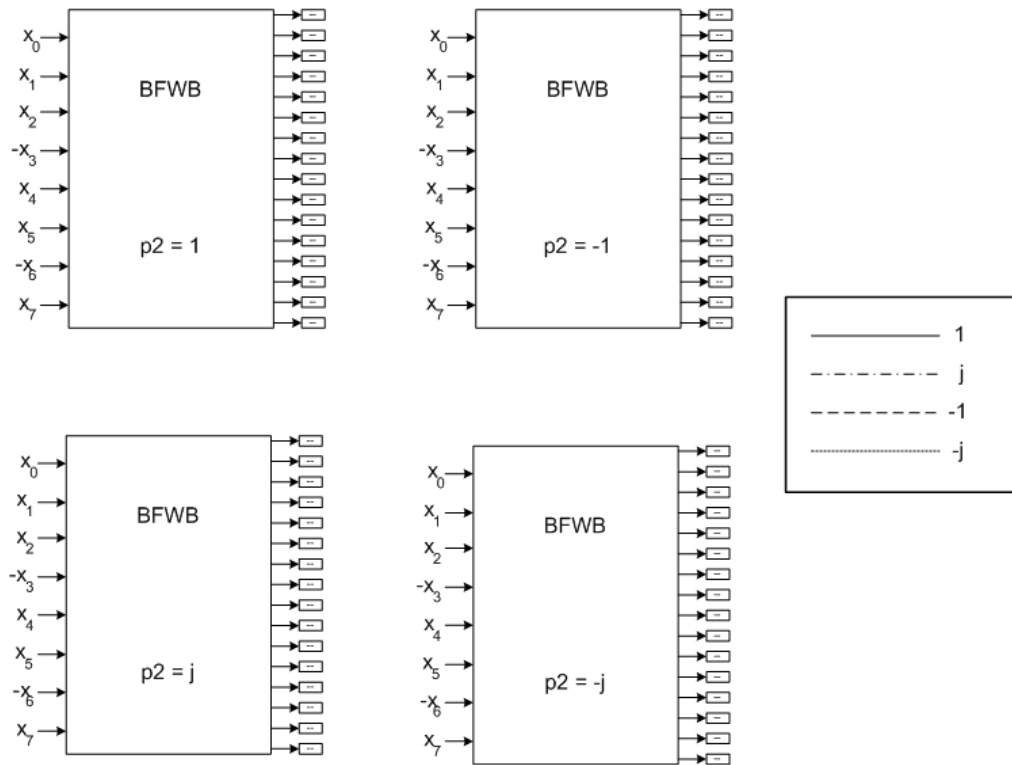


Figure 2.19: Modified Walsh Transform

Chapter 3

The WEP Vulnerabilities

3.1 Introduction

Part of the wireless security problem is that wireless's greatest strengths, in many ways, are also its greatest weaknesses. "Access is more convenient, but you have quite a bit less control" says Dain Gary, chief security officer at RedSiren, Pittsburgh.

When transmissions are broadcast over radio waves, interception and masquerading becomes trivial to anyone with a radio, and so there is a need to employ additional mechanisms to protect the communications.

The 802.11 standard for wireless LAN communications introduced the *Wired Equivalent Privacy* (**WEP**) protocol in an attempt to address these new problems bringing the security level of wireless systems closer to that of wired ones. The primary goal of WEP is to protect the confidentiality of user data from eavesdropping. WEP is part of an international standard; it has been integrated by manufacturers into their 802.11 hardware and is currently in widespread use. Unfortunately, WEP falls short of accomplishing its security goals. Despite employing the well-known and believed-secure RC4 cipher, WEP contains several major security flaws. The flaws give rise to a number of attacks, both passive and active, that allow eavesdropping on, and tampering with, wireless transmissions.

In the following sections, the WEP flaws are described in details.

3.2 Attack Practicality

Despite being transmitted over open radio waves, 802.11 traffic requires significant infrastructure to intercept. An attacker needs equipment capable of monitoring 2.4GHz frequencies and understanding the physical layer of the

802.11 protocol; for active attacks, it is also necessary to transmit at the same frequencies.

The necessary hardware to monitor and inject 802.11 traffic is readily available to consumers in the form of wireless Ethernet interfaces. All that is needed is to subvert it to monitor and transmit encrypted traffic.

Everyone can successfully carry out passive attacks using off-the-shelf equipment. Essentially what is required when sniffing any medium is to turn the NIC (Network Interface Card) into promiscuous mode; this allows the computer to "see" any packet it encounters on the medium. But the card must also be able to make sense of these packets: this is where the sniffing software comes into play.

Kismet is a passive wireless sniffing program for Linux and BSD. It will intercept, organize and log all data passing through the wireless medium. Kismet uses a true monitor mode to gather the full packet contents. (<http://www.kismetwireless.net>)

AirSnort runs under Linux and is one of the publicly available implementations of the Scott Fluhrer, Itsik Mantin and Adi Shamir WEP attack. (<http://airsnort.shmoo.com>)

Netstumbler rely upon the probe-request and response mechanism to gather SSIDs (Wireless Network Identification Names). (<http://www.netstumbler.org>)

Active attacks appear to be more difficult, but not beyond reach. The PCMCIA Orinoco cards produced by Lucent allow their firmware to be upgraded; a concerted reverse-engineering effort should be able to produce a modified version that allows injecting arbitrary traffic. The time investment required is non-trivial; however, it is a one-time effort; the rogue firmware can then be posted on a web site or distributed amongst underground circles.

Therefore, we believe that it would be prudent to assume that motivated attackers will have full access to the link layer for passive and even active attacks.

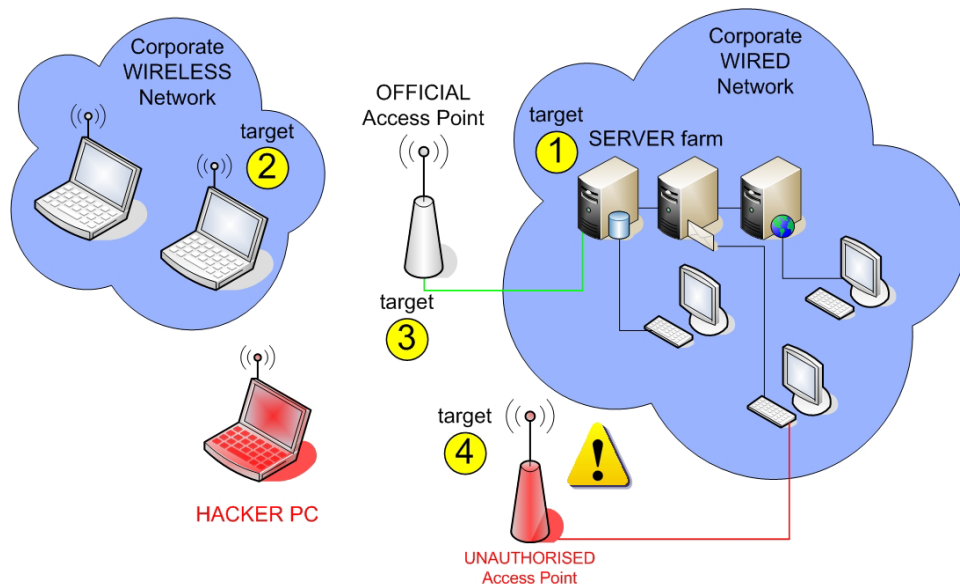


Figure 3.1: Hacker's targets

3.3 The Risks of Keystream Reuse

WEP provides data confidentiality using a stream cipher called RC4. Stream ciphers operate by expanding a secret key (or, as in the case of WEP, a public IV¹ and a secret key) into an arbitrarily long "keystream" of pseudorandom bits. Encryption is performed by XORing the generated keystream with the plaintext. Decryption consists of generating the identical keystream based on the IV and secret key and XORing it with the ciphertext.

A well-known weakness of stream ciphers is that encrypting two messages under the same IV and key can reveal information about both messages:

$$\begin{aligned}
 \text{If} \quad & C_1 = P_1 \oplus \text{RC4}(v, k) \\
 \text{and} \quad & C_2 = P_2 \oplus \text{RC4}(v, k) \\
 \text{then} \quad & C_1 \oplus C_2 = (P_1 \oplus \text{RC4}(v, k)) \oplus (P_2 \oplus \text{RC4}(v, k)) \\
 & = P_1 \oplus P_2.
 \end{aligned}$$

In other words, XORing the two ciphertexts (C_1 and C_2) together causes the keystream to cancel out, and the result is the XOR of the two plaintexts ($P_1 \oplus P_2$).

Thus, keystream reuse can lead to a number of attacks: as a special case, if the plaintext of one of the messages is known, the plaintext of the other

¹as explained before, on page 27 in the paragraph Security: the WEP Protocol)

is immediately obtainable. More generally, real-world plaintexts often have enough redundancy that one can recover both P_1 and P_2 given only $P_1 \oplus P_2$. Moreover, if we have n ciphertexts that all reuse the same keystream, we have what is known as a problem of depth n . Reading traffic in depth becomes easier as n increases, since the pairwise XOR of every pair of plaintexts can be computed, and many classical techniques are known for solving such problems (e.g., frequency analysis, dragging cribs, and so on). Note that there are two conditions required for this class of attacks to succeed:

- The availability of ciphertexts where some portion of the keystream is used more than once, and
- Partial knowledge of some of the plaintexts.

To prevent these attacks, WEP uses a per-packet IV to vary the keystream generation process for each frame of data transmitted. WEP generates the keystream $RC4(v, k)$ as a function of both the secret key k (which is the same for all packets) and a public initialization vector v (which varies for each packet); this way, each packet receives a different keystream. The IV is included in the unencrypted portion of the transmission so that the receiver can know what IV to use when deriving the keystream for decryption. The IV is therefore available to attackers as well, but the secret key remains unknown and maintains the security of the keystream.

The use of a per-packet IV was intended to prevent keystream reuse attacks. Nonetheless, WEP does not achieve this goal. Below several realistic keystream reuse attacks on WEP are described. First, it's discussed how to find instances of keystream reuse; then, it's shown how to exploit these instances by taking advantage of partial information on how typical plaintexts are expected to be distributed.

Finding instances of keystream reuse

One potential cause of keystream reuse is the improper IV management. Note that, since the shared secret key generally changes very rarely, reuse of IV's almost always causes reuse of some of the RC4 keystream. Since IV's are public, duplicate IV's can be easily detected by the attacker. Therefore, any reuse of old IV values exposes the system to keystream reuse attacks. A reuse of an IV value is called "collision".

The WEP standard recommends (but does not require) that the IV be changed after every packet. However, it does not say anything else about how to select IV's. The most of the PCMCIA cards available on the market reset the IV to 0 each time they were re-initialized, and then incremented the

IV by one for each packet transmitted. These cards re-initialize themselves each time they are inserted into the laptop, which can be expected to happen fairly frequently. Consequently, keystreams corresponding to low-valued IV's were likely to be reused many times during the lifetime of the key.

Even worse, the WEP standard has architectural flaws that expose all WEP implementations to serious risks of keystream reuse. The IV field used by WEP is only 24 bits wide, nearly guaranteeing that the same IV will be reused for multiple messages.

A back-of-the-envelope calculation shows that a busy access point sending 1500 byte packets and achieving an average 5Mbps bandwidth (the full transmission rate is 11Mbps) will exhaust the available space in less than half a day. Even for less busy installations, a patient attacker can readily find duplicates. Because the IV length is fixed at 24 bits in the standard, this vulnerability is fundamental: no compliant implementation can avoid it.

Implementation details can make keystream reuse occur even more frequently. An implementation that uses a random 24-bit IV for each packet will be expected to incur collisions after transmitting just 5000 packets (this is a consequence of the so called birthday paradox²), which is only a few minutes of transmission. Worse yet, the 802.11 standard does not even require that the IV be changed with every packet, so an implementation could reuse the same IV for all packets.

Exploiting keystream reuse to read encrypted traffic

Once two encrypted packets that use the same IV are discovered, various methods of attack can be applied to recover the plaintext. If the plaintext of one of the messages is known, it is easy to derive the contents of the other one directly.

There are many ways to obtain plausible candidates for the plaintext. Many fields of IP traffic are predictable, since protocols use well-defined structures in messages, and the contents of messages are frequently predictable. For example, login sequences are quite uniform across many users, and so the contents e.g., the **Password:** prompt or the welcome message may be known to the attacker and thus usable in a keystream reuse attack. As another example, it may be possible to recognize a specific shared library being transferred from a networked file system by analyzing traffic patterns and lengths; this would provide a large quantity of known plaintext suitable for use in a keystream reuse attack.

²This paradox says that the probability that at least two of 23 randomly selected people have the same birthday (same month and day but not necessary the same year) is greater than 50%.

There are also other ways to obtain known plaintext. It is possible to cause known plaintext to be transmitted by, for example, sending IP traffic directly to a mobile host from an Internet host under the attacker's control (see figure 3.2). The attacker may also send e-mail to users and wait for them to check it over a wireless link. Sending spam e-mail might be a good method of doing this without raising too many alarms.

Sometimes, obtaining known plaintext in this way may be even simpler. One access point would transmit broadcast packets in both encrypted and unencrypted form, when the option to control network access was disabled. In this scenario, an attacker with a conforming 802.11 interface can transmit broadcasts to the access point (they will be accepted, since access control is turned off) and observe their encrypted form as they are re-transmitted. Indeed, this is unavoidable on a subnet that contains a mixture of WEP clients with and without support for encryption: since broadcast packets must be forwarded to all clients, there is no way to avoid this technique for getting known plaintext.

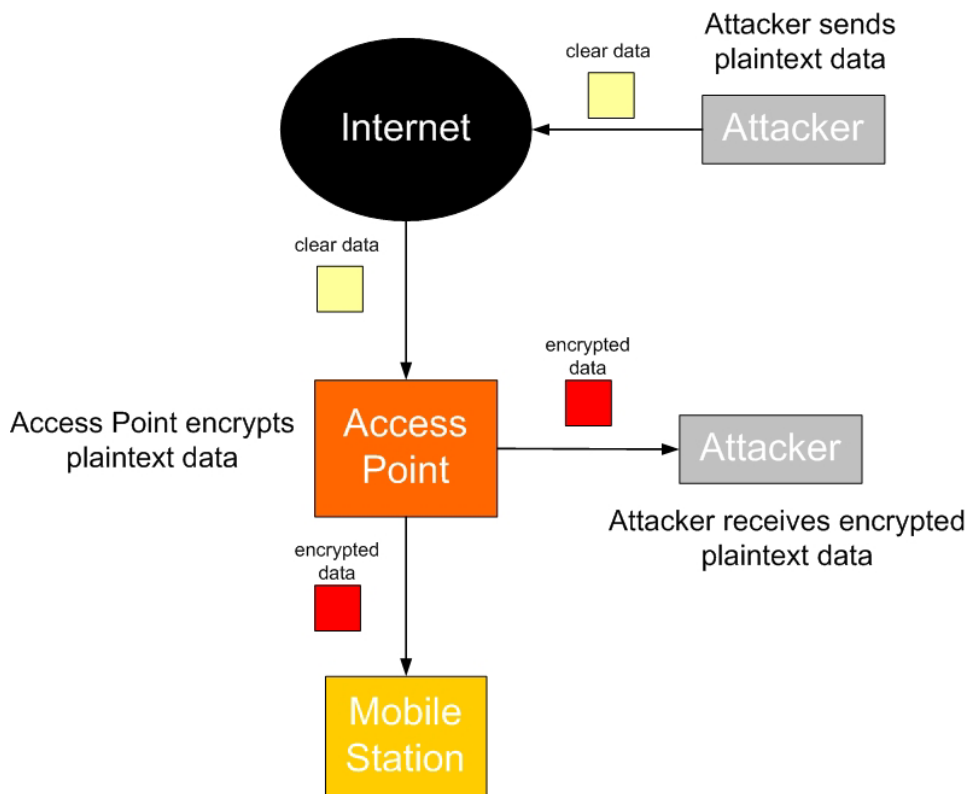


Figure 3.2: Forcing a known plaintext to be transmitted

3.3.1 Decryption Dictionaries

Once the plaintext for an intercepted message is obtained, either through analysis of colliding IV's, or through other ways, the attacker also learns the value of the keystream used to encrypt the message. It is possible to use this keystream to decrypt any other message that uses the same IV. Over time, the attacker can build a table of the keystreams corresponding to each IV. The full table has modest space requirements (perhaps 1500 bytes for each of the 2^{24} possible IV's, or roughly 24 GB) so it is possible that a dedicated attacker can, after some amount of effort, accumulate enough data to build a full decryption dictionary, especially when one considers the low frequency with which keys are changed. The advantage to the attacker is that, once such a table is available, it becomes possible to immediately decrypt each subsequent ciphertext with very little work.

Of course, the amount of work necessary to build such a dictionary restricts this attack to only the most persistent attackers who are willing to invest time and resources into defeating WEP security. It can be argued that WEP is not designed to protect from such attackers, since a 40-bit key can be discovered through brute-force in a relatively short amount of time with moderate resources. However, manufacturers have already extended WEP to support larger keys (e.g. 128 bit), and the dictionary attack is effective regardless of key size. (The size of the dictionary depends not on the size of the key, but only on the size of the IV, which is fixed by the standard at 24 bits.)

Further, the dictionary attack can be made by exploiting the behavior of PCMCIA cards that reset the IV to 0 each time they are reinitialized. Since typical use of PCMCIA cards includes reinitialization at least once per day, building a dictionary for only the first few thousand IV's will enable an attacker to decrypt most of the traffic directed towards the access point. In an installation environment with many 802.11 clients, collisions in the first few thousands IV's would be as much as needed by the attacker.

IV	Ciphertext
IV0	C0,1
...	...
IV31	C0,31
...	...
IVn	C0,n

Table 3.1: A decryption dictionary

3.3.2 Key Management

The 802.11 standard does not specify how distribution of keys is to be accomplished. It relies on an external mechanism to populate a globally-shared array of 4 keys. Each message contains a key identifier field specifying the index in the array of the key being used. The standard also allows for an array that associates a unique key with each mobile station; however, this option is not widely supported.

In practice, most installations use a single key for an entire network. This practice seriously affects the security of the system, since a secret that is shared among many users cannot stay very well hidden. Some network administrators try to overcome this problem by not revealing the secret key to end users, but rather configuring their machines with the key themselves. This, however, leads only to a marginal improvement, since the keys are still stored on the users computers.

The reuse of a single key by many users also helps make the attacks in this section more practical, since it increases chances of IV collision. The chance of random collisions increases proportionally to the number of users; even worse, PCMCIA cards that reset the IV to 0 each time they are reinitialized will all reuse keystreams corresponding to a small range of low-numbered IV's.

Also, the fact that many users share the same key means that it is difficult to replace compromised key material. Since changing a key requires every single user to reconfigure their wireless network drivers, such updates will be infrequent. In practice, it may be months, or even longer, between key changes, allowing an attacker more time to analyze the traffic and look for instances of keystream reuse.

3.4 Message Authentication

The WEP protocol uses an integrity checksum field to ensure that packets do not get modified in transit. The checksum is implemented as a CRC-32 checksum, which is part of the encrypted payload of the packet.

CRC checksum is insufficient to ensure that an attacker cannot tamper with a message: it is not a cryptographically secure authentication code. CRC's are designed to detect random errors in the message; however, they are not resilient against malicious attacks. This vulnerability of CRC is due by the fact that the message payload is encrypted using a stream cipher.

3.4.1 Message Modification

First, it's shown that messages may be modified in transit without detection, in violation of the security goals. This is possible, using the following property of the WEP checksum:

Property 1 *The WEP checksum is a linear function of the message.*

By this, we mean that checksumming distributes over the XOR operation, i.e., $c(x \oplus y) = c(x) \oplus c(y)$ for all choices of x and y . This is a general property of all CRC checksums.

One consequence of the above property is that it becomes possible to make controlled modifications to a ciphertext without disrupting the checksum. Let's fix our attention on a ciphertext C which is intercepted before it could reach its destination as shown in figure 3.3:

$$A \rightarrow (B) : \langle v, C \rangle.$$

We assume that C corresponds to some unknown message M , so that

$$C = \text{RC4}(v, k) \oplus \langle M, c(M) \rangle. \quad (3.1)$$

We claim that it is possible to find a new ciphertext C' that decrypts to M' , where $M' = M \oplus \Delta$ and Δ may be chosen arbitrarily by the attacker. Then, we will be able to replace the original transmission with our new ciphertext by spoofing the source,

$$(A) \rightarrow B : \langle v, C' \rangle,$$

and upon decryption, the recipient B will obtain the modified message M' with the correct checksum.

All that remains is to describe how to obtain C' from C so that C' decrypts to M' instead of M . The key observation is to note that stream ciphers, such as RC4, are also linear, so we can reorder many terms. The following trick is suggested: XOR the quantity $\langle \Delta, c(\Delta) \rangle$ against both sides of Equation 3.1 above to get a new ciphertext C' :

$$\begin{aligned} C' &= C \oplus \langle \Delta, c(\Delta) \rangle \\ &= \text{RC4}(v, k) \oplus \langle M, c(M) \rangle \oplus \langle \Delta, c(\Delta) \rangle \\ &= \text{RC4}(v, k) \oplus \langle M \oplus \Delta, c(M) \oplus c(\Delta) \rangle \\ &= \text{RC4}(v, k) \oplus \langle M', c(M \oplus \Delta) \rangle \\ &= \text{RC4}(v, k) \oplus \langle M', c(M') \rangle. \end{aligned}$$

In this derivation, we used the fact that the WEP checksum is linear, so that $c(M) \oplus c(\Delta) = c(M \oplus \Delta)$. As a result, we have shown how to modify C to obtain a new ciphertext C' that will decrypt to $P \oplus \Delta$.

This implies that it's possible to make arbitrary modifications to an encrypted message without fear of detection. Thus, the WEP checksum fails to protect data integrity, one of the three main goals of the WEP protocol.

Notice that this attack can be applied without full knowledge of M : the attacker only needs to know the original ciphertext C and the desired plaintext difference Δ , in order to calculate $C' = C \oplus \langle \Delta, c(\Delta) \rangle$. For example, to flip the first bit of a message, the attacker can set $\Delta = 1000 \dots 0$. This allows an attacker to modify a packet with only partial knowledge of its contents.

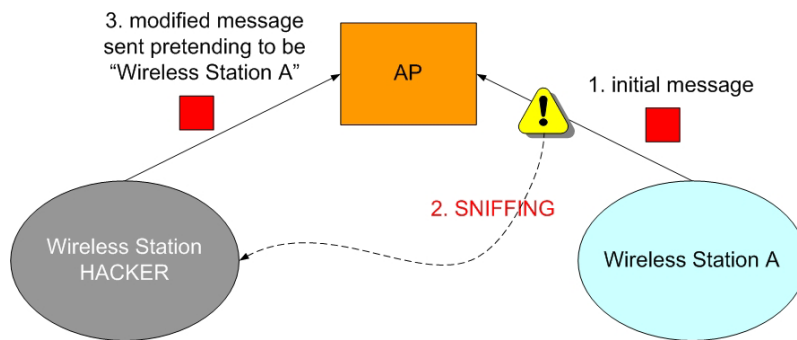


Figure 3.3: Message modification

3.4.2 Message Injection

Next, it's shown that WEP does not provide secure access control, using the following property of the WEP checksum:

Property 2 *The WEP checksum is an unkeyed function of the message.*

As a consequence, the checksum field can also be computed by the adversary who knows the message. This property of the WEP integrity checksum allows the circumvention of access control measures. If an attacker can get ahold of an entire plaintext corresponding to some transmitted frame, he will then be able to inject arbitrary traffic into the network. As it was explained in the previous section, knowledge of both the plaintext and ciphertext reveals the keystream. This keystream can subsequently be reused to create a new packet, using the same IV. That is, if the attacker ever learns the complete

plaintext P of any given ciphertext packet C , he can recover keystream used to encrypt the packet:

$$P \oplus C = P \oplus (P \oplus \text{RC4}(v, k)) = \text{RC4}(v, k).$$

He can now construct an encryption of a message M' :

$$(A) \rightarrow B : \langle v, C' \rangle,$$

where

$$C' = \langle M', c(M') \rangle \oplus \text{RC4}(v, k).$$

Note that the rogue message uses the same IV value as the original one. Therefore, the attack works only because of the following behavior of WEP access points:

Property 3 *It is possible to reuse old IV values without triggering any alarms at the receiver.*

When an attacker knows an IV along with its corresponding keystream sequence $\text{RC4}(v, k)$, this IV-reuse property is what allows to reuse known keystream and circumvent the WEP access control mechanism.

A natural defense against this attack would be to disallow the reuse of IV's in multiple packets and require that all receivers enforce this prohibition. However, the 802.11 standard does not do this. While the 802.11 standard strongly recommends against IV reuse, it does not require it to change with every packet. Hence, every receiver must accept repeated IV's or risks non-interoperability with compliant devices.

Note that this attack does not rely on Property 1 of the WEP checksum (linearity). In fact, substituting any unkeyed function in place of the CRC will have no effect on the viability of the attack.

3.4.3 Message Decryption

What may be surprising is that the ability to modify encrypted packets without detection can also be leveraged to decrypt messages sent over the air. Consider WEP from the point of view of the adversary. Since WEP uses a stream cipher presumed to be secure (RC4), attacking the cryptography directly is probably hopeless. But if an attacker cannot decrypt the traffic by himself, there is still someone who can: the access point. In any cryptographic protocol, the legitimate decryptor must always possess the secret key in order to decrypt, by design. The idea, then, is to trick the access point into decrypting some ciphertext for us. As it turns out, the ability to modify transmitted packets provides two easy ways to exploit the access point in this way.

IP redirection

The first way is called "IP redirection" attack, and can be used when the WEP access point acts as an IP router with Internet connectivity; note that this is a fairly common scenario in practice, because WEP is typically used to provide network access for mobile laptop users and others.

In this case, the idea is to sniff an encrypted packet off the air, and use the technique of "Message Modification" to modify it so that it has a new destination address: one the attacker controls. The access point will then decrypt the packet, and send the packet off to its (new) destination, where the attacker can read the packet, now in the clear. Note that the modified packet will be travelling *from* the wireless network *to* the Internet, and so most firewalls will allow it to pass unmolested (see figure 3.4).

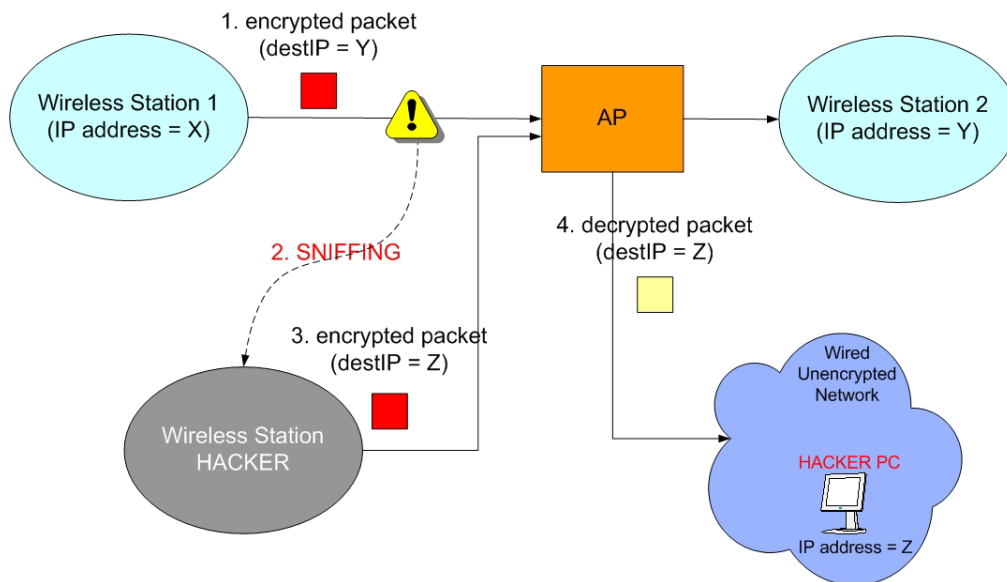


Figure 3.4: IP redirection

The easiest way to modify the destination IP address is to figure out what the original destination IP address is, and then apply the technique of "Message Modification" to change it to the desired one. Figuring out the original destination IP address is usually not difficult; all of the incoming traffic, for example, will be destined for an IP address on the wireless subnet, which should be easy to determine. Once the incoming traffic is decrypted, the IP addresses of the other ends of the connections will be revealed, and outgoing traffic can then be decrypted in the same manner.

In order for this attack to work, however, it is needed to not only modify the destination IP address, but also to ensure that the IP checksum in the modified packet is still correct, otherwise the decrypted packet will be dropped by the access point. Since the modified packet differs from the original packet only in its destination IP address, and since both the old and new values for the destination IP address are known, we can calculate the required change to the IP checksum caused by this change in IP address. Suppose the high and low 16-bit words of the original destination IP address were D_H and D_L , and we wish to change them to D'_H and D'_L . If the old IP checksum was χ (which we do not necessarily know, since it is encrypted), the new one should be

$$\chi' = \chi + D'_H + D'_L - D_H - D_L$$

(where the additions and subtractions here and below are one's-complement).

The trick is that we only know how to modify a packet by applying an XOR to it, and we don't necessarily know what we need to XOR to χ to get χ' , even though we *do* know what we would need to *add* (namely, $D'_H + D'_L - D_H - D_L$).

Three ways to try to correct the IP checksum of the modified packet are discussed below:

1) The IP checksum for the original packet is known:

If it happens to be the case that the attacker somehow knows χ , then he simply calculates χ' as above, and modifies the packet by XORing in $\chi \oplus \chi'$, which will change the IP checksum to the correct value of χ' .

2) The original IP checksum is not known:

If χ is not known, the task is harder. Given $\xi = \chi' - \chi$, it's needed to calculate $\Delta = \chi' \oplus \chi$. In fact, there is not enough information to calculate Δ given only ξ . For example, if $\xi = 0x\text{CAFE}$, it could be that:

- $\chi' = 0x\text{CAFE}$, $\chi = 0x0000$, so $\Delta = 0x\text{CAFE}$
- $\chi' = 0xD00D$, $\chi = 0xD502$, so $\Delta = 0xD502$
- $\chi' = 0x1EE7$, $\chi = 0x53E8$, so $\Delta = 0x4D0F$
- ...

However, not all 2^{16} values for Δ are possible, and some are much more likely than others. In the above example, there are four values for Δ ($0x3501$,

0x4B01, 0x4D01, 0x5501) which occur more than 3% of the time each. Further, the attacker is free to make multiple attempts any incorrect guesses will be silently ignored by the access point. Depending on the value of ξ , a small number of attempts can succeed with high probability. Finally, a successful decryption of one packet can be used to bootstrap the decryption of others; for example, in a stream of communication between two hosts, the only field in the IP header that changes is the identification field. Thus, knowledge of the full IP header of one packet can be used to predict the full header of the surrounding packets, or narrow it down to a small number of possibilities.

3) Arrange that $\chi = \chi'$:

Another possibility is to compensate for the change in the destination field by a change in another field, such that the checksum of the packet remains the same. Any header field that is known and does not affect packet delivery is suitable; for example, the source IP address. Assuming the source IP address of the packet to be decrypted (we can obtain it, for example, by performing the attack in the previous item on one packet to decrypt it completely, and then using this simpler attack on subsequent packets once we read the source address from the first one), the attacker simply subtracts 1 from the low 16-bit word of the source IP address, and the resulting packet will have the same IP checksum as the original. However, it is possible that modifying the source address in this way will cause a packet to be dropped based on egress filtering rules; other header fields could potentially be used instead.

Highly resourceful attackers with monitoring access to an entire class B network can even perform the necessary adjustments in the destination field alone, by choosing $D'_L = D_H + D_L - D'_H$. For example, if the original destination address in a packet is 10.20.30.40 and the attacker holds control over the 192.168.0.0/16 subnet, selecting the address 192.168.103.147 results in identical IP header checksum values, and the packet will be delivered to an address he controls.

Reaction attacks

There is another way to manipulate the access point and break WEP-encrypted traffic that is applicable whenever WEP is used to protect TCP/IP traffic. This attack does not require connectivity to the Internet, so it may apply even when IP redirection attacks are impossible. However, it is effective only against TCP traffic; other IP protocols cannot be decrypted using this attack.

In this attack, an attacker monitors the reaction of a recipient to a TCP packet and use what he observes to infer information about the unknown

plaintext. This attack relies on the fact that a TCP packet is accepted only if the TCP checksum is correct, and when it is accepted, an acknowledgement packet is sent in response. Note that acknowledgement packets are easily identified by their size, without requiring decryption. Thus, the reaction of the recipient will disclose whether the TCP checksum was valid when the packet was decrypted.

The attack, then, proceeds as shown in figure 3.5. We intercept a ciphertext $\langle v, C \rangle$ with unknown decryption P :

$$A \rightarrow (B) : \langle v, C \rangle.$$

We flip a few bits in C and adjust the encrypted CRC accordingly to obtain a new ciphertext C' with valid WEP checksum. We transmit C' in a forged packet to the access point:

$$(A) \rightarrow B : \langle v, C' \rangle.$$

Finally, we watch to see whether the eventual recipient sends back a TCP ACK (acknowledgement) packet; this will allow us to tell whether the modified text passed the TCP checksum and was accepted by the recipient.

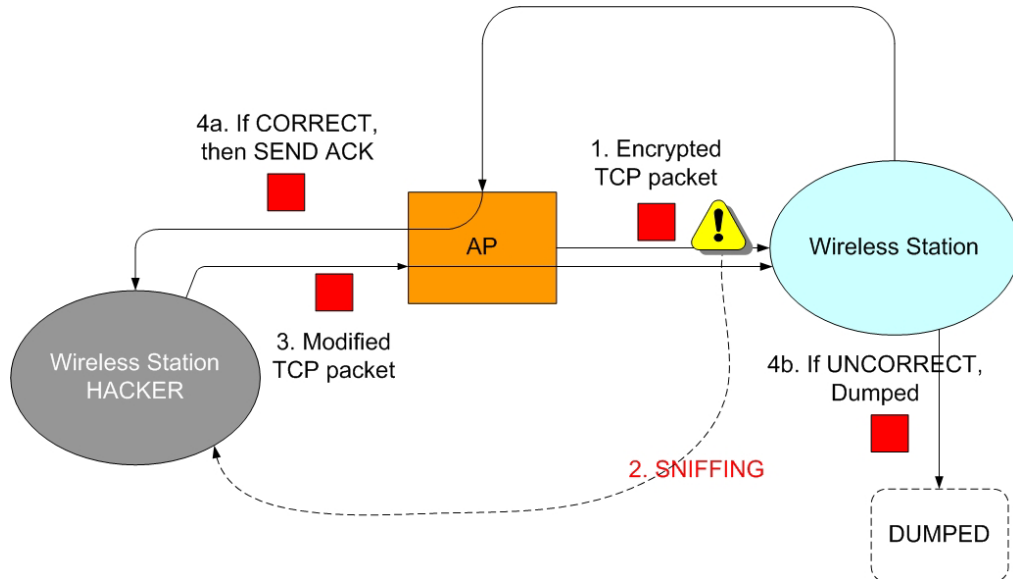


Figure 3.5: Reaction attack

Note that we may choose which bits of C to flip in any way we like, using techniques from subsection 3.4.1. The key technical observation is

as follows: by a clever choice of bit positions to flip, we can ensure that the TCP checksum remains undisturbed exactly when the one-bit condition $P_i \oplus P_{i+16} = 1$ on the plaintext holds. Thus, the presence or absence of an ACK packet will reveal one bit of information on the unknown plaintext. By repeating the attack for many choices of i , we can learn almost all of the plaintext P , and then deducing the few remaining unknown bits will be easy using classical techniques.

In the next section it's explained precisely how to choose which bits to flip. For now, the details are not terribly important. Instead, the main point is that we have exploited the receiver's willingness to decrypt arbitrary ciphertexts and feed them to another component of the system that leaks a tiny bit of information about its inputs. The recipient's reaction to our forged packet, either acknowledging or ignoring it, can be viewed as a side channel, similar to those exploited in timing and power consumption attacks, that allows us to learn information about the unknown plaintext. Thus, we have used the recipient as an oracle to unknowingly decrypt the intercepted ciphertext for us. This is known as a *reaction attack*, as it works by monitoring the recipient's reaction to our forgeries.

Reaction attacks were initially discovered by Bellare and Wagner in the context of the IP Security protocol, where their existence was blamed on the use of encryption without also using a MAC for message authentication. As a result, Bellare proposed a design principle for IP Security: all encryption modes of operation should also use a MAC. It seems that the same rule of thumb applies to the WEP protocol as well, for the presence of a secure MAC (rather than the insecure CRC checksum) would have prevented these attacks.

The technical details

Here there are technical details on how to choose new forged packets C' to trick the recipient into revealing information about the unknown plaintext P .

Recall that the TCP checksum is the one's-complement addition of the 16-bit words of the message M . Moreover, one's-complement addition behaves roughly equivalently to addition modulo $2^{16} - 1$. Hence, roughly speaking, the TCP checksum on a plaintext P is valid only when $P \equiv 0 \pmod{2^{16} - 1}$.

We let $C' = C \oplus \Delta$, so that Δ specifies which bit positions to flip, and we choose Δ as follows: pick i arbitrarily, set bit positions i and $i + 16$ of Δ to one, and let Δ be zero elsewhere. It is a convenient property of addition modulo $2^{16} - 1$ that $P \oplus \Delta \equiv P \pmod{2^{16} - 1}$ holds exactly when $P_i \oplus P_{i+16} = 1$. Since we assume that the TCP checksum is valid for the original packet (i.e.,

$P \equiv 0 \pmod{2^{16} - 1}$, this means that the TCP checksum will be valid for the new packet (i.e., $P \oplus \Delta \equiv P \pmod{2^{16} - 1}$) just when $P_i \oplus P_{i+16} = 1$. This gives us our one bit of information on the plaintext, as claimed.

3.5 Fluhrer, Mantin, and Shamir Attack

The Fluhrer, Mantin, and Shamir known IV attack utilizes the fact that, in some cases, knowledge of the IV and the first output byte leaks information about the key bytes. They refer to these key-leaking cases as *resolved*.

The implementation of RC4 in WEP was found to have weak keys. Having a weak key means that there is more correlation between the key and the output than there should be for good security. If a packet is encrypted using a weak key, the first three bytes of the key are taken from the IV that is sent unencrypted with the packet. Out of the 16 millions IV values, around 9000 of them are considered to be weak IV. This weakness can be exploited by a passive attack.

The attacker captures as much traffic as possible. Once done, it retrieves the packets with weak IV that will correspond to weak keys. It will need to try a much small number of keys and of generated packets to determine the key. It has been proved that to determine a 104 bit WEP key, you have to capture between 2000 and 4000 interesting packets. On an enterprise network, millions of packets are transmitted per day so it is only a matter of days or weeks to break a so-called 128 bit key. Some manufacturers have started not to use weak IV values in their IV generation algorithm. But if only one station on the network uses a weak IV, then this attack can succeed.

AirSnort is an open source software that was released in August 2001 as the first public implementation of the Fluhrer/Mantin/Shamir attack against WEP. AirSnort requires approximately 5-10 million encrypted packets to be gathered to break the network key.

Algorithm:

1. AirSnort captures all incoming frames and looks for frames with a weak initialization vector.
2. The program then matches the IV with a familiar key byte. It categorizes the samples against each key byte and uses these samples to try to crack the key. Less than 100 samples are needed to obtain a key.

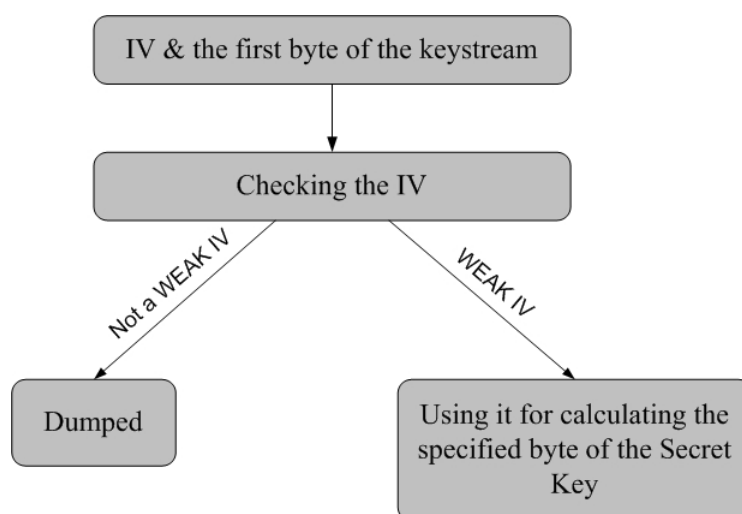


Figure 3.6: Working of airsnort

3.6 Conclusions

The attacks in these sections demonstrate that the use of stream ciphers is dangerous, because the reuse of keystream can have devastating consequences. Any protocol that uses a stream cipher must take special care to ensure that keystream *never* gets reused. This property can be difficult to enforce.

For this reason, many of the same precautions and security measures used in the wired world must also be applicable in a wireless environment. The deployment of firewalls, VPNs, encryption and hardware security as well as the development of comprehensive security policies and regular network monitoring are all part of an effective wireless security program.

Chapter 4

IPSec and L2TP

4.1 Introduction on IPSec

The *Internet Protocol* (**IP**) underlies the vast majority of large corporate and academic networks as well as the Internet. It is flexible, powerful and has served people's networking needs well for decades. IP's strength lies in its easily and flexibly routed packets.

However, IP's strength is also its weakness. The way IP routes packets makes large IP networks vulnerable to a range of security risks:

- **spoofing**, in which one machine on the network masquerades as another
- **sniffing**, in which an eavesdropper listens in on a transmission between two other parties
- **session hijacking**, in which a sophisticated attacker employing both those techniques takes over an established communications session and masquerades as one of the communicating parties.

Because these vulnerabilities limit and complicate the use of large IP networks (including the Internet) for sensitive communications, an international group organized under the *Internet Engineering Task Force* (**IETF**) has developed the *IP Security* (**IPSec**) protocol suite, a set of IP extensions that provide security services at the network level. IPSec technology is based on modern cryptographic technologies, making possible very strong data authentication and privacy guarantees.

The IPSec group's work is conceptually unique in that it seeks to secure the network itself, rather than the applications that use it. Because it secures the network itself, the IPSec protocol suite guarantees security for any application using the network. IPSec makes possible the realization of the

secure Virtual Private Network (secure VPN), a private and secure network carved out of a public and/or insecure network. Secure VPNs are as safe as isolated office LANs or WANs run entirely over private lines and far more cost-effective.

The IPsec protocol suite provides three overall pieces:

- an *authentication header (AH)* for IP that lets communicating parties verify that data was not modified in transit and that it genuinely came from its apparent source
- an *encapsulation security payload (ESP)* format for IP that encrypts data to secure it against eavesdropping during transit
- a *protocol negotiation and key exchange protocol (IKE)* that allows communicating parties to negotiate methods of secure communication.

The fundamental strength of the IPsec group's approach is that their security works at a low network level. So just as IP is transparent to the average user, so are IPsec-based security services, unseen servants functioning in the background, ensuring that your communications are secure.

Just as IP's power and flexibility make it universal, IPsec promises to become *the new international standard*, answering to a diverse range of security needs, allowing vastly different networks around the world to interconnect and to communicate securely.

4.2 IPsec modes

IPsec protocols supports two modes, **transport mode** and **tunnel mode**, and each IPsec security services protocol can be used in either mode.

Transport mode

Transport mode is designed for host-to-host communication and does not afford total protection for the IP packets transmitted between the two hosts. In transport mode the security protocol header is inserted between the IP header and the upper layer protocol header, protecting only the upper layer payload of the packet. Transport mode cannot be applied at gateways and routers, where the communication and cryptographic endpoints are not necessarily the same entity. This is due to how the packet is constructed, fragmented and reassembled.

Tunnel mode

To protect the entire IP packet, tunnel mode is used. In tunnel mode the packet to be secured is "wrapped" in a new IP packet, and both the header and the payload of the original packet are afforded IPSec protection. This is also known as *IP-in-IP tunnelling*, and is typically used in security gateways. In tunnel mode the communication and cryptographic endpoints need not be the same. The gateway could be the cryptographic endpoint, where the IPSec processing is conducted. Once the IPSec protected packet has been decrypted by the gateway, the original packet is forwarded to the communication endpoint.

Tunneling services provide also for address resolution and address hiding between private networks. An IPSec-compliant security gateway can take an entire IP packet from a node within the network it protects and encapsulate it inside a new IP packet, before sending it out through the public network.

This feature has two main applications:

- it can allow nodes that have an illegal IP address (meant only for internal use) to communicate with other nodes across a public network
- it can conceal the addresses of sensitive internal nodes (with legal addresses), protecting them even against denial of service attacks.

Figures 4.1 and 4.2 illustrate the two IPSec protocols applied in transport and tunnel modes.

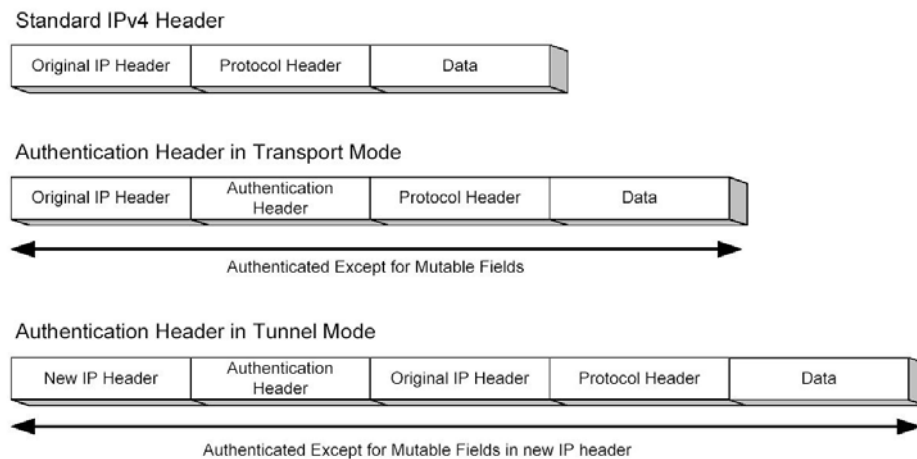


Figure 4.1: Authentication Header in Transport and Tunnel Modes

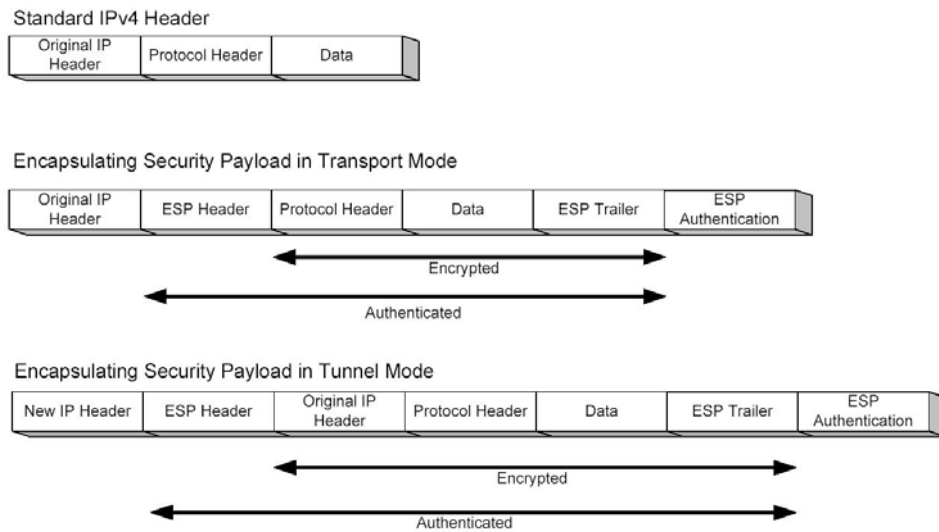


Figure 4.2: Encapsulating Security Payload in Transport and Tunnel Modes

4.3 Authentication, Integrity and Confidentiality Services

The basic building blocks of IPSec, the *Encapsulating Security Payload (ESP)* protocol and the *Authentication Header (AH)* protocol, use cryptographic techniques for ensuring data confidentiality and digital signatures for authenticating the data's source.

4.3.1 ESP

The *IP datagram*, or *IP packet*, is the fundamental unit of communications in IP networks. IPSec handles the encryption at the packet level. The protocol it uses is called **ESP**.

ESP supports pretty much any kind of symmetric encryption. The default standard built into ESP that assures basic interoperability is *56-bit DES*. ESP also supports some authentication (as can the AH, the two have been designed with some overlap).

The ESP (see figure 4.3) follows the standard IP header in an IP datagram, and contains both the data and all higher level protocol headers relying on IP for routing. The figure does not show the IP header.

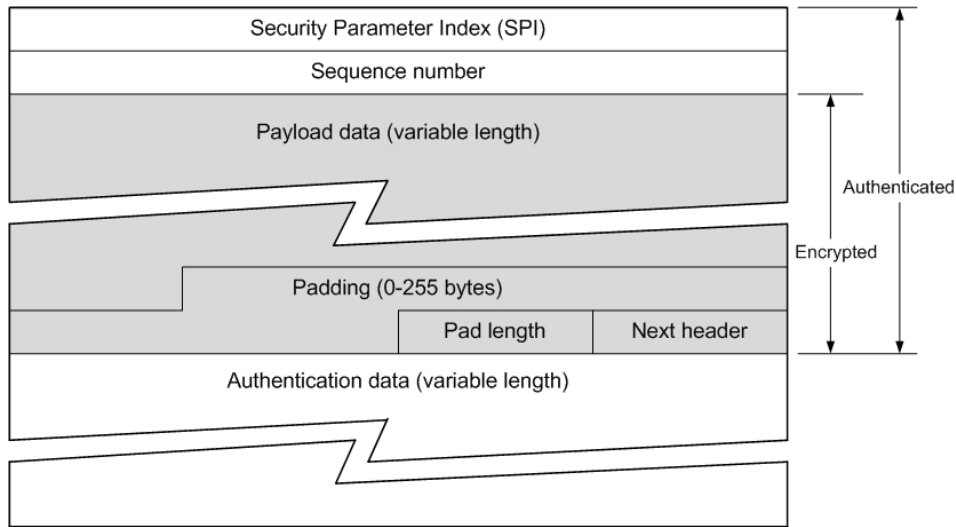


Figure 4.3: The Encapsulating Security Payload format

The ESP contains six parts. The first two parts are not encrypted but are authenticated:

- The *Security Parameter Index (SPI)* is an arbitrary 32-bit number that specifies to the device receiving the packet what group of security protocols the sender is using for communication (which algorithms, which keys, and how long those keys are valid).
- The *Sequence number* is a counter that increases each time a packet is sent to the same address using the same SPI. It indicates which packet is which and how many packets have been sent with the same group of parameters. The sequence number provides protection against replay attacks, in which an attacker copies a packet and sends it out of sequence, to confuse communicating nodes.

The remaining parts (with the exception of the authentication data) are all encrypted during transmission across the network. When unencrypted, they look like this:

- The *Payload data* is the actual data being carried by the packet.
- The *Padding* (from 0 to 255 bytes of data) allows, for the fact that certain types of encryption algorithms require the data to be a multiple of a certain number of bytes, to confuse sniffers trying to estimate how much data is being transmitted.

- The *Pad length* field specifies how much of the payload is padding as opposed to data.
- The *Next header* field, like a normal IP Next header field, identifies the type of data carried and the protocol above.
- The final field is an *authentication field*.

Note that the ESP is added after a standard IP header (one that contains as its protocol field a number that says there's an ESP following, instead of a TCP header). Because the packet has a standard IP header, the network can route it with standard IP equipment. So IPSec is backwards-compatible with IP routers and other equipment that isn't yet become IPSec aware.

ESP can support any number of encryption protocols; it's up to the user to decide which ones to use. It's possible to use different protocols for each party with whom you're communicating. But IPSec specifies a basic **DES-CBC** (*Cipher Block Chaining mode*) cipher as its default, to guarantee a minimal interoperability among IPSec networks. ESP's encryption support is designed for use by symmetric encryption algorithms. IPSec mostly uses asymmetric algorithms for such specialized purposes as negotiating which keys to use for the symmetric encryption.

The ESP also provides for a secure VPN, a service called *tunneling*. Tunneling takes the original IP packet header and encapsulates it within the ESP. Then it adds to the packet a new IP header containing the address of a gateway (a type of secure VPN equipment). Tunneling allows to pass illegal IP addresses through a public network (like the Internet) that otherwise wouldn't accept them. Tunneling with ESP also has the advantage of hiding the original source and destination addresses from users on the public network, defeating or at least reducing the power of traffic analysis attacks. A traffic analysis attack uses network monitoring to determine whos saying how much to whom. An attacker doing traffic analysis doesn't know what is being said, but does know only how much is being said.

The ESP authentication field, an optional field in the ESP, contains something called an *Integrity Check Value (ICV)*: essentially a digital signature, computed over the remaining part of the ESP (minus the authentication field itself). It varies in length depending on the authentication algorithm used. It may also be omitted entirely, if authentication services are not selected for the ESP.

The authentication is calculated on the ESP packet once encryption is complete. The ICV supports symmetric (classical) authentication schemes. The source encrypts a hash of the data payload (or just composes a keyed

hash of the payload), and attaches this as the authentication field. The recipient confirms there has been no tampering and that the payload did come from the expected source by checking the authentication field. The current IPSec standard specifies **HMAC** (a symmetric signature scheme) with hashes SHA-1 and **MD5** as mandatory algorithms for doing authentication.

DES and 3DES

The *Data Encryption Standard* (**DES**) uses 56-bit symmetric keys to encrypt data in 64-bit blocks.

The 56-bit key provides 72,057,594,037,927,900 possible combinations. This sounds impressive, and it would take up to 20 years for typical business computers to run this many combinations. But, more focused, well-funded hacker organizations with a bigger inventory of powerful computers could break it in about 12 seconds. DES has been developed even further with its **3DES** ("triple-DES") system that encrypts information multiple times. For example, with 3DES, the data is encrypted once using a 56-bit key. The resulting cipher-text is then decrypted using a second 56-bit key. This results in clear-text that doesn't look anything like what was originally encrypted. Finally, the data is re-encrypted using a third 56-bit key. This technique of *encrypting, decrypting, and encrypting* (**EDE**) increases the key length from 56 bits to 168 bits.

4.3.2 AH

The IPSec suite's second protocol, the Authentication Header, provides authentication services but does not address confidentiality. The AH may be applied alone, in concert with the ESP, or in a nested fashion when using tunnel mode. Authentication provided by the AH differs from that provided in the ESP in that the ESPs authentication services do not protect the IP header that precedes the ESP. The AH services protect this external IP header, along with the entire contents of the ESP packet. The AH does not protect all of the fields in the external IP header because some of them change in transit, and the sender cannot predict how they might change. The standard is designed so AH works around these fields. In the packet, the AH goes after the IP header but before the ESP (if present) or other higher level protocol (like TCP, in the absence of ESP).

The parts are as follows:

- the *Next header* field indicates what the higher level protocol following the AH is (ESP or TCP, for example)

- the *Payload length* field is an 8-bit field specifying the size of the AH, in 32 bit words (groups of 4 bytes)
- the *Reserved* field is reserved for future use and is currently always set to zero
- the *SPI*, as in the ESP packet, identifies a set of security parameters (algorithms and keys) to use for this connection
- the *Sequence number*, also as in the ESP packet, is a number that increases for each packet sent with a given SPI, for the purposes of keeping track of the order the packets go in, and to make sure that the same set of parameters is not used for too many packets
- finally, the *Authentication data* is the actual ICV, or digital signature, for the packet. It's much the same as the ICV used in the ESP authentication field. It may include padding to bring the length of the header to an integral multiple of 32 bits (in IPv4) or 64 bits (IPv6).

Figure 4.4 shows the AH format:

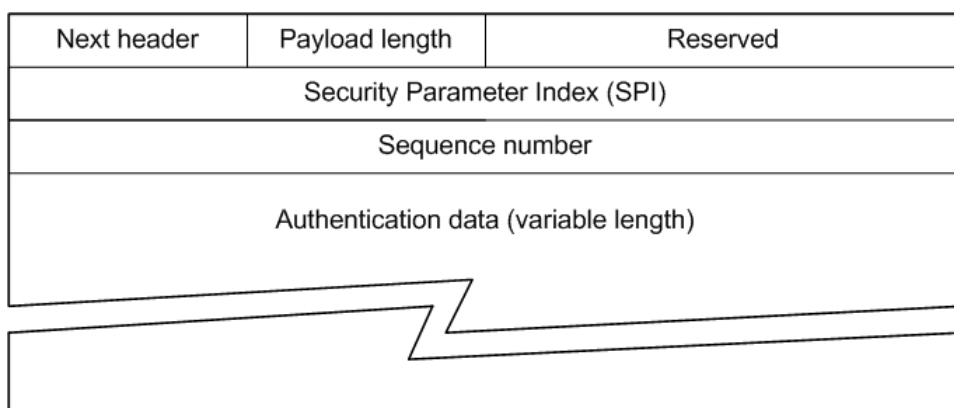


Figure 4.4: The Authentication Header format

Like the ESP, the AH can be used to implement tunneling mode. Also as in the ESP, IPsec requires specific algorithms to be available for implementing the AH. These are the minimum any IPsec implementation must support to guarantee minimal interoperability. All IPsec implementations, under the standard, must support at least HMAC-MD5 and HMAC-SHA-1 (the HMAC symmetric authentication scheme supported by MD5 or SHA-1 hashes) for the AH.

4.3.3 AH and ESP

The AH and ESP protocols are the building blocks of IPSec. The two protocols provide the pieces that are needed to build a secure VPN. They provide the fundamental services: confidentiality, authentication, and integrity.

The encryption services provided by the AH and ESP represent a powerful technology for keeping data secret, for verifying its origin, and for protecting it from undetected tampering. But they mean little without a wisely-designed infrastructure to support them, to distribute keys, and to negotiate protocols between communicating parties. Proposed secure VPN systems succeed or fail based on the strength of this infrastructure and its scalability. And that's where IPSec's IKE protocol suite comes in.

4.4 Protocol Negotiation and Key Management

To communicate with someone using authentication and encryption services (like those provided by IPSec's ESP and AH), you need to do three things:

1. negotiate with other people the protocols, encryption algorithms, and keys, to use
2. exchange keys easily (this might include changing them often)
3. keep track of all these agreements.

IPSec provides mechanisms to do all three.

4.4.1 The Security Association - SA

The first problem IPSec's designers solved was actually number three, how to keep track of all the details, and which keys and which algorithms to use. They did it by bundling everything together in something called the *Security Association (SA)*. An SA groups together all the things you need to know about how you communicate securely with someone else.

A security association is identified by:

- *security parameter index (SPI)*: a key that identifies a set of parameters for this association, stored in the Security Association Database
- *destination address* (currently only unicast addresses)
- *security protocol identifier*: AH or ESP

The SA, under IPSec, specifies:

- the mode of the authentication algorithm used in the AH and the keys to that authentication algorithm
- the ESP encryption algorithm mode and the keys to that encryption algorithm
- the presence and size of (or absence of) any cryptographic synchronization to be used in that encryption algorithm
- how you authenticate your communications (using what protocol, what encrypting algorithm, and what key)
- how you make your communications private (again, what algorithm, and what key)
- how often those keys are to be changed
- the authentication algorithm, mode, and transform for use in ESP plus the keys to be used by that algorithm
- the key lifetimes
- the lifetime of the SA itself
- the SA source address
- a sensitivity level descriptor.

The SA can be considered as a secure channel through the public network to a certain person, group of people, or network resource. It is like a contract with whomever is at the other end. The SA also has the advantage that it lets construct classes of security channels.

4.4.2 Security Parameter Index - SPI

The SA is a concept. The SPI is more concrete. The SPI is a number that uniquely identifies an SA. The SPI, together with the SA concept, makes keeping track of keys and protocols easy and automatic.

The SPI is an arbitrary 32-bit number the system picks to represent that SA whenever someone negotiates an SA. The SPI can not be encrypted in the packet because it is used to keep track of how to decrypt the packet.

It works like this. When negotiating an SA, the recipient node assigns an SPI it isn't already using, and preferably one it hasn't used in a while. It then

communicates this SPI to the node with which it negotiated the SA. From then until that SA expires, whenever that node wishes to communicate using that SA, it uses that SPI to specify it. The node, on receipt, would look at the SPI to determine which SA it needs to use. Then it authenticates and/or decrypts the packet according to the rules of that SA, using the agreed-upon keys and algorithms to verify (depending on the terms of the SA) that the data really did come from the node it claims, that the data has not been modified, and that no one between those nodes read the data.

4.4.3 Internet Key Exchange - IKE

IKE, the IPSec groups answer to protocol negotiation and key exchange through the Internet, is actually a hybrid protocol. It integrates the *Internet Security Association and Key Management Protocol (ISAKMP)* with a subset of the Oakley key exchange scheme.

IKE provides a way to:

- agree on which protocols, algorithms, and keys to use (negotiation services)
- ensure from the beginning of the exchange that you are talking to whom you think you are talking to (primary authentication services)
- manage those keys after they have been agreed upon (key management)
- exchange material for generating those keys safely.

Key exchange is a closely related service to SA management. When it's needed to create an SA, it's needed to exchange keys. So IKE wraps them both up together, and delivers them as an integrated package.

There is one other way to exchange keys. IPSec specifies that compliant systems support manual keying as well. That means if you wish to use manual (face-to-face) key exchange for certain situations, you still can. But IPSec's designers also assume that in most situations, for most large enterprises, this would be impractical.

Overview of Oakley

Oakley is a refinement of Diffie-Hellman key exchange scheme. Diffie-Hellman has the following problems that Oakley solves:

- no authentication
- vulnerable to man-in-the-middle attack

- computationally intensive, therefore vulnerable to *Denial of Service Attack* (**DoS Attack**)

Oakley uses the following features to counter these weaknesses:

- *cookies* to counter clogging attacks
- *groups* global parameters of the Diffie-Hellman key exchange (q and α). Oakley allows parties to negotiate groups.
- *nonces* to protect against replay attacks
- *authenticated Diffie-Hellman* to protect against man-in-the-middle attack. Authentication is based on either:
 - digital signature (with public/private keys)
 - public-key encryption
 - symmetric-key encryption

Overview of ISAKMP

ISAKMP defines procedures and packet format (i.e., a protocol) to establish, negotiate, modify, and delete security associations.

In essence, ISAKMP defines a header that introduces the following types of payloads:

proposal payload used for SA negotiation

- SA protocol (AH or ESP)
- sender's SPI
- a list of *transform payloads*

transform payload identifies a cryptographic function for a specific protocol function (e.g., 3DES for ESP), and its parameters.

key exchange payload Oakley, or Diffie-Hellman, or RSA-based key exchange, etc.

identification payload identifies the two ISAKMP peers

certificate payload passes a public-key certificate

hash payload verification (authentication) hash computed over some state information of this running ISAKMP

nonce payload random data used during the exchange

notification payload either error or status information associated with the current SA or with the SA negotiation

delete payload identifies one or more SAs that the sender has deleted from its database

IKE Phases

IKE functions in two phases.

Phase 1 is where the two ISAKMP peers establish a secure, authenticated channel with which to communicate. This is called the ISAKMP Security Association (not to be confused with the SAs that the protocol is trying to establish).

Phase 2 is where Security Associations are negotiated on behalf of services such as IPsec or any other service which needs key material and/or parameter negotiation.

IKE Modes

Oakley provides three modes of exchanging keying information and setting up SAs: two for IKE phase 1 exchanges, and one for phase 2 exchanges.

Main mode accomplishes a phase 1 IKE exchange by establishing a secure channel.

Aggressive mode is another way of accomplishing a phase 1 exchange: it is a little simpler and a little faster than main mode, but does not provide identity protection for the negotiating nodes, as they must transmit their identities before having negotiated a secure channel through which to do so.

Quick mode accomplishes a phase 2 exchange by negotiating an SA for general purpose communications.

Establishing a secure channel for negotiation

To establish an IKE SA, the initiating node proposes six things:

- encryption algorithms (to protect data)

- hash algorithms (to reduce data for signing)
- an authentication method (for signing data)
- information about a group over which to do a Diffie-Hellman exchange
- a *Pseudo-Random Function* (**PRF**) used to hash certain values during the key exchange for verification purposes (this is optional, you can also just use the hash algorithm)
- the type of protection to use (ESP or AH).

Diffie-Hellman key exchange scheme

One of the most annoying things about passing encrypted data around a public network is the number of opportunities an attacker has to get hold of encrypted material. It is possible to reduce the risk of their ever deciphering it by using larger and larger keys. But the larger the key, the slower and more complex the encryption and this can impair network performance.

A good compromise solution is to use reasonably large keys, and to keep changing them. But this presents difficulties too.

What is needed, is a method of generating a new key that is in no way dependent on the value of the current key. So if someone gets hold of the current key, it only gives them a small part of the overall picture, and they would have to break yet another entirely unrelated key to get the next part. Cryptographers call this concept "perfect forward secrecy". IKE uses a scheme called **Diffie-Hellman** to do this.

A Diffie-Hellman exchange works like this: two people independently and randomly generate values much like a public/private key pair. Each sends their public value to the other (using authentication to close out the man-in-the-middle). Each then combines the public key they received with the private key they just generated, using the Diffie-Hellman combination algorithm. The resulting value is the same on both sides, and therefore can be used for fast symmetric encryption by both parties. But no one else in the world can come up with the same value from the two public keys passed through the net, since the final value also depends on the private values, which remain secret.

The derived Diffie-Hellman key can be used either as a session key for subsequent exchanges, or to encrypt yet another randomly generated key, which can then be passed through the net quite safely. Note that authentication is needed to protect even Diffie-Hellman exchanges against the man-in-the-middle. Diffie-Hellman alone does not solve this problem. It would be

complicated, but without authentication a man-in-the-middle could use an active attack to get in on the action and plant his own keys. But if the key exchange mechanism is protected by an authentication scheme, Diffie-Hellman allows to generate new shared keys to use for symmetric encryption which are independent of older keys, providing perfect forward secrecy. And since symmetric encryption techniques are a lot faster, this can be quite useful in network communications. Before establishing a Diffie-Hellman exchange it is needed to agree on a few things. That is what the Diffie-Hellman parameter in the IKE SA is for. The parameter contains information on a group to perform the Diffie-Hellman exchange. The group consists of generation material used for coming up with keys.

1) Main mode

Main mode provides a mechanism for establishing the first phase IKE SA, which is used to negotiate future communications. The object here is to agree on enough things (authentication and confidentiality algorithms, hashes, and keys) to be able to communicate securely long enough to set up an SA for future communication. The steps in full will be:

1. use Main mode to bootstrap an IKE SA
2. use Quick mode within that IKE SA to negotiate a general SA
3. use that SA to communicate from now until it expires.

The first step, securing an IKE SA using Main mode, occurs in three two-way exchanges between the SA initiator and the recipient (see figure 4.5). In the first exchange (1 and 2 in the figure), the two agree on basic algorithms and hashes. In the second (3 and 4 in the figure), they exchange public keys for a Diffie-Hellman exchange, and pass each other nonces random numbers the other party must sign and return to prove their identify. In the third (5 and 6 in the figure), they verify those identities.

So the following is how IKE establishes its own IKE SA, step by step, using Main mode, and established digital signatures for authentication.

Each of the pieces is carried in its own payload, but it is possible to pack any number of these payloads into a single IKE packet. The parties actually use the generated shared Diffie-Hellman value in three permutations, once they derive it. Both parties have to hash it three times, generating first a derivation key (to be used later for generating additional keys in Quick mode), then an authentication key (for authentication), and then, finally the encryption key to be used for the IKE SA.

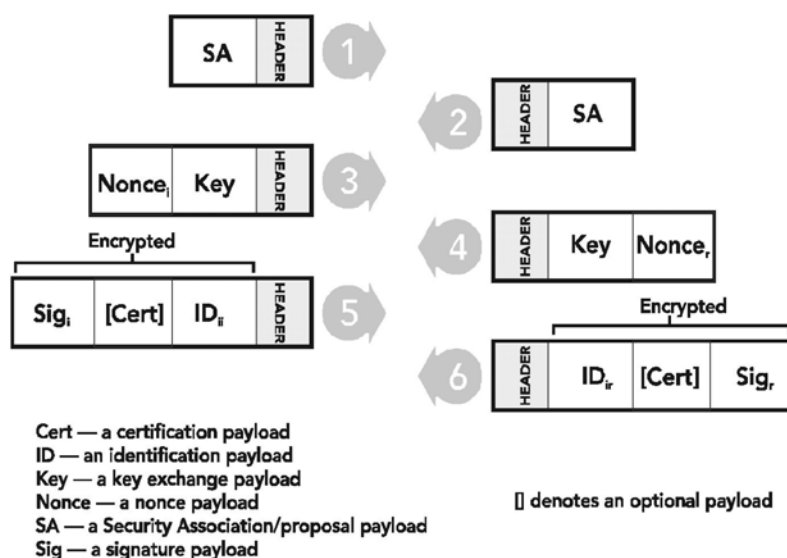


Figure 4.5: IKE Main mode

2) Aggressive mode

Aggressive mode provides the same services as Main mode. It establishes the original IKE SA. It looks much the same as Main mode except that it is accomplished in two exchanges, rather than three, with only one round trip, and a total of three packets rather than six.

In Aggressive mode, the proposing party generates a Diffie-Hellman pair at the beginning of the exchange, and does as much as is practical with that first packet, proposing an SA, passing the Diffie-Hellman public value, sending a nonce for the other party to sign, and sending an ID packet which the responder can use to check their identity with a third party. The responder then sends back everything needed to complete the exchange — really an amalgamation of all three response steps in Main mode, and all that is left for the initiator to do is to confirm the exchange (see figure 4.6).

The end result is that an Aggressive mode exchange attains the same goal as a Main mode exchange, except that Aggressive mode does not provide identity protection for the communicating parties. That is to say, in Aggressive mode, the parties exchange identification information prior to establishing a secure SA in which to encrypt it. So someone monitoring an aggressive exchange can actually identify who has just formed a new SA. The advantage of Aggressive mode, however, is speed.

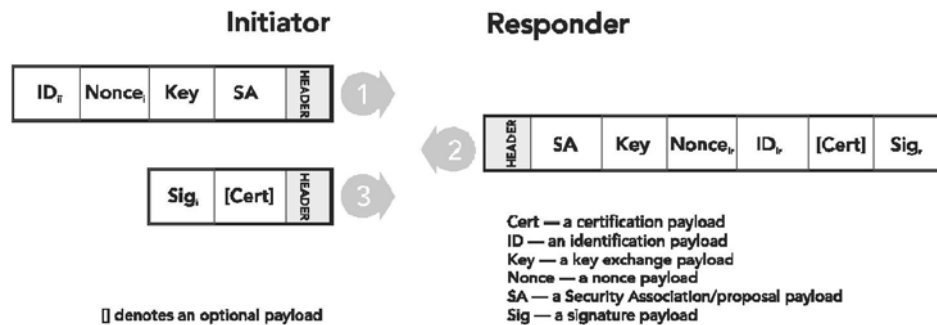


Figure 4.6: IKE Aggressive mode

3) Quick mode

Once two communicating parties have established an IKE SA using Aggressive mode or Main mode, they can use Quick mode. Quick mode has two purposes: negotiating general IPSec services, and generating fresh keying material. Quick mode is less complex than either Main or Aggressive mode. Since it's already inside a secure tunnel (every packet is encrypted), it can also afford to be a little more flexible.

Quick mode packets are always encrypted, and always start with a hash payload. The hash payload is composed using the agreed-upon PRF and the derived authentication key for the IKE SA. The hash payload is used to authenticate the rest of the packet. Quick mode defines which parts of the packet are included in the hash. Key refreshing can be done one of two ways. If you don't want or need perfect forward secrecy, Quick mode can just refresh the keying material already generated (in Main or Aggressive mode) with additional hashing. The two communicating parties can exchange nonces through the secure channel, and use these to hash the existing keys.

If it is needed a perfect forward secrecy, its still possible to request an additional Diffie-Hellman exchange through the existing SA and change the keys that way. Basic Quick mode is a three packet exchange, like Aggressive mode.

If the parties do not require perfect forward secrecy, the initiator sends a packet with the Quick mode hash, with proposals and a nonce. The respondent then replies with a similar packet, but generating their own nonce and including the initiators nonce in the Quick mode hash for confirmation. The initiator then sends back a confirming Quick mode hash of both nonces, completing the exchange. Finally, both parties perform a hash of a concatenation of the nonces, the SPI, and the protocol values from the ISAKMP

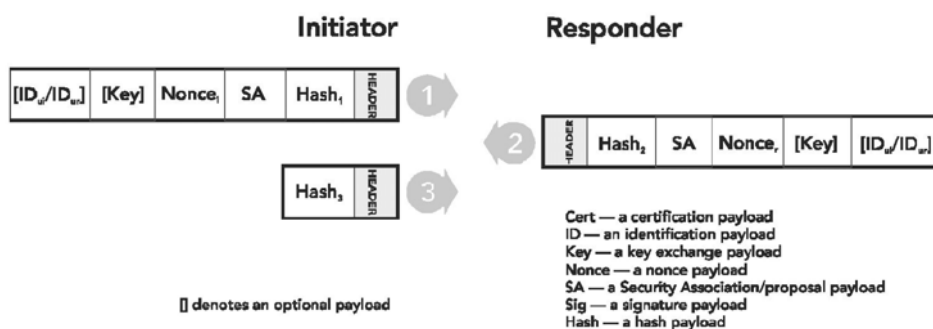


Figure 4.7: IKE Quick mode

header that initiated the exchange, using the derivation key as the key for the hash. The resulting hash becomes the new password for that SA. If the parties do require perfect forward secrecy, the initiator first generates a public/private key pair, and sends the public key along with the initiation packet (along with the hash and nonce). The recipient then responds with their own public key and nonce, and both parties then generate the shared key using a Diffie-Hellman exchange, again fully protected by the Quick mode hashes, and by full encryption within the IKE SA.

4.4.4 Negotiating the SA

After all those exchanges to generate the IKE SA, actually establishing the general purpose SA is relatively simple.

To generate a new SA, the initiator sends a Quick mode message, protected by the IKE SA, requesting the new SA. A single SA negotiation actually results in two SAs, one inbound, to the initiator, and one outbound. Each IPSec SA is one way and the node on the receiving end of that SA always chooses its own SPI to ensure it is the only SA using that reference. So, using Quick mode, the initiator tells the respondent which SPI to use in future communications with it, and the respondent follows up with its own selected SPI.

Each SPI, in concert with the destination IP address and the protocol, uniquely identifies a single, IPSec SA. However, these SAs are always formed in pairs (inbound and outbound), and these pairs have identical parameters (keys, authentication and encryption algorithms, and hashes), apart from the SPI itself.

4.5 IPSec Processing

Essentially, all IPSec processing is the enforcement of a security policy to ensure IP communication between to end points is secure. The security policy defines the security services to be applied at the IPSec endpoint, and every IP packet processed has to be evaluated against the policy regardless of whether it is protected by IPSec or not. Security policies are maintained in a *Security Policy Database (SPD)*. IPSec architecture specifies that a separate SPD be maintained for every IPSec enabled interface. Two tables are defined in the SPD for inbound and outbound policy. RFC 2401 mandates that an administrative user interface be implemented to manage the SPD.

Entries in the SPD are somewhat similar to a firewall rule set. Each entry has to indicate how the traffic that matches that entry is processed, and one of three actions need to be specified - bypass, reject or proceed with IPSec processing. Each policy entry also has a number of selectors that are used to identify and granulise the policy application process. These selectors include source address, destination address, user ID or system name, transport layer protocol and source and destination ports.

IPSec uses two completely independent processes in outbound and inbound IPSec traffic processing. Regardless of whether the packet has to be processed by IPSec or not, every packet has to be compared against the security policy database (SPD). Based on the policy defined for that packet the packet is dropped, routed or passed on to the IPSec kernel for processing.

4.5.1 Outbound Packet Processing

Once the routing decisions have been made, if the outbound packet has to be processed by IPSec, a number of selector fields in the packet are compared against the outbound policies in the SPD. More than one selector fields are used for this matching because there could be more than one policy defined for that packet. Policies in the SPD are ordered, and the first matching policy is selected for enforcement. The policy usually includes information on the SA (or SA bundle) for that packet. If there is no SA defined, key management mechanisms are used to generate an SA for that packet. If key management mechanisms are not present, then the packet is dropped. If there is an SA, or an SA can be generated, then the packet is processed by IPSec, using the parameters defined by the SA. This process is illustrated in figure 4.8 on page 80.

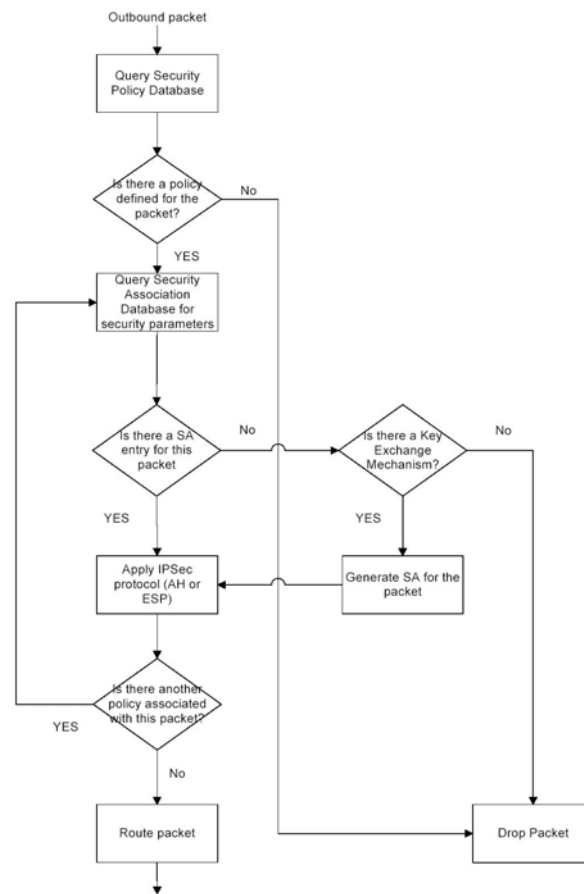


Figure 4.8: Outbound Packet Processing

4.5.2 Inbound Packet Processing

As indicated above, inbound packets are processed differently to outbound packets. Firstly any fragmented packets need to be reassembled. Then the IP next protocol field is checked for the AH or ESP values that indicate IPsec has been applied to that packet. If the packet is authenticated or encrypted using IPsec, it will have a SPI included in the AH or ESP header. Using the SPI, and the destination IP address, the SAD is queried to identify the appropriate SA for that packet. If there is no SA that can be matched to packet the packet is dropped. If there is a match, the packet is processed.

The first step in processing the packet is to match the selectors of the packet to the selectors in the SA. This process is repeated until all IPsec headers are processed, and the next header is a transport or an IP header. Then the SPD is queried using the back pointers from the SAs or the packets

selectors. Once match has been found, and the packet has been processed the packet is presented to the transport layer. Figure 4.9 illustrates how inbound packets are processed by IPSec.

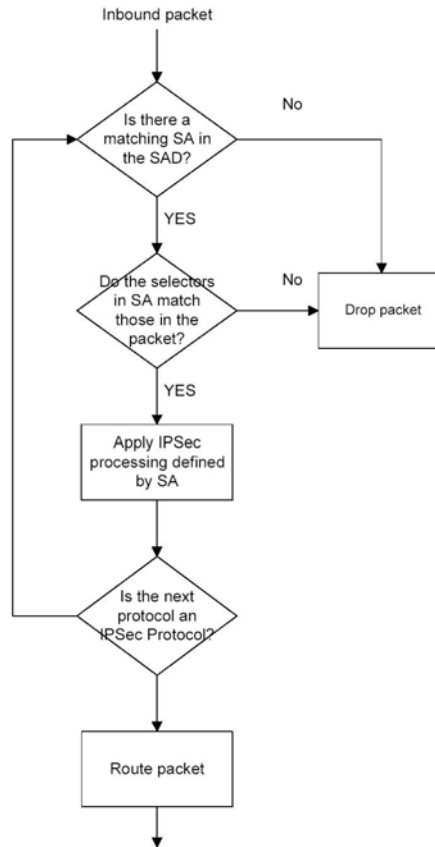


Figure 4.9: Inbound Packet Processing

4.6 IPSec Implementation

RFC 2401 defines three ways in which IPSec could be implemented.

1. Integrating IPSec in to the native IP implementation of the host or security gateway.
2. Inserting an IPSec layer between the IP layer and the network drivers. This implementation is known as the *bump-in-the-stack* (**BITS**) implementation.

- Using an external device such as an outboard crypto processor. This implementation is known as the *bump-in-the-wire* (**BITW**) implementation.

IPsec is implemented at the IP layer, thus providing security services to the upper layer protocols.

IPsec can be implemented between, two hosts, two gateways or between a host and a gateway.

Some examples of these of implementations are:

- Two servers synchronizing a database, either internally or across the Internet (figure 4.10).
- Two gateways, providing secure communication between the two networks connected by the two gateways (figure 4.11).
- A gateway and host/s as in remote access solutions (figure 4.12).

IPsec Between two hosts

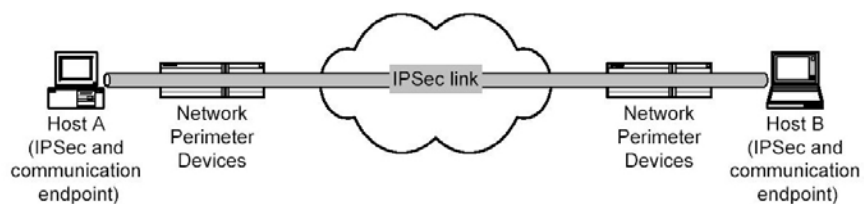


Figure 4.10: Host to host implementation of IPsec

IPsec between two gateways

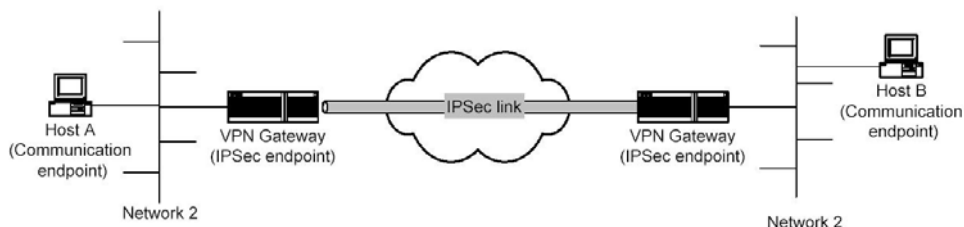


Figure 4.11: Gateway to gateway implementation of IPsec

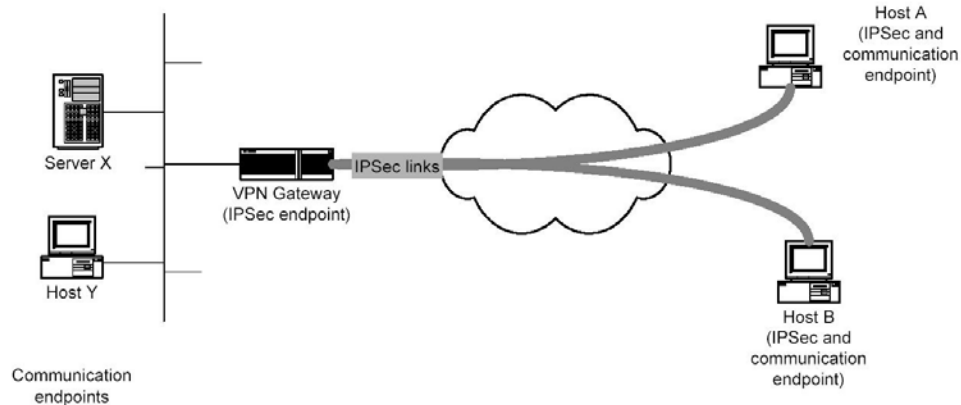
IPSec between a gateway and a host(s)

Figure 4.12: Host to gateway implementation of IPSec

4.7 Performance Consideration

Performance question should be carefully considered when deploying IPSec, because the encryption and decryption processes burn up a lot of CPU cycles.

IPSec has an impact on CPU load and network latency because its use will increase processor utilization while increasing IP traffic and IP packet sizes. Depending on the number of concurrent users and the choice of protocol and encryption algorithms employed, the CPU load could grow by as much as 90% when IPSec is used. A worst case scenario in terms of performance would involve using ESP with 3DES encryption with many clients.

There are two ways to address the performance hits to manage them to acceptable levels. One is to use extra processors to distribute the CPU load and the other is to offload the encryption/decryption process to network interface cards that can handle the computational stress.

4.7.1 The Impact of IPSec on WLAN Performance

Deploying IPSec over an unreliable and relatively low bandwidth medium incurs appreciable overheads and exposes IPSec hosts to DoS attacks.

The factors that most influence IPSec performance are packet size and available CPU power of the WLAN device. The overheads associated with packet size vary depending on the IPSec configuration. In tunnel mode, for example, an IP header is encapsulated by an outer IP header and an IPSec header, which can represent an appreciable additional overhead. In the case

of non-IP messages encapsulated by L2TP¹, the payload contains *Point-to-Point Protocol (PPP)* packets within the L2TP and *User Datagram Protocol (UDP)*, adding a further overhead to the IPSec packets.

The impact of CPU power on performance when using IPSec may be noticeable when using ESP. Informal tests have suggested that low-end processors (such as a Pentium 75) may struggle to meet the demands of ESP.

The use of AH is unlikely to have a measurably adverse affect on performance. However, it will help mitigate against DoS attacks that use rogue IPSec packets to consume CPU power, because it will detect and discard invalid packets much quicker than ESP can.

In summary, the following configurations are most affected by the use of ESP security:

- Low-powered or heavily loaded CPUs, due to IPSec processing overheads;
- Low bandwidth networks, due to IPSec bandwidth overheads;
- IPSec links carrying tunnelled or encapsulated traffic, due to the inefficiency of current standards;
- Links that rely on IP-based Quality of Service and related mechanisms (such as certain kinds of web caching and router-based congestion controls), due to the concealment by IPSec ESP of message contents.

4.8 Vulnerabilities Affecting IPSec-protected WLANs

One of the greatest vulnerabilities in WLAN technologies is due to the inherent broadcast nature of the communications medium. Due to this, it is important to realize that passive interception (*eavesdropping*) of WLAN communications is likely to be possible in the majority of scenarios, and possibly from a considerable distance. Additionally, the comparative ease with which an adversary can inject traffic onto a WLAN means that possibilities such as remote exploitation, message tampering and replay attacks, DoS and even social engineering attacks must be considered. Perhaps the greatest factor is that whilst traditional networks can rely on physical security measures, with the exception of radio frequency (RF) management, this is not possible with a WLAN.

¹later examined in section 4.10 on page 91

4.8.1 Confidentiality

The loss of confidentiality is a primary concern with intercept attacks, particular where mission-critical data is being transmitted. Although confidentiality is a core IPSec service, with IP traffic and dependent higher layer protocols such as email and file and print services being protected, it does not protect all network traffic. For instance, IPsec was not designed to, and therefore does not, protect data link layer traffic or non-IP network layer traffic.

In the 802.11 protocol, such information includes SSIDs, MAC addresses of transmitters and receivers, BSSIDs and the various informational components contained within probe responses and beacon frames. At the data link layer, this includes *Address Resolution Protocol (ARP)* and the *Dynamic Host Configuration Protocol (DHCP)*. Nor does IPSec provide complete protection of IP headers; in transport mode the original IP headers are unencrypted and in tunnel mode in a WLAN environment the outer IP header may be a near-duplicate of the original IP header, depending on the IPSec configuration. This is likely to be the case where one host computer is communicating directly with another host computer and an IPSec gateway is not involved.

This effectively means that there may be little difference between tunnel and transport modes in a WLAN. Thus, the additional benefits which tunnel mode can provide over transport mode, such as hiding the details of internal network configurations, are not achievable in a WLAN environment.

An eavesdropper reading an IPSec packet can determine from the IP header the IP addresses² and MAC addresses³ of the sending and receiving hosts, the size of the enclosed data, the data's checksum, and various IP flags and options. Whilst the data is encrypted when using ESP and thus this does not represent a direct threat to the confidentiality of the data, it does reveal the existence and attributes of a relationship between two hosts that may assist an eavesdropper in characterizing the content. Due to the broadcast nature of WLANs, it is difficult to prevent an attacker from gaining access to this information.

The effect of an attacker being able to obtain this information is that they can effectively map out the logical structure of the network configuration. This is usually one of the first steps attackers perform when attempting to break in to any network.

IPSec cannot protect non-IP protocols that are not encapsulated or tunneled within IP, since IPSec is designed solely to protect IP traffic. Non-IP

²In tunnel mode, one or both of the IP addresses may belong to a gateway device.

³Note that the MAC addresses may additionally be read from the 802.11 headers.

transport protocols such as Novel IPX, NetBEUI and AppleTalk are therefore exposed to eavesdroppers. Unfortunately, there is no generic solution to this problem. The workaround recommended by Microsoft is to encapsulate non-IP traffic within the L2TP, which is further encapsulated within IPSec-protected IP packets.

In summary, there is no direct threat to the confidentiality of standard IP traffic when IPSec is deployed using ESP with strong authentication. However, it is important to note that broadcast and multicast IP traffic may not be protected by IPSec, depending on the IPSec implementation. Furthermore, the confidentiality of non-IP traffic is at stake unless it is suitably encapsulated within IPSec-protected IP packets. Running a VPN over a WLAN does not directly increase the risk of loss of confidentiality compared to a VPN running within a wired network. However, the wireless nature of the communications magnifies existing eavesdropping threats by aiding in the interception of traffic. WLANs also expose information to man-in-the-middle attacks due both to the lack of authentication of 802.11 management frames and the ease with which an attacker can inject traffic onto a WLAN.

4.8.2 Integrity

The use of authenticated IPSec mitigates the risk posed by these vulnerabilities when transmitting IP traffic because rogue packets will be identified and discarded. As discussed previously, non-IP traffic may be similarly protected if it is encapsulated within IPSec-protected IP packets, e.g. through the use of L2TP in conjunction with IPSec. However, traffic at lower levels, such as data link layer traffic and 802.11 management frames, remains vulnerable as it cannot be protected by IPSec.

4.8.3 Resistance to Denial of Service Attacks

WLANs are extremely vulnerable to DoS attacks. The greatest DoS threats are loss or degradation of network connectivity and power consumption attacks. These may be achieved through jamming of the RF signal, injecting traffic or by manipulating the 802.11 protocol. Motives for such attacks include:

- Undermining operational momentum;
- Forcing the intervention of skilled network personnel;
- Reducing user confidence in the networks effectiveness;

- Discouraging users from specific behavior, such as using secure encryption methods.

Packet injection attacks force targeted hosts to expend time, CPU resources and battery power handling traffic, and may utilize valid or bogus packets to achieve their aim. The use of IPSec provides attackers with additional opportunities for attack. Although ESP and AH authentication both provide strong protection against packet spoofing, useful time and power must still be consumed verifying the packet. The time lost due to handling bogus but ostensibly valid packets is limited to the duration required to verify the *Hash Message Authentication Code* (**HMAC**) used. A more determined attacker might therefore choose to inject spoofed IPSec key exchanges or signature payloads, forcing the target to perform a time-consuming modular exponentiation operation. The attack would be particularly effective with unauthenticated ESP (without using AH as well) since this mode has less DoS protection, and the resulting encryption operation would be slower than authentication only.

An attacker may also inject 802.11 management frames. Since these are unauthenticated they can be easily spoofed and are always honored by the receiving device. Probably the most damaging types are deauthentication and disassociation frames. It is possible to use these types of frames against either individual hosts or the entire WLAN to simulate low bandwidth or loss of network access.

In summary, 802.11 WLANs are extremely vulnerable to denial or degradation of service attacks. Attackers may inject arbitrary traffic in order to consume bandwidth, validlooking rogue IPSec traffic to consume host resources, or 802.11 protocol frames to deny or degrade service. With the exception of RF containment, there are no real countermeasures against hostile packet injection on WLANs.

4.8.4 Traffic Flow Analysis

The quantity of traffic flow intelligence that can be recovered from a network running IPSec is determined by the networks topology, the ease of physical access and the configuration of IPSec. Wired networks are comparatively hard to wiretap effectively. Switched architectures, filtering devices such as firewalls and routers, and the need for either physical access or remote penetration all present significant barriers to eavesdroppers. Interception of WLAN traffic, on the other hand, requires only adequate range and line of sight. It should also be noted that the range required to allow analysis of traffic might be significantly greater than the usual operating limits of an

access point.

A successful intercept provides basic intelligence such as timing information, rates of flow, packet sizes and MAC addresses. However, the IP headers are a potentially richer source, which must be transmitted unencrypted in order to allow non-IPsec networks to perform routing functions. When IPsec encapsulates or transports data, it creates and modifies a range of so-called "mutable" fields: *Type-Of-Service (TOS)*, flags, fragment offset, *Time-To-Live (TTL)*, header checksum and options. These fields reveal clues about both the type of traffic IPsec is transporting and the underlying topology. For example, on a bandwidth-constrained WLAN, the TOS field may be used to reflect traffic precedence and handling criteria, such as "maximum security", "maximum reliability" or "minimum delay".

The TTL field may also be used by an eavesdropper to deduce the hop count⁴ of intercepted traffic, and from this the approximate size of the infrastructure involved and whether it is likely to be private or public. A TTL that appears to be close to its original value may help to infer additional details of the targets topology. Thus, the TTL field can provide an eavesdropper with information about the network size and topology and the operating systems in use, even when IPsec is being used.

The manner in which IPsec is used also potentially reveals basic traffic flow intelligence. For example, when not using compression ESP will produce packets with fixed size headers, allowing an observer to identify smaller packet types by size. To counteract this, ESP allows padding to be added to packets. However, this is of limited practical use because the padding is constrained to an unrealistic 255 bytes. If traffic flow security is an important concern, network administrators can enable this feature and configure the wireless networks *Maximum Transmission Unit (MTU)* size to match the small frames. Unfortunately, this would incur a significant performance penalty due to the overheads associated with medium contention (CSMA/CA in the case of 802.11) and frame acknowledgements.

An eavesdropper may be able to use the unencrypted data sent by IKE, the default protocol responsible for security associations, to identify the security algorithms in use. This may show that different hosts are using different security configurations and thus may be operating at different security levels. Thus, it could indicate to an attacker which hosts may be less well protected. Ensuring that all IPsec devices use the same security algorithm and settings,

⁴Although *Internet Assigned Numbers Authority (IANA)* states that the TTL should be initialized to 64, different implementations use different values. Additionally, different types of traffic should use different values (such as 64 for TCP and 1 for IGMP.) It is thus feasible to deduce the initialization value by inspecting traffic, since different platforms and traffic types are often not hard to distinguish.

thus avoiding any differentiation, can mitigate this. However, as described above, analysis of the amount of traffic and the IP headers may provide similar information to an attacker.

IPSec offers some basic mechanisms to help defeat traffic analysis. For example, an administrator may configure IPSec to use a shorter maximum packet size. The size would correspond approximately with another class of traffic used on the WLAN that has a shorter than normal packet size. The effect would be to blend the appearance of the two classes of traffic with respect to packet size. However, a potential disadvantage of this approach is that smaller packet sizes would significantly reduce throughput.

In summary, even when IPSec is being used, WLANs significantly ease the task of traffic analysis in most circumstances. The main vulnerabilities are:

- Ease and safety of interception. Whereas wired networks require a physical wiretap or remote compromise, WLAN interception can be conducted passively from a distance;
- Broadcast monitoring. A wiretap or remote compromise does not guarantee access to all network traffic. WLAN intercepts provide all traffic from stations within range;
- Useful clues within IP headers. IP headers are not comprehensively protected by IPSec, and provide a range of potential intelligence;
- Ability to inject arbitrary frames. This could be used, for example, to force renewal of IPSec negotiation for DoS purposes or to determine the security algorithms in use.

The mitigating factors are:

- Basic countermeasures against analysis of interframe timing, packet size, SPI index and topology, such as adjustment of TTL and fragmentation threshold. However, these are crude and inefficient measures;
- Unstable medium. Interframe timing and latency analysis is inherently less reliable. Retries and corruption more likely;
- Limited bandwidth. Higher probability of increased latency and retries.

4.9 Conclusions on IPSec

A number of potential vulnerabilities remain when using the IPSec standard to protect a WLAN based upon the IEEE 802.11 standard. Due to this

risks, we believe at this time that these vulnerabilities can only be mitigated through careful network administration and accurate topology design.

It should also be noted that much depends upon the IPSec implementation used.

In addition to normal security "best practice", a number of other steps should be performed when using IPSec in a WLAN environment. These are summarized below:

- Where WLANs are deployed, wireless intrusion detection or health monitoring systems should be used to help detect and identify attacks;
- To mitigate the risk posed by layer 2 attacks, unwanted layer 2 traffic should be filtered between the wired and wireless network segments. This can be achieved with a number of mechanisms, including:
 - access point-based packet filtering;
 - switch-based packet filtering;
 - bridging firewall;
 - router-based access points;
 - IPSec gateway that additionally filters packets.
- An appropriately configured "personal" firewall should be installed on each device using IPSec in the WLAN environment. This should help to ensure that traffic, such as broadcasts, cannot be used to attack the device;
- WEP should be used as a complement to an IPSec VPN. It will secure data transmissions in order to prevent trivial lower-level traffic injection attacks via the WLAN. WEP authentication should be disabled, since it is trivial for eavesdroppers to recover usable keystreams from intercepted authentication exchanges. It is also important to note that WEP has significant key management issues and is vulnerable to several active cryptographic and DoS attacks;
- To hinder traffic analysis, significant network resources, such as servers, should only be deployed on the wired network segment and made accessible only via an IPSec gateway so that their network addresses are concealed. To mitigate against characterization of stations as belonging to important users, it is recommended that mechanisms which mark traffic according to priority (such as quality of service) be avoided;

- IPSec should always be deployed with authenticated ESP and AH as well as strong, mutual authentication in order to protect the confidentiality and integrity of network traffic. Non-IP traffic should be encapsulated within similarly protect IP packets, e.g. through the use of L2TP in conjunction with IPSec;
- Communications within the WLAN should use transport mode. There is no security benefit to be gained from using tunnel mode within a WLAN environment, and its use will increase bandwidth overheads. However, if two wired networks are connected over a wireless segment, it is better to use tunnel mode;
- Consideration should be given to taking steps to reduce the amount of RF energy from the WLAN that is visible to an attacker. These may include reducing power levels, using directional antennas, careful network planning and controlling a physical perimeter around the network.

4.10 Introduction on L2TP

The *Layer 2 Tunnel Protocol (L2TP)* is an emerging *Internet Engineering Task Force (IETF)* standard that combines the best features of two existing tunneling protocols: Cisco's *Layer 2 Forwarding (L2F)* and Microsoft's *Point-to-Point Tunneling Protocol (PPTP)*. L2TP is an extension to the *Point-to-Point Protocol (PPP)*, which is an important component for VPNs. VPNs allow users to connect to their corporate intranets or extranets.

The two main components that make up L2TP are the *L2TP Access Concentrator (LAC)*, which is the device that physically terminates a call and the *L2TP Network Server (LNS)*, which is the device that terminates and possibly authenticates the PPP stream.

Generally, a user connects to a *Network Access Server (NAS)* through ISDN, ADSL, dialup POTS or other service, runs PPP over that connection. In this configuration, the L2TP PPP session endpoints are both on the same NAS.

L2TP uses packet-switched network connections to make it possible for the endpoints to be located on different machines. The user has an L2TP connection to an access concentrator, which then tunnels individual PPP frames to the NAS, so that the packets can be processed separately from the location of the circuit termination. This means that the connection can terminate at a local circuit concentrator, eliminating possible long-distance

charges, among other benefits. From the user's point of view, there is no difference in the operation.

4.11 Overview on L2TP

Using L2TP tunneling, an *Internet Service Provider (ISP)*, or other access service, can create a virtual tunnel to link customer's remote sites or remote users with corporate home networks. The L2TP access concentrator (LAC) located at the ISP's *point of presence (POP)* exchanges PPP messages with remote users and communicates by way of L2TP requests and responses with the customer's L2TP network server (LNS) to set up tunnels. L2TP passes protocol-level packets through the virtual tunnel between end points of a point-to-point connection. Frames from remote users are accepted by the ISP's POP, stripped of any linked framing or transparency bytes, encapsulated in L2TP and forwarded over the appropriate tunnel. The customer's home gateway accepts these L2TP frames, strips the L2TP encapsulation, and processes the incoming frames for the appropriate interface. Figure 4.13 shows the L2TP tunnel detail and how user connects to the LNS to access the designated corporate intranet.

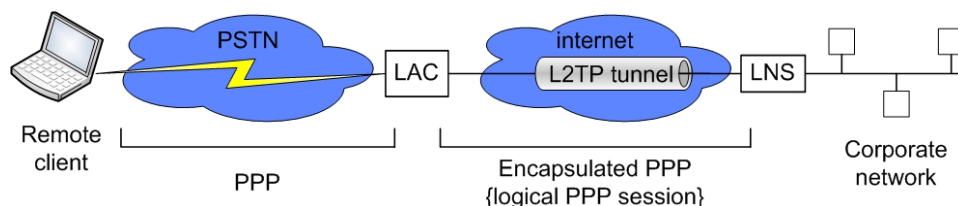


Figure 4.13: L2TP tunnel details

1. The client dials the ISP using an analog phone or ISDN. In this case, the client's computer is configured with PPP, although a client may also run L2TP directly.
2. When the call arrives at the ISP's LAC, the LAC performs a call check by contacting a RADIUS server. The RADIUS server responds with an accept or reject message. If accepted, the reply will also specify that an L2TP tunnel is needed.
3. The LAC creates a tunnel to the LNS at the client's corporate site. This is done by sending a message to UDP port 1701. An authentication procedure takes place between the LAC and the LNS.

4. A tunnel is set up and the client begins communicating with the corporate *Link Control Protocol (LCP)* using PPP. The client first sends PPP authentication information to the LCP, which in turn authenticates the client.

The LCP strips off the L2TP header to access the PPP frames. Note that the LAC does not authenticate the client during the set-up phase, but it does check with RADIUS to make sure that the dial-up session is allowed. The client is authenticated by the corporate server, just as if he or she logged on from a node directly attached to that network.

4.12 L2TP Frame Structure

The client's PPP frames are encapsulated into IP packets with an L2TP tunneling header and sent across the Internet connection.

The L2TP packet is carried within the UDP/IP datagram. L2TP uses the well-known UDP port 1701. The initiator of the tunnel picks up an available source UDP port on its own system and sends the request to this well known port. The acceptor of this request picks up a free port on its system as a source port and sends a response to the initiator's source port. Subsequent packets will be exchanged between these ports. It is also recommended that the UDP checksum be enabled for both control and payload packets.

The format of the L2TP packet is shown in figure 4.14:

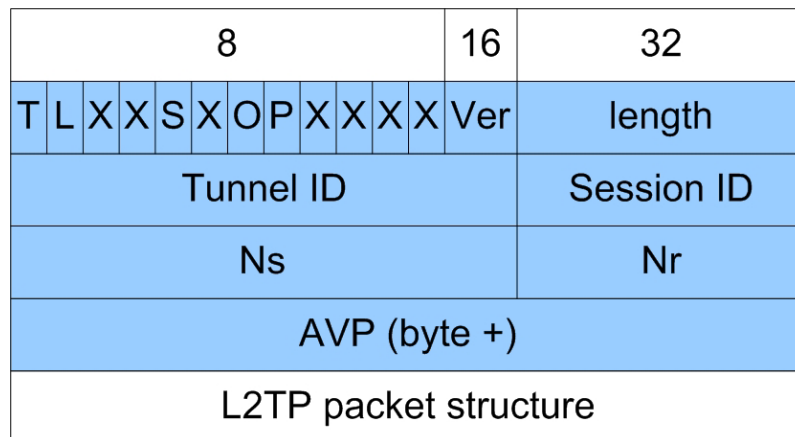


Figure 4.14: L2TP packet frame

T The T bit indicates the type of message. It is set to 0 for data messages and 1 for control messages.

- L** When set, this indicates that the Length field is present, indicating the total length of the received packet. Must be set for control messages.
- X** The X bits are reserved for future extensions. All reserved bits are set to 0 on outgoing messages and are ignored on incoming messages.
- S** If the S bit is set, both the Nr and Ns fields are present. S must be set for control messages.
- O** When set, this field indicates that the Offset Size field is present in payload messages. This bit is set to 0 for control messages.
- P** If the Priority (P) bit is 1, this data message receives preferential treatment in its local queuing and transmission. LCP echo requests used as a keepalive for the link, for instance, are generally sent with this bit set to 1. Without it, a temporary interval of local congestion could result in interference with keepalive messages and unnecessary loss of the link. This feature is only for use with data messages. The P bit has a value of 0 for all control messages.

Ver The value of the ver bit is always 002. This indicates a version 1 L2TP message.

Length Overall length of the message, including header, message type AVP, plus any additional AVP's associated with a given control message type.

Tunnel ID Identifies the tunnel to which a control message applies. If an Assigned Tunnel ID has not yet been received from the peer, Tunnel ID must be set to 0. Once an Assigned Tunnel ID is received, all further packets must be sent with Tunnel ID set to the indicated value.

Call ID Identifies the user session within a tunnel to which a control message applies. If a control message does not apply to a single user session within the tunnel (for instance, a Stop-Control-Connection-Notification message), Call ID must be set to 0.

Nr The sequence number expected in the next control message to be received.

Ns The sequence number for this data or control message.

Data messages have two additional fields before the AVP as shows in figure 4.15:

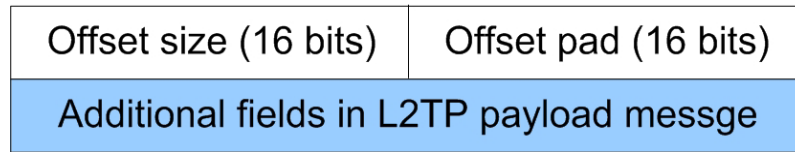


Figure 4.15: L2TP additional fields

Offset size This field specifies the number of bytes past the L2TP header at which the payload data is expected to start. It is recommended that data thus skipped be initialized to 0s. If the offset size is 0, or the O bit is not set, the first byte following the last byte of the L2TP header is the first byte of payload data.

AVP The AVP (Attribute-Value Pair) is a uniform method used for encoding message types and bodies throughout L2TP. The format of the AVP is given in figure 4.16:

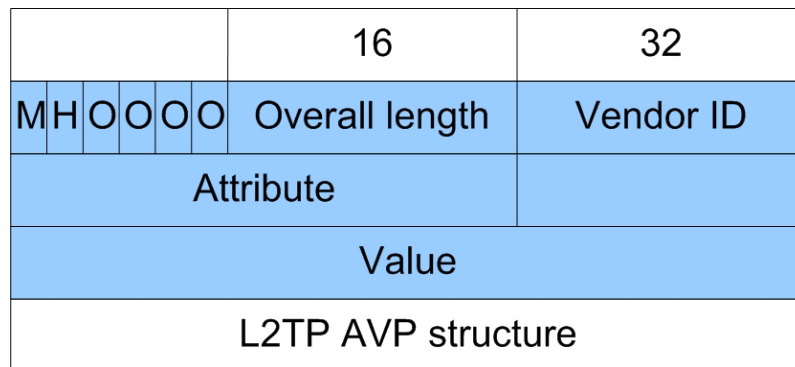


Figure 4.16: L2TP AVP structure

M The first six bits are a bit mask, describing the general attributes of the AVP. The M bit, known as the mandatory bit, controls the behavior required of an implementation which receives an AVP which it does not recognize.

H The hidden bit controls the hiding of the data in the value field of an AVP. This capability can be used to avoid the passing of sensitive data, such as user passwords, as cleartext in an AVP.

- Overall length** Encodes the number of octets (including the overall length field itself) contained in this AVP. It is 10 bits, permitting a maximum of 1024 bytes of data in a single AVP.
- Vendor ID** The IANA assigned SMI Network Management Private Enterprise Codes value, encoded in network byte order.
- Attribute** The actual attribute, a 16-bit value with a unique interpretation across all AVP's defined under a given Vendor ID.
- Value** The value field follows immediately after the Attribute field, and runs for the remaining octets indicated in the overall length (i.e., overall length minus six octets of header).

Chapter 5

PKI and Digital Certificates

5.1 Overview

A *Public Key Infrastructure* (**PKI**) is a system that is responsible for managing digital certificates and the keys that sign them. A PKI enables an organization to use encryption and digital signature services within a network and across applications.

To perform its services, components within the PKI rely heavily on the use of public key techniques. These techniques require a public and private key pair to perform encryption and decryption operations. The critical key in these operations is called the "root key".

It is critical to understand how the root key is used within the PKI and the importance of securing this root key.

The services that you require from a PKI depend on your organization's security requirements and the features offered by the PKI. A comprehensive PKI should be able to:

- generate, store, and revoke certificates
- back up keys and manage key histories
- update key pairs and certificates

As well, a PKI must encourage its users to make use of its services by providing client software that makes services transparent to users.

5.2 Understanding the Function of Components

The following diagram shows the major components of a PKI:

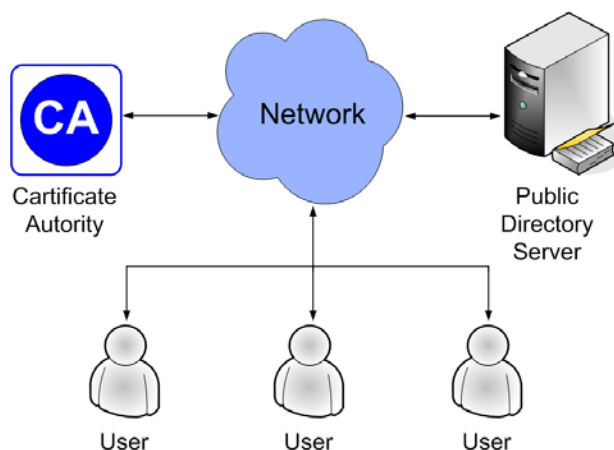


Figure 5.1: The PKI major component

The PKI components communicate over a network. For example, the PKI can use an Internet, an extranet, or an internal network.

The certificate authority is the core component of a PKI. This component issues, signs, revokes, and manages certificates for users.

The public directory server stores copies of certificates and *Certificate Revocation Lists* (CRLs).

Identifying users

A certificate is proof of a user's identity for electronic transactions. Certificates enable users to communicate with the assurance that they are who they claim to be, even if they do not have a personal relationship. The users trust each other because they share a common trust point, the certificate authority that issues the certificates.

A certificate works much like a travel passport. Before a passport is issued, an entity verifies the identity of the traveler. Countries that trust the certifying entity accept the passport as proof of the traveler's identity, even though the traveler is not known to these countries. The entity that issues the passport is a "trusted third party."

Verifying users

Within a PKI, the trusted third party is the certificate authority. The certificate authority verifies the identity of a user before issuing a certificate to the user.

After verifying the identity of a user, the certificate authority generates a certificate that includes

- its name as the issuer of the certificate
- the user's name
- the user's public key
- a description of the function of the key
- a date that indicates how long the certificate is valid

The certificate also includes the digital signature of the certificate authority. This signature guarantees that the identity of the certificate holder user has been verified by the certificate authority. Any user who trusts the signature of the certificate authority can also trust the identity of the certificate holder.

Processing a Request for a Certificate

Figure 5.2 explain how a certificate is requested.

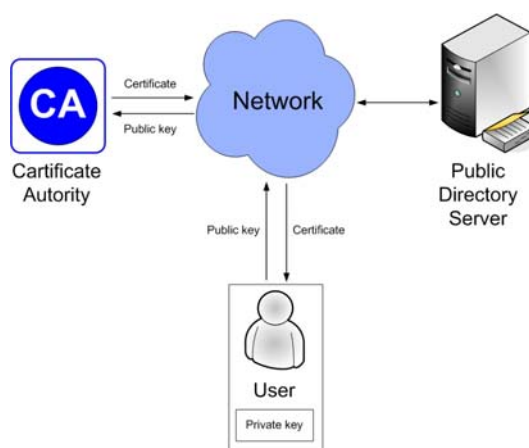


Figure 5.2: The PKI processing request

When a user requires a certificate, the user generates a public/private key pair. The user keeps the private key and sends the public key to the certificate authority.

After establishing the relationship between the public key and the user, the certificate authority generates a certificate, sends it to the user, and stores a copy of the certificate in the public directory server.

5.3 Recognizing the vulnerabilities

To perform its functions, a certificate authority relies heavily on the use of public key techniques, which require a public and private key pair to perform encryption and decryption operations. The critical key in these operations is called the "root key." This key is the private key of the certificate authority's public and private key pair. For example, the certificate authority uses its root key to sign certificates and CRLs, and to cross-certify other certificate authorities.

The security of the certificate authority's root key is extremely important. For example, assume that some certificates signed by the certificate authority are used to perform business transactions. If an intruder accesses the root key without detection, the intruder can create and sign certificates, and use them to falsely represent him or herself in business transactions.

If the root key is compromised, all certificates and CRLs ever signed by the certificate authority are called into question.

Protecting the root key using the highest possible security measures must be the most important concern. Unauthorized access to the root key is the equivalent of being able to print illegal travel passports with impunity.

5.4 Conclusions

Digital certificates deployment in an industrial environment is too complicated and expensive since it requires a set of software and hardware devices that makes even more complicated the network structure and management.

Moreover, since digital certificates need an accurate and constant management (every certificate has a limited life time: 1 year), if something goes wrong, the entire logistic activity can heavily be compromised.

Therefore, while in such an industrial environment it's not supposed to be constantly present an expert network manager who takes care of this critical system, **we strongly discourage the deployment of digital certificate** although they assure to the entire network a very high security level.

Chapter 6

Demo Brewery Network Design

6.1 Introduction

Demo Brewery is the typical example of the industrial environment for which we have to develop a wireless security solution.

The Demo site is situated in Russia and it is one of the biggest Breweries supported by Syskron. It involves all the difficulties typically related to an industrial environment adding also the problems due to the hostile weather (the temperature reaches -30°C). For this reason all the network equipment must be water proof and protected against all the atmospheric agents.

6.2 Network Equipment

Here is the list of the used hardware devices:

- **Access Point:** Orinoco AP-500 inserted in a special box to protect it against all the atmospheric agents (see figure 6.1).
- **Mobile PC:** DloG Industrial PC DNeT IPC 5/100 (Microsoft Windows 2000, AMD K6 400MHz, 128MB RAM) mounted on every forklift; it's a special industrial device with heating feature that allows this device to be operational at temperatures up to -30°C (see figure 6.2).
- **Bar-code reader:** DATALOGIC Dragon Mobile 433MHz radio system; it's an industrial hand-held laser scanner used to track small packages (beer glasses, promotion material, etc.) (see figure 6.3).
- **Wireless printer:** generic wireless printer that supports IEEE 802.11b standard.



Figure 6.1: The Access Point special box



Figure 6.2: The mobile PC mounted on an forklift



Figure 6.3: The bar-code reader

6.3 Network Operation

The wireless network is located inside the building as shown in figure 6.4.

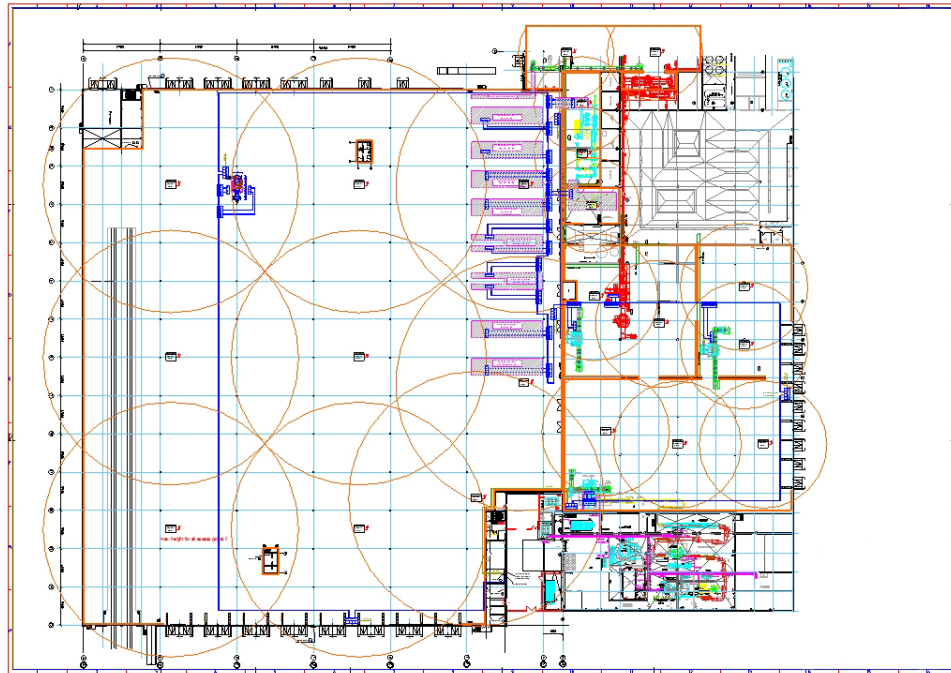


Figure 6.4: The planimetry

This wireless network is used to support all the stock exchange operations; it is attached to the wired central office network (the private side) where it's located the main database server. Every forklift is connected via wireless to the central database; querying this database, they receive the orders about where to load or unload the packages in the tracks or trains that are waiting for.

6.4 Network Design

The most important point in the network design development is to consider carefully the roaming problem just because the mobile stations are always in movement. To avoid interference, the channels selected should be properly spaced apart. Figure 6.5 illustrate how one can reuse a channel and still avoid interference. The client will seamlessly roam between these access points.

For the client to be able to roam seamlessly, it is necessary for the access points to:

- Be connected to the same IP subnet, so the client won't have to change IP address
- Have the same ESS ID to identify the wireless network
- Have the same WEP keys so that the client knows how to encrypt the data
- All the adjacent access points must have DIFFERENT channel frequencies to avoid interference.

If one or more of these requirements are not met, network communication for the client halt.

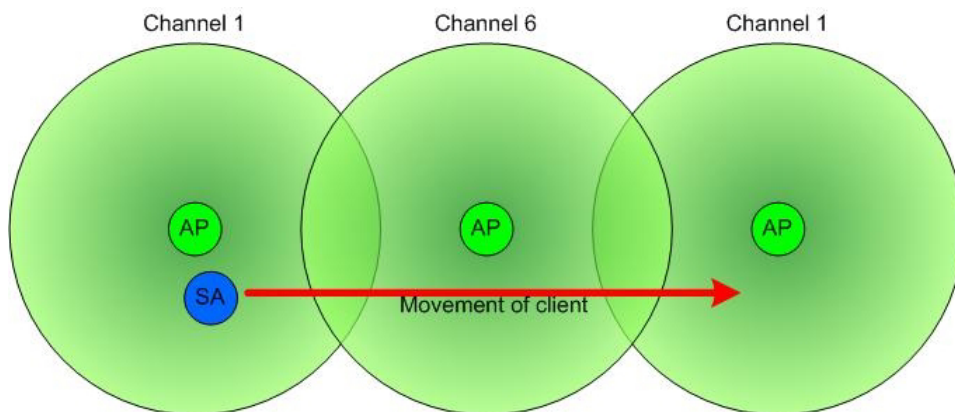


Figure 6.5: The Access Points roaming

6.4.1 Channel Frequencies Allocation

The distance between the center frequency of channel, when multiple channels are used, must be at least 22MHz (distance from center to center); in this way only side lobes may overlap as shown in figure 6.6.

Otherwise, with an insufficient channel separation, main lobes also overlap; so one channel senses the other channels transmission as Interference Noise as shown in figure 6.7.

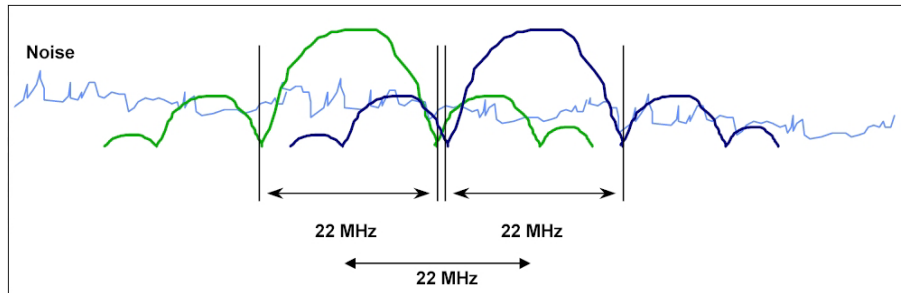


Figure 6.6: The optimal channel separation

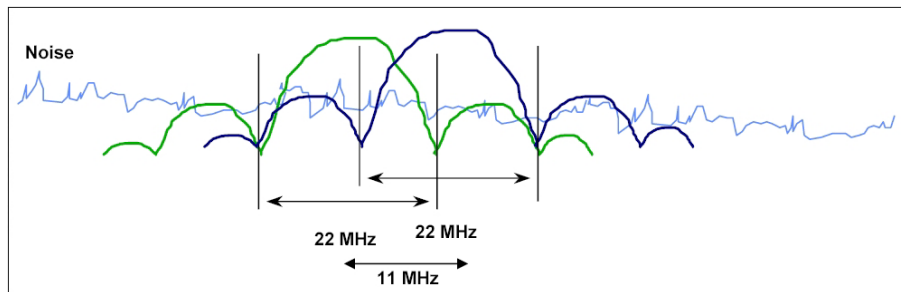


Figure 6.7: The insufficient channel separation

The available frequencies for the corresponding selectable sub-channels are:

Channel	Frequency
1	2412 MHz
2	2417 MHz
3	2422 MHz
4	2427 MHz
5	2432 MHz
6	2437 MHz
7	2342 MHz
8	2447 MHz
9	2452 MHz
10	2457 MHz
11	2462 MHz

Table 6.1: The radio characteristics for 2.4GHz Frequency Band

Upon these considerations, we decided to allocate the channels as shown in figure 6.8.

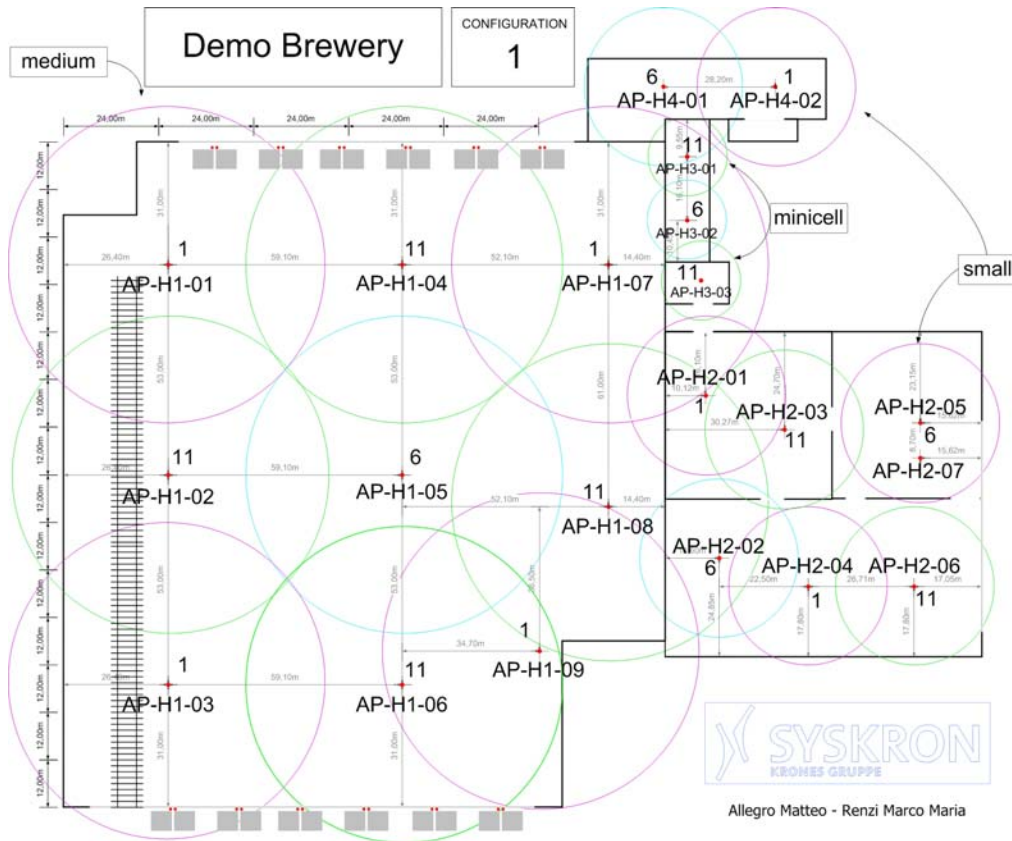


Figure 6.8: The Channel Frequencies Allocation plan

6.5 Security Implementation

Demo Ethernet Network was configured as follows:

Segment:	from 192.168.14.0 to 192.168.14.255
Subnet mask:	255.255.255.0
Gateway:	192.168.14.1
Domain:	Demo_WMS

The only security countermeasure was the WEP enabling; there was also no distinction between the wireless segment and the wired/private segment

located in the central office. In this way any malicious listener could collect information from the wireless network in order to gain access to the main database server.

For this reason, we thought first to divide the old network in two different segments with a gateway, and then to secure all the communication from the wireless side to the gateway with a VPN structure using IPSec tunnels. So that the overall data transmission through the gateway is secured and completely invisible for any listener. In fact the communication between two wireless devices is not allowed; the only possible connection is between a wireless station and the central database, through the gateway.

It's important to notice that the gateway can optionally act like a stateful firewall **only** for the non-tunneled incoming traffic; in fact the IPSec encapsulation prevents any data analysis before it is decrypted. In this way, in order to apply filtering rules to the IPSec secured traffic incoming from the clients located in the public network, it's necessary to use a second firewall inspection after the gateway decrypts IPSec data. **NB:** this firewall device must belong to the private network and we suggest to locate it between the gateway and the central database server as shown in figure 6.9.

The only devices not taking part to the VPN are the bar-code readers, the APs and the wireless printers; they affect the overall security as follows:

- the bar-code readers don't belong to the IP network since they communicate to the mobile PCs through a low power, license free radio in the 433 MHz band; this allows bi-directional communication between the base station and the host apart from the Wi-Fi standard.

For this reason, they don't add any additional security weakness to the public wireless network.

- the APs, by default, don't support IPSec but they can be crossed by VPN tunnels in a completely transparent way. So, when the administrator needs to access the APs for managing purposes, the established communication is not secured.

For this reason, they add relevant security weakness to the public wireless network only if the management session takes place from the public network via wireless; therefore it's strongly recommended that APs management is performed by the administrator from the private wired network only.

- The wireless-printers don't support IPSec; therefore they are supposed to receive print jobs from the print server in a clear not-secured communication.

For this reason, they add no security weakness to the public wireless network if the print server is located inside the private LAN, behind a gateway with the appropriate firewall rules that allow non-tunneled communication only from the private to the public side. Therefore an eventual intruder could gain access to the wireless printer but NOT to the private network.

In figure 6.9 is shown our concept about the WLAN security implementation with a VPN structure using IPsec tunnels.

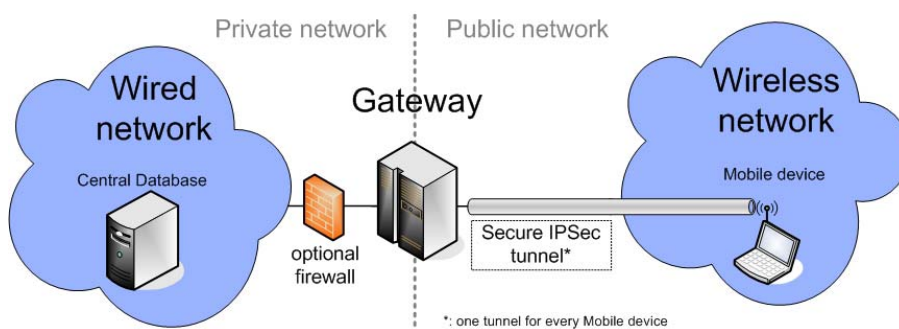


Figure 6.9: The security concept

6.6 Products Selection

In order to develop our security concept, we looked for a solution that could satisfy the following requirements:

- scalability
- reliability
- non-proprietary technology
- world wide support
- reasonable tradeoff between quality and price.

Scalability

Since the product we need must be employed in heterogeneous industrial environments and must follow eventual changes in the company needs, the

solution must be *fully scalable*; in this way we have a product that can fit small and big networks without the need to change it.

Our purpose is to find a "box" that can be easily upgraded (e.g. adding licences for new VPN tunnels or adding extra features) meeting the requirements of growing companies.

Reliability

Since this solution is going to be used in an industrial environment, it must be *reliable* in order to grant that the production won't stop for any eventual failure. Moreover, once configured and installed, it must work without any further technical fix.

Non-proprietary technology

Since it has to work with any network equipment, this product must use *standard technology*. In this way the company is not obliged to change all the existent network devices, saving time and money.

World wide support

Since this product has to be employed in different sites all over the world, it must be developed by a company that provides *world wide support*.

Reasonable tradeoff between quality and price

The product evaluation is based also on the comparison between quality and price; it must be found the *right tradeoff* in order to have a product that fit the company needs at a reasonable price.

6.6.1 Available products on the market

Now we'll have a look to the solutions we found on the market, with a brief description that summarize their main characteristics.

Reefedge

This company presents a suite of devices to manage, secure and monitor the entire wireless network activity; however Reefedge doesn't offer world wide support. For this reason we have discarded this solution.

Checkpoint

Checkpoint, world leader in firewall development, provides a firewall box (**Checkpoint Safe@Office 105**) capable of VPN tunneling. This box was born as a firewall with the extra feature of 3 VPN users every 22 firewall users. In order to satisfy our requirements (at least 25 VPN users), we'd have to choose the biggest model (**Checkpoint Safe@Office 225U**) that provides 30 VPN users every 150 firewall users at a very high price. For this reason we have discarded this solution.

Perfigo

This company presents a suite of devices to manage, secure and monitor the entire wireless network activity; however Perfigo doesn't offer world wide support. For this reason we have discarded this solution.

Bluesocket

This company presents a suite of devices to manage, secure and monitor the entire wireless network activity; however Bluesocket doesn't offer world wide support. For this reason we have discarded this solution.

Symbol

Even if Symbol is one of wireless technology leaders, at the moment only provides solution for network monitoring. This company promotes a VPN software which seems to be currently under development since no detailed information about it are available. After we asked for additional informations, we received an answer two months later. For this reason we have discarded this solution.

Cisco

It offers a complete proprietary solution. It works only with Cisco network devices and doesn't support wireless devices and access points from other vendors. For this reason we have discarded this solution.

Nortel Networks

Nortel Networks provides a complete solution (**Contivity Series VPN gateway**) that meets all our requirements. It's composed by the **Contivity 1010 VPN gateway** box in conjunction with the **Nortel VPN Client** software. For this reason we took this product into consideration.

AirFlow Networks

This company presents a suite of devices to manage, secure and monitor the entire wireless network activity; however AirFlow Networks doesn't offer world wide support. For this reason we have discarded this solution.

Vernier Networks

This company presents a suite of devices to manage, secure and monitor the entire wireless network activity; however Vernier Networks doesn't offer world wide support. For this reason we have discarded this solution.

Aruba

It offers a complete solution only for big networks (starting from 1000 users), providing Power Over Ethernet (POE). The eventual presence of fiber optics to ethernet converters doesn't allow the employment of POE; for this reason we have discarded this solution.

Nokia

Even though Nokia has a world wide support and develops non-proprietary technology, its products (Nokia gateways) are mainly firewalls with limited VPN features. Only the biggest and most expensive models, with hundreds of firewall users, provide the sufficient amount of VPN tunnels. For this reason we have discarded this solution.

NCP

NCP provides a complete software solution (NCP Secure Communications) that meets all our requirements. It's composed by the **NCP Secure Server** and the **NCP Secure Client**. It provides fully configurable features with the highest security level. For this reason we took this product into consideration.

SonicWALL

SonicWALL provides a complete solution (SonicWALL - Internet Security Appliance) that meets all our requirements. It's composed by the **SonicWALL TZ 170** box in conjunction with the **SonicWALL Global VPN Client** software. For this reason we took this product into consideration.

Chapter 7

Orinoco AP-500 test

7.1 Introduction on Orinoco AP Manager

The Orinoco AP Manager program has been designed to monitor a network from a central location (e.g. the LAN administrator station) enabling the administrator to monitor wireless performance in areas that cannot easily be reached.

The Orinoco AP Manager is capable to:

- display a standard set of SNMP variables to monitor general LAN traffic performance in the network
- display remote link test measurements between a (remote) Orinoco AP-500 and a wireless station connected to the selected access point.

7.2 Remote Monitoring

In order to monitor the Orinoco AP-500, the AP Manager must first connect to the target access point; this can be done selecting the target access point from the local list or enter the IP address of the desired access point (see figure 7.1). In situations where no IP addresses are assigned automatically (e.g. by a DHCP server), the IP address will be the factory default 153.69.254.254; so the administrator must change this factory IP address upon first configuration.

N.B.: only access points in the same subnet of the Orinoco management station are displayed in the list.

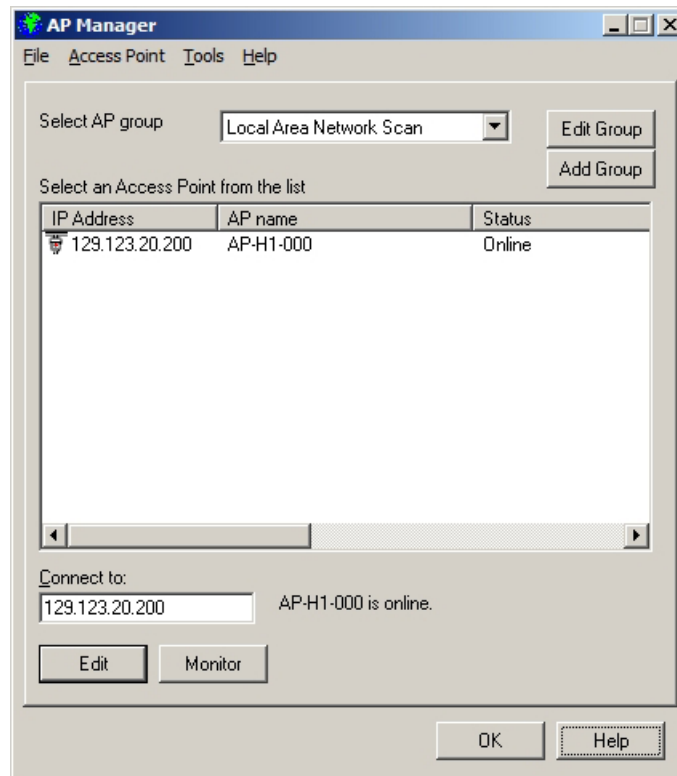


Figure 7.1: The main AP Manager window

In figure 7.1, by clicking on the *Monitor* button you'll be prompted to monitor the wireless network by connecting to the target access point (in our case: 129.123.20.200). Before this, a window for password request appears as shown in figure 7.2; then the monitor window of the AP Manager is displayed and all the monitor tabs are accessible (figure 7.3).

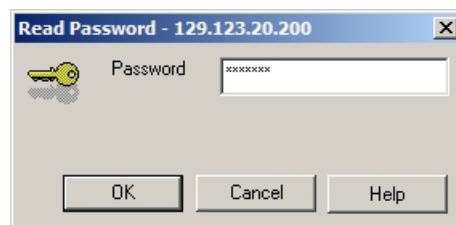


Figure 7.2: The password request window

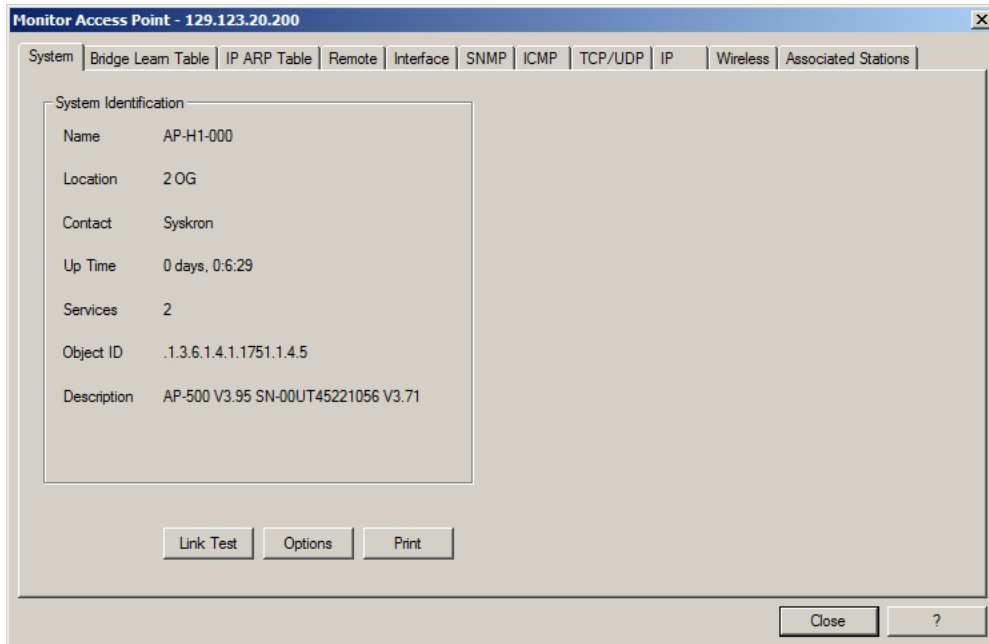


Figure 7.3: The system information window

7.2.1 System tab

In particular we have decided to show the *System* tab which does not provide on-line statistics, but it's primarily used to verify the version level of the embedded software that is loaded into the Orinoco AP-500. These information are the following:

- The **Name**, **Location** and **Contact** represent the values that have been entered in the corresponding fields of the *SNMP* tab in the *Edit mode* when the access point was configured (figure 7.10 on page 122).
- The **Up Time** field displays the time interval measured from the last reset
- The **Services** and **Object ID** do not display relevant information to end-users; they are only needed when contacting the Orinoco technical support
- The **Description** is the most important field of this screen. It allows to determine if the Orinoco AP-500 is running with the latest embedded software available.

It's also possible to display the system interval parameters to monitor the Orinoco AP-500 by clicking on the *Options* button, as shown in figure 7.4.

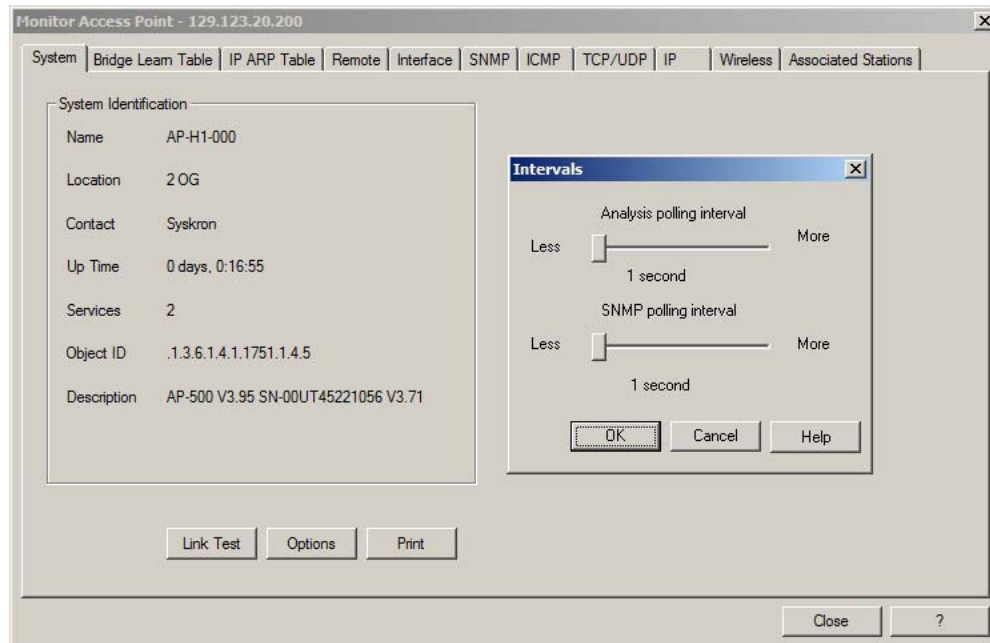


Figure 7.4: The Intervals window

In the Intervals window two different time interval parameters can be set to change the monitor interval settings:

- Analysis polling interval (used for remote link test)
- SNMP polling interval (used for SNMP statistics).

While the remote link test (explained in the next paragraph) will proceed continuously collecting measurement results, the selected AP will transfer the results to the LAN administrator station at regular intervals, that can vary from 1 to 15 seconds. The data displayed in the remote tab refreshes at regular intervals that can vary between 1 second and 5 minutes. Adjusting the refresh rate of both of them depends upon the situation; it's better to use a short time interval when troubleshooting or when a full bandwidth is available, instead a longer interval is preferable when running remote statistics only for background display purpose.

Proceeding the exploration of the AP Manager, the *remote Link Test* button enables the administrator to investigate the radio link connection between

the access point (the "initiator station") and one of the stations connected to it. Figure 7.5 shows the "Select a Remote Partner for 129.123.20.200" window where the station for the link test can be selected (x10ddlaptop or SNIFFER2-PC).

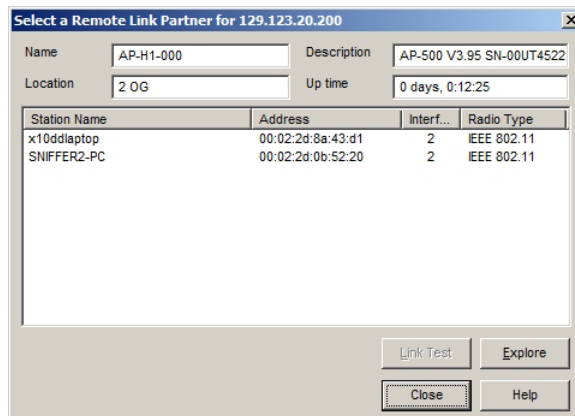


Figure 7.5: The Select a Remote Link Partner for 129.123.20.200 window

After having selected a station from the list and clicked the *Link Test* button, the "Remote Link Test" window is displayed (figure 7.6) showing some interesting information about the radio signal.

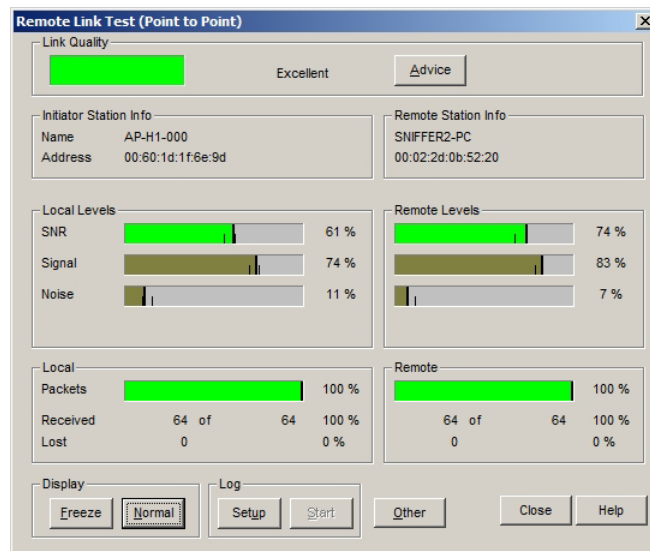
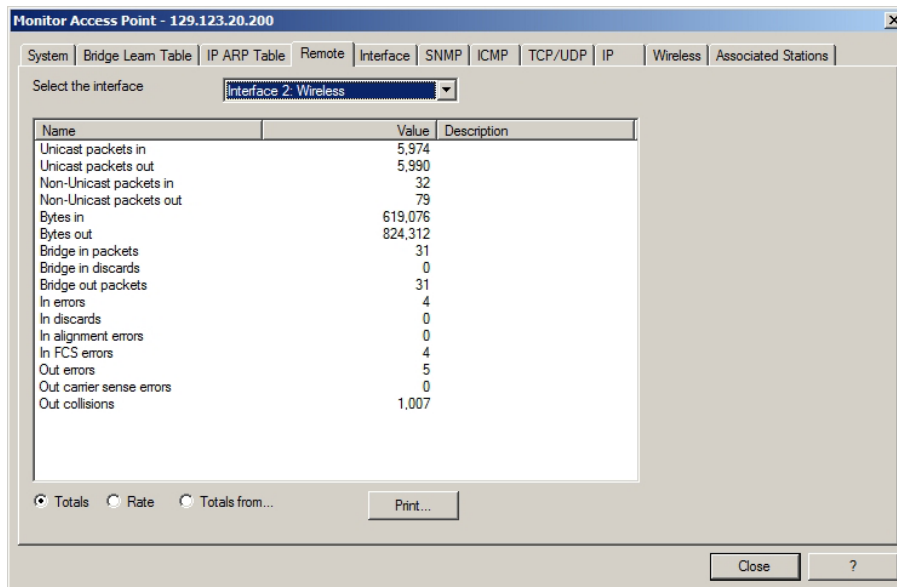


Figure 7.6: The Remote Link Test window

7.2.2 Remote tab

The *Remote* tab statistics allow you to monitor a set of SNMP variables for each of the Orinoco AP-500 interfaces (both Ethernet and Wireless); in figure 7.7 are displayed the statistics for the wireless interface.



Name	Value	Description
Unicast packets in	5,974	
Unicast packets out	5,990	
Non-Unicast packets in	32	
Non-Unicast packets out	79	
Bytes in	619,076	
Bytes out	824,312	
Bridge in packets	31	
Bridge in discards	0	
Bridge out packets	31	
In errors	4	
In discards	0	
In alignment errors	0	
In FCS errors	4	
Out errors	5	
Out carrier sense errors	0	
Out collisions	1,007	

Figure 7.7: The Remote statistics information window

There is a wide range of variables that provide information about the performance of the selected access point; the indicator which provides the main monitoring information is called the *Errors to Bridge Packets* ratio. Other ratios which have particular diagnostic value are:

- In errors / Bridge in packets
- Out errors / Bridge out packets
- Out collisions / Bridge out packets.

Table 7.1 provides diagnostic information referring to each of these three ratios.

Errors to Bridge Packets ratio	Conclusion
0.1% or less	Status: Performance is Good. Impact: None.
0.1% and 1%	Status: Performance is acceptable Impact: Network performance is OK, but the network might not perform as well as expected.
1% or more	Status: Performance is poor. Impact: The performance problem may be caused by network cabling or connections.
2% or more	Status: Performance is very poor. Impact: The network operating systems is likely to face severe performance problems.

Table 7.1: The Errors to Bridge Packets ratio

7.2.3 Wireless tab

The IEEE information on the Wireless tab (in the monitor mode) allows to track frame activity on the IEEE interface of the AP-500.

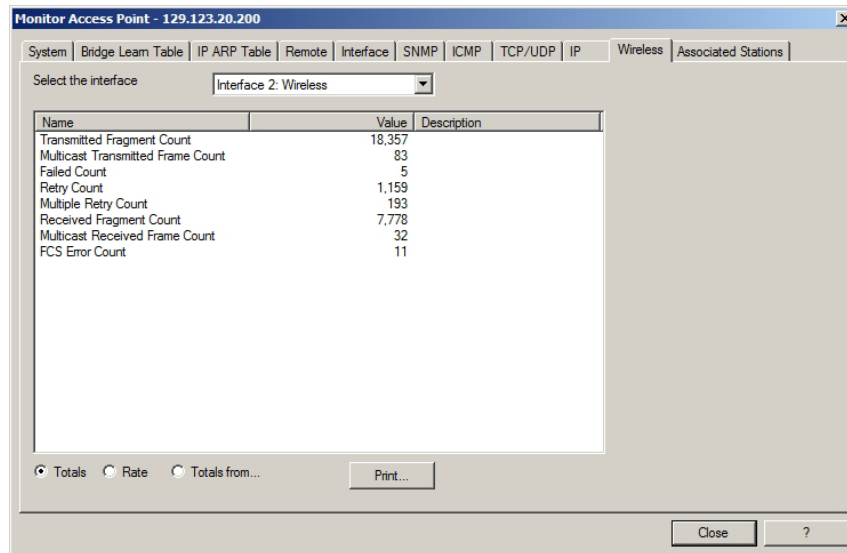


Figure 7.8: The Wireless information window

The three indicators that need particular attention are:

- **Retry Count:** counts the number of frames that are lost (due to collisions) during the initial transmission. During normal operation, the Retry Count should be less than 3% of the Transmitted Fragment Count.
- **Multiple Retry Count:** counts the number of frames that are lost after the initial transmission. During normal operation, the Multiple Retry Count will be less than 3% of the Retry Count.
- **Failed Count:** counts the number of frames that have reached the Retry Limit. Failed frames will no longer attempt to retransmit. If the Failed Count is 1% or more of the Multiple Retry Count, the network may be suffering from interferences.

7.3 Remote Configuring

In order to configure the Orinoco AP-500, the AP Manager must first connect to the target access point as shown before in figure 7.1, and then enter the *Edit* configuration pressing the *Edit* button. As in the Remote Monitor mode, a password would eventually be required!

7.3.1 Wireless Interfaces tab

The most important settings present here are about the *Wireless Interfaces* tab, where the administrator is asked to setup the security settings for the AP.

To exclude unknown and unauthorized computing devices from establishing a wireless connection to the network, the administrator can use the following options:

- Closing the network to all stations that have not been programmed with the correct Orinoco network name.
- Use access control tables to build a list of authorized stations allowed to establish a wireless connection with the network.

To provide an higher level of security to the wireless data transmitted, it's possible to use the *Wired Equivalent Privacy (WEP)* data encryption. It's also possible to specify up to 4 different keys to decrypt wireless data,

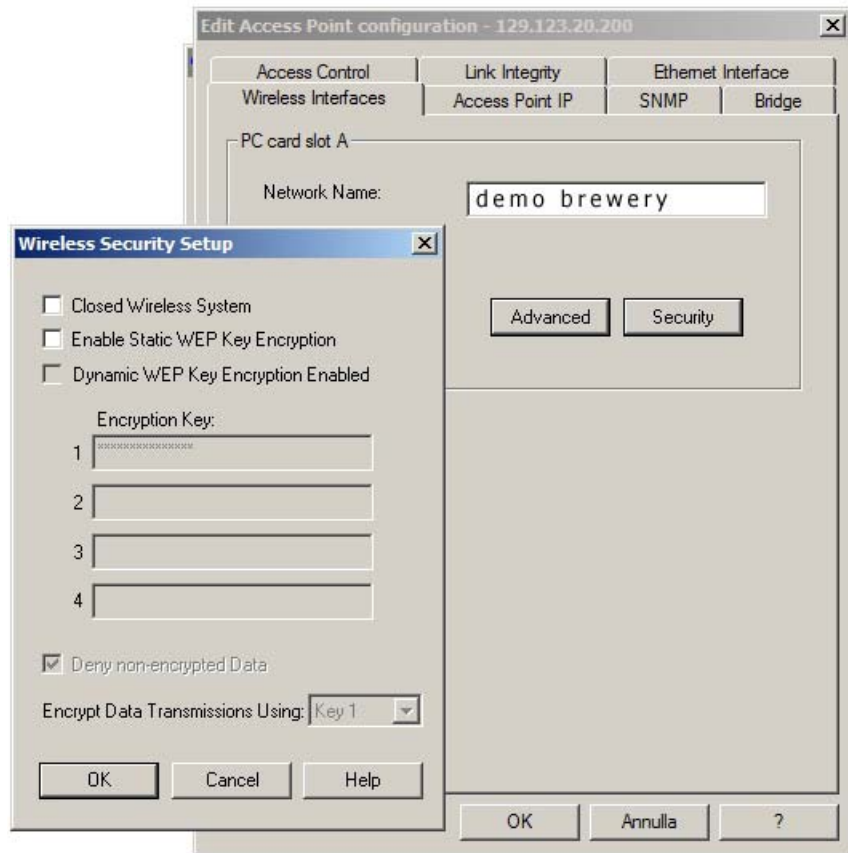


Figure 7.9: The Wireless Security Setup: WEP option window

and select one of the specified decryption key values to encrypt data. The option to use 4 different keys for decrypting wireless data allows the administrator to change WEP keys at regular intervals, without affecting network performance.

In addition, it's possible to restrict access to the Orinoco AP-500 configuration to a limited number of authorized stations via SNMP IP access list. The administrator has also the capabilities to set up Trap Host Alerts; the Trap Host mechanism can inform a network administrator when somebody resets the Orinoco AP-500, performs the forced reload procedure or if there is an authentication failure or a link up or down is detected. The trap host alert will enable the network administrator to verify whether the reset or forced reload action was an authorized action or not.

7.3.2 SNMP tab

In figure 7.10 we can in particular see the *SNMP* information window in which the administrator is allowed to change all the settings available.

Access Control Link Integrity Ethernet Interface

Wireless Interfaces Access Point IP **SNMP** Bridge

Read Password: [masked]

Read/Write Password: [masked]

System Contact: Syskron

System Name: AP-H1-000

System Location: 2 OG

Trap Host IP Address: 0.0.0.0

Trap Host Password: [masked]

SNMP IP Access List

Address	Mask	Interface
<All will be permitted>		

Buttons: Add, Delete, Edit

Bottom buttons: OK, Annulla, ?

Figure 7.10: The Edit Access Point SNMP window

7.3.3 Remote update

As shown in figure 7.11, the AP Manager also gives to the network manager the opportunity to download-upload the configuration files and to update the software without directly accessing the Orinoco AP-500.

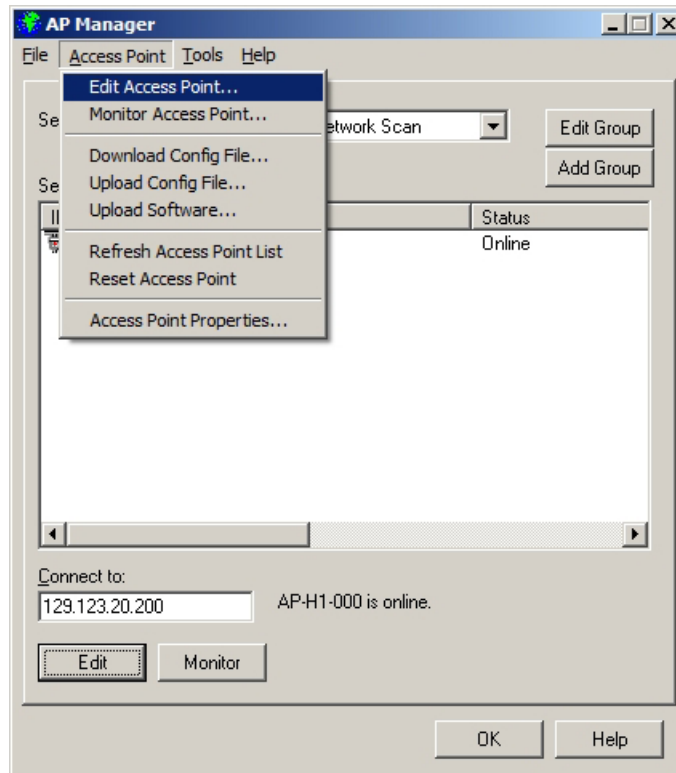


Figure 7.11: The Remote update option window

7.4 Introduction on Orinoco AP-500 MIB

The Orinoco AP-500 uses the following MIBs:

1. RFC1213 - Management Information Base for Network Management of TCP/IP-based internets: MIB-II
2. RFC1398 - Definitions of Managed Objects for the Ethernet-Like Interface Types
3. RFC1493 - Definitions of Managed Objects for Bridges

4. Private MIB (property of Lucent Technologies Inc.) - Management Information Base for monitoring the wireless LAN activity.

The second version of the Management Information Base (MIB-II) is used with network management protocols in TCP/IP-based internets.

The MIB used by the Orinoco AP-500 is composed by the following groups:

- 1.3.6.1.2.1.1 - system
- 1.3.6.1.2.1.2 - interfaces
- 1.3.6.1.2.1.3 - at
- 1.3.6.1.2.1.4 - ip
- 1.3.6.1.2.1.5 - icmp
- 1.3.6.1.2.1.6 - tcp
- 1.3.6.1.2.1.7 - udp
- 1.3.6.1.2.1.8 - egp
- 1.3.6.1.2.1.10 - transmission - dot3
- 1.3.6.1.2.1.11 - snmp
- 1.3.6.1.2.1.17 - dot1dBridge

7.5 MIB Groups

7.5.1 The System group

This group provides contact, administrative, location and service information regarding the managed node.

- 1.3.6.1.2.1.1.1 - sysDescr
- 1.3.6.1.2.1.1.2 - sysObjectID
- 1.3.6.1.2.1.1.3 - sysUpTime
- 1.3.6.1.2.1.1.4 - sysContact
- 1.3.6.1.2.1.1.5 - sysName
- 1.3.6.1.2.1.1.6 - sysLocation
- 1.3.6.1.2.1.1.7 - sysServices

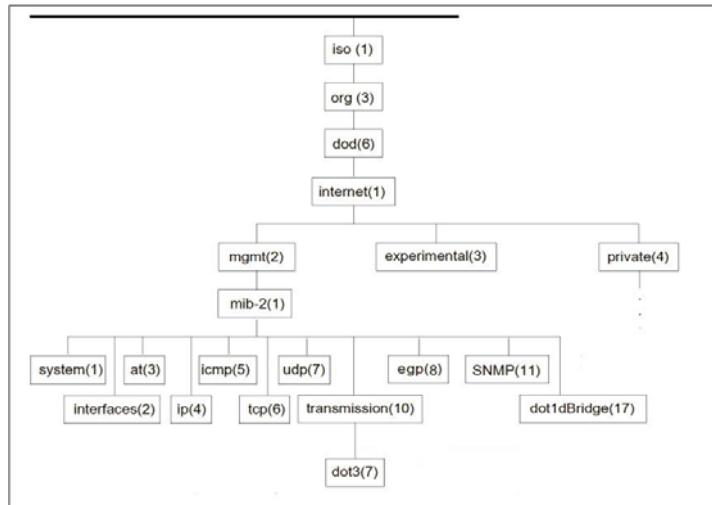


Figure 7.12: The MIB tree

sysDescr - system description (string[128])

sysObjectId - vendor's object identifier (objectid)

sysUpTime - time, in hundredths of seconds, since last system restart (timeticks)

sysContact - name of person to contact

sysName - fully qualified domain name of device

sysLocation - physical location of device

sysServices - services offered by device

7.5.2 The Interfaces group

This group describes the interfaces handled by the proxy agent.

- 1.3.6.1.2.1.2.1 - ifNumber
- 1.3.6.1.2.1.2.2 - ifTable (23 more)
 - 1.3.6.1.2.1.2.2.1.1 - ifIndex
 - 1.3.6.1.2.1.2.2.1.2 - ifDescr
 - 1.3.6.1.2.1.2.2.1.3 - ifType
 - 1.3.6.1.2.1.2.2.1.4 - ifMtu
 - 1.3.6.1.2.1.2.2.1.5 - ifSpeed
 - 1.3.6.1.2.1.2.2.1.6 - ifPhysAddress
 - 1.3.6.1.2.1.2.2.1.7 - ifAdminStatus

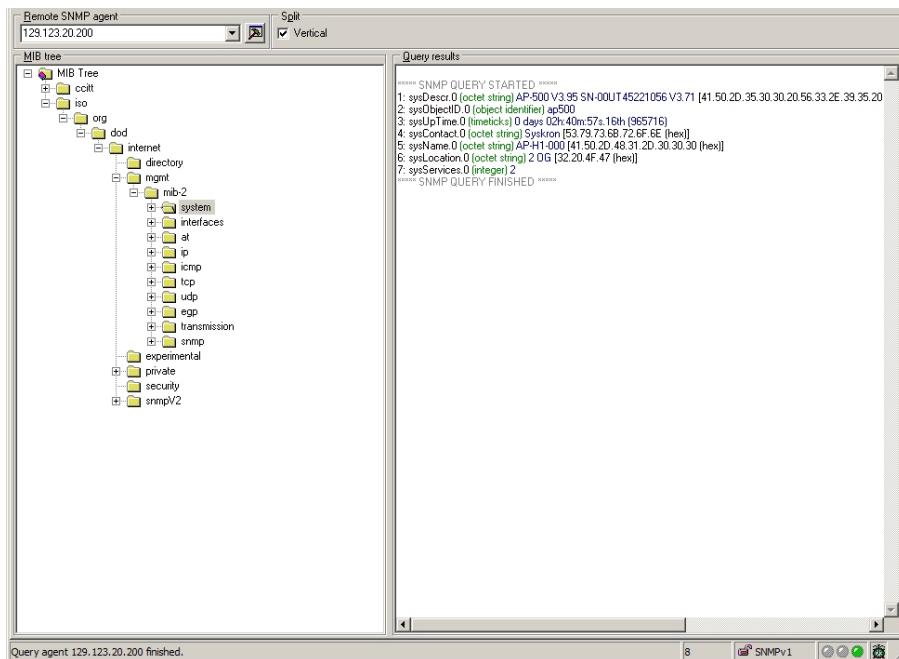


Figure 7.13: The System group

- 1.3.6.1.2.1.2.2.1.8 - ifOperStatus
- 1.3.6.1.2.1.2.2.1.9 - ifLastChange
- 1.3.6.1.2.1.2.2.1.10 - ifInOctets
- 1.3.6.1.2.1.2.2.1.11 - ifInUcastPkts
- 1.3.6.1.2.1.2.2.1.12 - ifInNUcastPkts
- 1.3.6.1.2.1.2.2.1.13 - ifInDiscards
- 1.3.6.1.2.1.2.2.1.14 - ifInErrors
- 1.3.6.1.2.1.2.2.1.15 - ifInUnknownProtos
- 1.3.6.1.2.1.2.2.1.16 - ifOutOctets
- 1.3.6.1.2.1.2.2.1.17 - ifOutUcastPkts
- 1.3.6.1.2.1.2.2.1.18 - ifOutNUcastPkts
- 1.3.6.1.2.1.2.2.1.19 - ifOutDiscards
- 1.3.6.1.2.1.2.2.1.20 - ifOutErrors
- 1.3.6.1.2.1.2.2.1.21 - ifOutQLen

- 1.3.6.1.2.1.2.2.1.22 - ifSpecific

ifNumber - number of network interfaces (int).

The **ifTable** reports status statistics about a particular interface selector (int) RFC1213 interface, using `ifIndex` as the key.

ifIndex - interface selector (int)
ifDescr - interface description (string[128])
ifType - interface type (int)
ifMtu - maximum transmission unit (int)
ifSpeed - interface speed, in bits per second (gauge)
ifPhysAddress - media physical address (octet[36])
ifAdminStatus - administrative status (int)
ifOperStatus - operational status (int)
ifLastChange - time, in hundredths of seconds, since operational state was entered (timeticks)
ifInOctets - number of bytes received (counter)
ifInUcastPkts - number of unicast packets accepted (counter)
ifInNUcastPkts - number of non-unicast packets accepted (counter)
ifInDiscards - number of packets discarded (counter)
ifInErrors - number of malformed packets received (counter)
ifInUnknownProtos - number of packets of unknown protocol (counter)
ifOutOctets - number of octets sent (counter)
ifOutUcastPkts - number of unicast packets sent (counter)
ifOutNUcastPkts - number of non-unicast packets sent (counter)
ifOutDiscards - number of outbound packets discarded (counter)
ifOutErrors - number of output errors (counter)
ifOutQlen - length of output packet queue (gauge)
ifSpecific - information specific to the media being used to realize the interface

7.5.3 The Address Translation group

This group contains one table which is the union across all interfaces of the translation table for converting a NetworkAddress into a subnetwork-specific address. The translation table is equivalent to the ARP cache; in fact it contains the NetworkAddress to "physical" address equivalence.

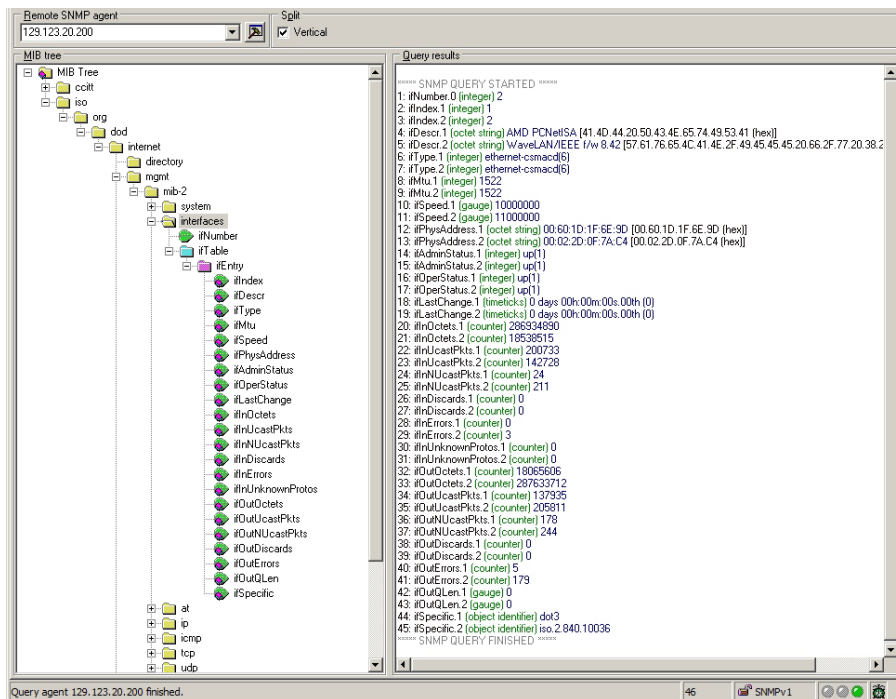


Figure 7.14: The Interfaces group

- 1.3.6.1.2.1.3.1 - atTable (4 more)
 - 1.3.6.1.2.1.3.1.1.1 - atIfIndex
 - 1.3.6.1.2.1.3.1.1.2 - atPhysAddress
 - 1.3.6.1.2.1.3.1.1.3 - atNetAddress

The **ARP** table key consists of the interface number, the constant value 1, and an IP address in dot notation. All fields must be separated by spaces or tabs. Note that the address translation group is marked deprecated in MIB-II.

- atIf2ndex - interface for this entry (int)
- atPhysAddress - media physical address (octet[36])
- atNetAddress - network address (netaddress)

7.5.4 The IP group

This group provides IP informations regarding the managed node.

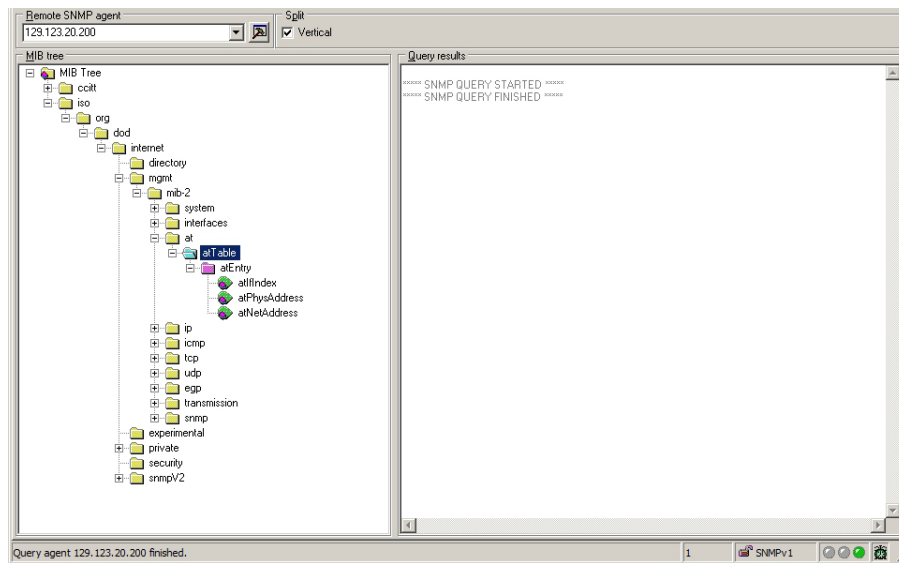


Figure 7.15: The at group

- 1.3.6.1.2.1.4.1 - ipForwarding
- 1.3.6.1.2.1.4.2 - ipDefaultTTL
- 1.3.6.1.2.1.4.3 - ipInReceives
- 1.3.6.1.2.1.4.4 - ipInHdrErrors
- 1.3.6.1.2.1.4.5 - ipInAddrErrors
- 1.3.6.1.2.1.4.6 - ipForwDatagrams
- 1.3.6.1.2.1.4.7 - ipInUnknownProtos
- 1.3.6.1.2.1.4.8 - ipInDiscards
- 1.3.6.1.2.1.4.9 - ipInDelivers
- 1.3.6.1.2.1.4.10 - ipOutRequests
- 1.3.6.1.2.1.4.11 - ipOutDiscards
- 1.3.6.1.2.1.4.12 - ipOutNoRoutes
- 1.3.6.1.2.1.4.13 - ipReasmTimeout
- 1.3.6.1.2.1.4.14 - ipReasmReqds
- 1.3.6.1.2.1.4.15 - ipReasmOKs
- 1.3.6.1.2.1.4.16 - ipReasmFails

- 1.3.6.1.2.1.4.17 - ipFragOKs
- 1.3.6.1.2.1.4.18 - ipFragFails
- 1.3.6.1.2.1.4.19 - ipFragCreates
- 1.3.6.1.2.1.4.20 - ipAddrTable (6 more)
 - 1.3.6.1.2.1.4.20.1.1 - ipAdEntAddr
 - 1.3.6.1.2.1.4.20.1.2 - ipAdEntIfIndex
 - 1.3.6.1.2.1.4.20.1.3 - ipAdEntNetMask
 - 1.3.6.1.2.1.4.20.1.4 - ipAdEntBcastAddr
 - 1.3.6.1.2.1.4.20.1.5 - ipAdEntReasmMaxSize
- 1.3.6.1.2.1.4.21 - ipRouteTable (14 more)
 - 1.3.6.1.2.1.4.21.1.1 - ipRouteDest
 - 1.3.6.1.2.1.4.21.1.2 - ipRouteIfIndex
 - 1.3.6.1.2.1.4.21.1.3 - ipRouteMetric1
 - 1.3.6.1.2.1.4.21.1.4 - ipRouteMetric2
 - 1.3.6.1.2.1.4.21.1.5 - ipRouteMetric3
 - 1.3.6.1.2.1.4.21.1.6 - ipRouteMetric4
 - 1.3.6.1.2.1.4.21.1.7 - ipRouteNextHop
 - 1.3.6.1.2.1.4.21.1.8 - ipRouteType
 - 1.3.6.1.2.1.4.21.1.9 - ipRouteProto
 - 1.3.6.1.2.1.4.21.1.10 - ipRouteAge
 - 1.3.6.1.2.1.4.21.1.11 - ipRouteMask
 - 1.3.6.1.2.1.4.21.1.12 - ipRouteMetric5
 - 1.3.6.1.2.1.4.21.1.13 - ipRouteInfo
- 1.3.6.1.2.1.4.22 - ipNetToMediaTable (5 more)
 - 1.3.6.1.2.1.4.22.1.1 - ipNetToMediaIfIndex
 - 1.3.6.1.2.1.4.22.1.2 - ipNetToMediaPhysAddress
 - 1.3.6.1.2.1.4.22.1.3 - ipNetToMediaNetAddress
 - 1.3.6.1.2.1.4.22.1.4 - ipNetToMediaType
- 1.3.6.1.2.1.4.23 - ipRoutingDiscards

ipForwarding - IP forwarding (int)
ipDefaultTTL - default time-to-live (int)
ipInReceives - number of input datagrams (counter)
ipInHdrErrors - number of input datagrams discarded due to header errors (counter)
ipInAddrErrors - number of input datagrams discarded due to address errors (counter)
ipInForwDatagrams - number of datagrams forwarded (counter)
ipInUnknownProtos - number of input datagrams discarded due to unknown protocol (counter)
ipInDiscards - number of input datagrams discarded due to other reasons (counter)
ipInDelivers - number of datagrams delivered (counter)
ipOutRequests - number of output datagrams requested (counter)
ipOutDiscards - number of output datagrams discarded for other reasons (counter)
ipOutNoRoutes - number of output datagrams discarded for no route (counter)
ipReasmTimeout - IP reassembly timeout, in seconds (int)
ipReasmReqds - number of IP fragments received (counter)
ipReasmOKs - number of datagrams reassembled (counter)
ipReasmFails - number of reassembly failures (counter)
ipFragOKs - number of datagrams fragmented (counter)
ipFragFails - number of datagrams for which fragmentation failed (counter)
ipFragCreates - number of fragments created (counter)

The **ipAddrTable** table reports statistics about the IP address table. The IP address table key is a host or network IP address in dot notation.

ipAdEntAddr - IP address of this entry (netaddress)
ipAdEntIfindex - interface associated with this entry (int)
ipAdEntNetMask - subnet mask associated with this entry (int)
ipAdEntBcastAddr - IP broadcast address of this entry
ipAdEntReasmMaxSiz - maximum size of IP datagram that can be reassembled by the entity

The **ipRoutingTable** table reports statistics about the IP routing table. The IP routing table key is a route or host IP address in dot notation.

ipRouteDest - destination IP address (netaddress)
ipRouteIf2ndex - interface to use (int)

ipRouteMetric1 - route metric 1 (int)
ipRouteMetric2 - route metric 2 (int)
ipRouteMetric3 - route metric 3 (int)
ipRouteMetric4 - route metric 4 (int)
ipRouteNextHop - next hop IP address (netaddress)
ipRouteType - type of route (int)
ipRouteProto - routing protocol used (int)
ipRouteAge - age of this route entry, in seconds (int)
ipRouteMask - subnet mask for route
ipRouteMetric5 - route metric 5 (int)
ipRouteInfo - a reference to MIB definitions specific to the particular routing protocol which is responsible for this route

The **ipNetToMediaTable** table maps IP addresses to physical addresses.

ipNetToMediaIfindex - interface number
ipNetToMediaPhysAddress - media address of mapping
ipNetToMediaNetAddress - IP address of mapping
ipNetToMediaType - type of mapping

ipRoutingDiscards - the number of routing entries which were chosen to be discarded even though they are valid.

7.5.5 The ICMP group

The icmp group reports statistics about the ICMP activity of the managed device.

- 1.3.6.1.2.1.5.2 - icmpInErrors
- 1.3.6.1.2.1.5.3 - icmpInDestUnreaches
- 1.3.6.1.2.1.5.4 - icmpInTimeExcds
- 1.3.6.1.2.1.5.5 - icmpInParmProbs
- 1.3.6.1.2.1.5.6 - icmpInSrcQuenchs
- 1.3.6.1.2.1.5.7 - icmpInRedirects
- 1.3.6.1.2.1.5.8 - icmpInEchos
- 1.3.6.1.2.1.5.9 - icmpInEchoReps
- 1.3.6.1.2.1.5.10 - icmpInTimestamps

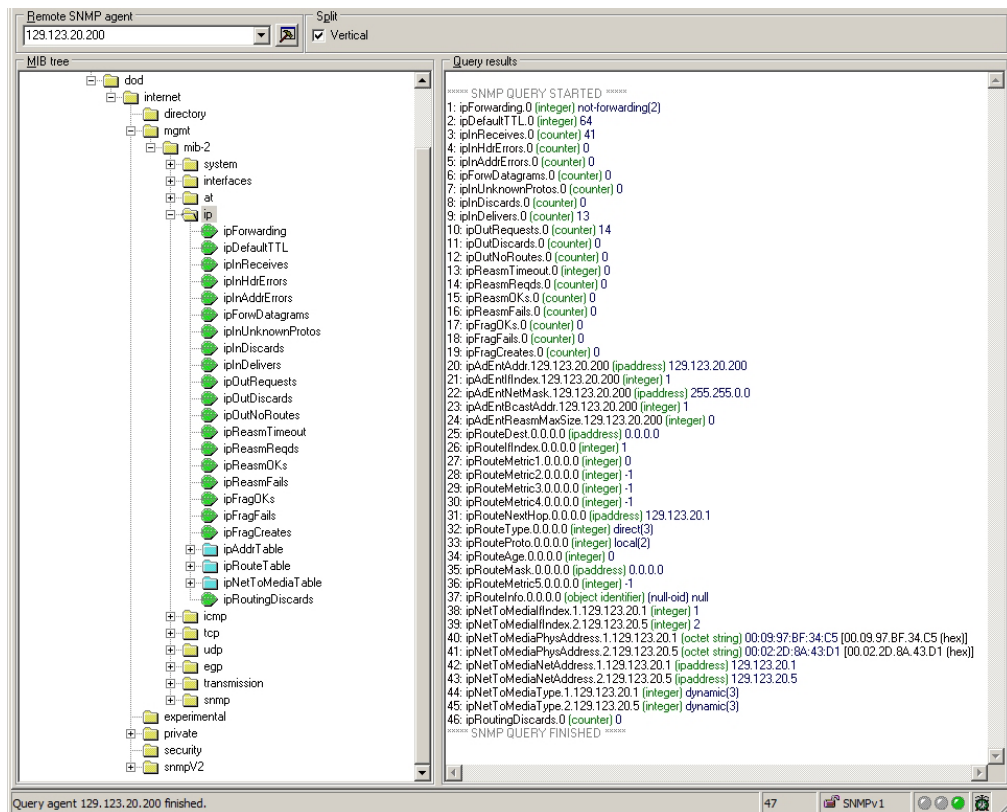


Figure 7.16: The ip group

- 1.3.6.1.2.1.5.11 - icmpInTimestampReps
- 1.3.6.1.2.1.5.12 - icmpInAddrMasks
- 1.3.6.1.2.1.5.13 - icmpInAddrMaskReps
- 1.3.6.1.2.1.5.14 - icmpOutMsgs
- 1.3.6.1.2.1.5.15 - icmpOutErrors
- 1.3.6.1.2.1.5.16 - icmpOutDestUnreachs
- 1.3.6.1.2.1.5.17 - icmpOutTimeExcds
- 1.3.6.1.2.1.5.18 - icmpOutParmProbs
- 1.3.6.1.2.1.5.19 - icmpOutSrcQuenchs
- 1.3.6.1.2.1.5.20 - icmpOutRedirects
- 1.3.6.1.2.1.5.21 - icmpOutEchos

- 1.3.6.1.2.1.5.22 - icmpOutEchoReps
- 1.3.6.1.2.1.5.23 - icmpOutTimestamps
- 1.3.6.1.2.1.5.24 - icmpOutTimestampReps
- 1.3.6.1.2.1.5.25 - icmpOutAddrMasks
- 1.3.6.1.2.1.5.26 - icmpOutAddrMaskReps

icmpInMsgs - number of ICMP messages received (counter)

icmpInErrors - number of ICMP messages received with errors (counter)

icmpInDestUnreachs - number of ICMP destination unreachables received (counter)

icmpInTimeExcds - number of ICMP time exceeded received (counter)

icmpInParmProbs - number of ICMP parameter problems received (counter)

icmpInSrcQuenches - number of ICMP source quenches received (counter)

icmpInRedirects - number of ICMP redirects received (counter)

icmpInEchos - number of ICMP echo requests received (counter)

icmpInEchoReps - number of ICMP echo replies received (counter)

icmpInTimestamps - number of ICMP timestamp requests received (counter)

icmpInTimestampReps - number of ICMP timestamp replies received (counter)

icmpInAddrMasks - number of ICMP address mask requests received (counter)

icmpInAddrMaskReps - number of ICMP address mask replies received (counter)

icmpOutMsgs - number of ICMP messages requested to be sent (counter)

icmpOutErrors - number of ICMP messages not sent due to errors (counter)

icmpOutDestUnreachs - number of ICMP destination unreachables sent (counter)

icmpOutTimeExcds - number of ICMP time exceeded sent (counter)

icmpOutParmProbs - number of ICMP parameter problems sent (counter)

icmpOutSrcQuenches - number of ICMP source quenches sent (counter)

icmpOutRedirects - number of ICMP redirects sent (counter)

icmpOutEchos - number of ICMP echo requests sent (counter)

icmpOutEchoReps - number of ICMP echo replies sent (counter)

icmpOutTimestamps - number of ICMP timestamp requests sent (counter)

icmpOutTimestampReps - number of ICMP timestamp replies sent (counter)

icmpOutAddrMasks - number of ICMP address mask requests sent (counter)

icmpOutAddrMaskReps - number of ICMP address mask replies sent (counter)

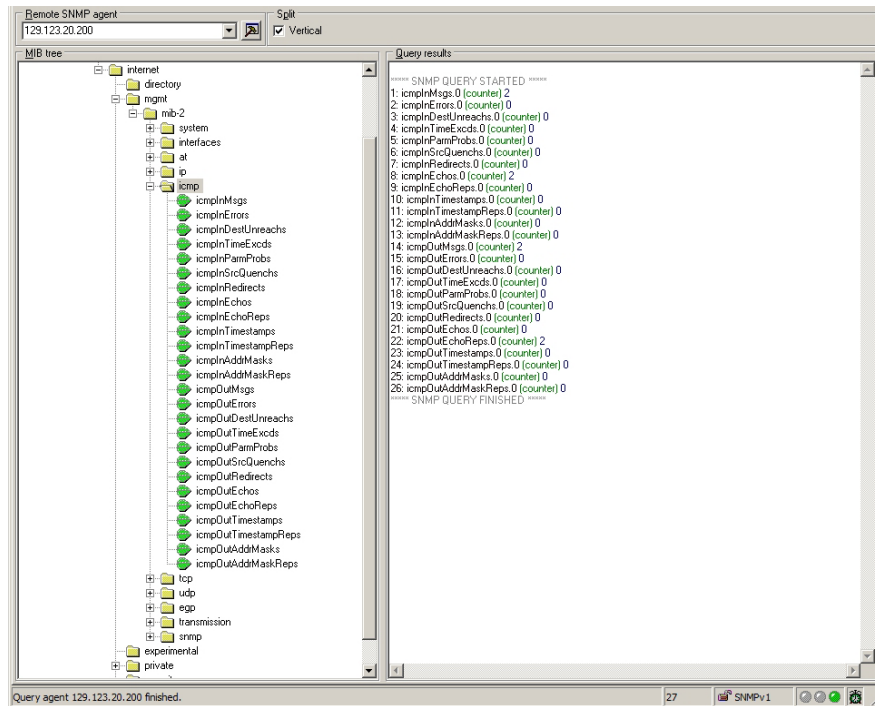


Figure 7.17: The icmp group

7.5.6 The TCP group

The tcp group reports statistics about the TCP connections to the managed device. These informations are transient; they persist only as long as the connection in question.

- 1.3.6.1.2.1.6.1 - tcpRtoAlgorithm
- 1.3.6.1.2.1.6.2 - tcpRtoMin
- 1.3.6.1.2.1.6.3 - tcpRtoMax
- 1.3.6.1.2.1.6.4 - tcpMaxConn
- 1.3.6.1.2.1.6.5 - tcpActiveOpens
- 1.3.6.1.2.1.6.6 - tcpPassiveOpens
- 1.3.6.1.2.1.6.7 - tcpAttemptFails
- 1.3.6.1.2.1.6.8 - tcpEstabResets
- 1.3.6.1.2.1.6.9 - tcpCurrEstab
- 1.3.6.1.2.1.6.10 - tcpInSegs

- 1.3.6.1.2.1.6.11 - tcpOutSegs
- 1.3.6.1.2.1.6.12 - tcpRetransSegs
- 1.3.6.1.2.1.6.13 - tcpConnTable (6 more)
 - 1.3.6.1.2.1.6.13.1.1 - tcpConnState
 - 1.3.6.1.2.1.6.13.1.2 - tcpConnLocalAddress
 - 1.3.6.1.2.1.6.13.1.3 - tcpConnLocalPort
 - 1.3.6.1.2.1.6.13.1.4 - tcpConnRemAddress
 - 1.3.6.1.2.1.6.13.1.5 - tcpConnRemPort
- 1.3.6.1.2.1.6.14 - tcpInErrs
- 1.3.6.1.2.1.6.15 - tcpOutRsts

tcpRtoAlgorithm - TCP round trip algorithm (int)
tcpRtoMin - minimum round trip time, in milliseconds (int)
tcpRtoMax - maximum round trip time, in milliseconds (int)
tcpMaxConn - maximum number of TCP connections (int)
tcpActiveOpens - number of TCP connections actively open (counter)
tcpPassiveOpens - number of TCP connections passively open (counter)
tcpAttemptFails - number of failed connection attempts (counter)
tcpEstabResets - number of connection resets (counter)
tcpCurrEstab - number of TCP connections currently open (gauge)
tcpInSegs - number of segments received on TCP connections (counter)
tcpOutSegs - number of segments sent on TCP connections (counter)
tcpRetransSegs - number of TCP segments retransmitted (counter)
tcpInErrs - number of TCP segments discarded because of format error
tcpOutRsts - number of resets generated

The **tcpConnTable** table contains information about this entity's existing TCP connections. The TCP connection table key is a socket pair consisting of a local IP address expressed in dot notation, followed by a local port number, followed by a remote IP address in dot notation, followed by a remote port number. All fields must be separated by spaces or tabs.

tcpConnState - TCP connection state (int)
tcpConnLocalAddress - local IP address (netaddress)
tcpConnLocalPort - local TCP port (int)
tcpConnRemAddress - remote IP address (netaddress)
tcpConnRemPort - remote TCP port (int)

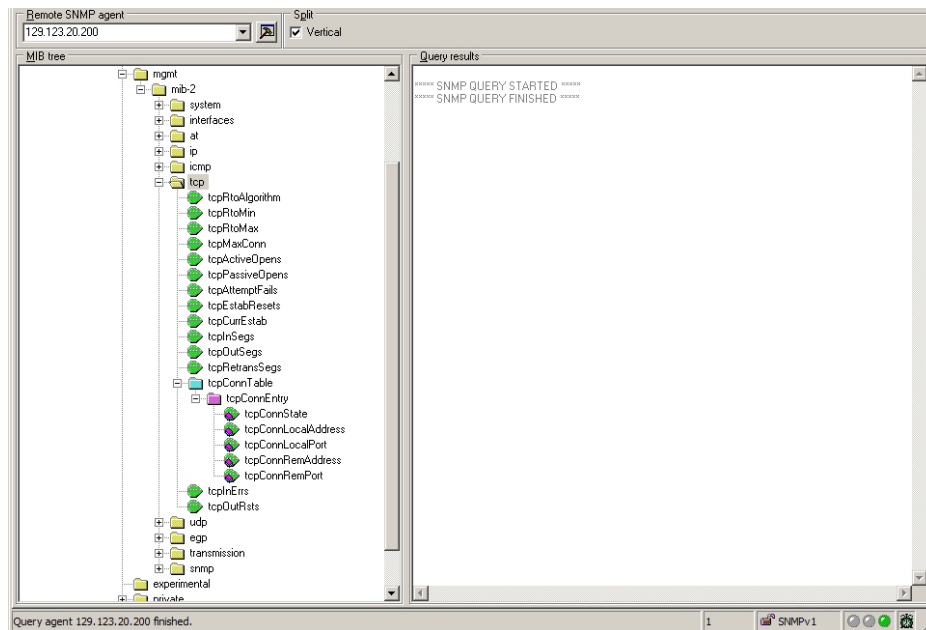


Figure 7.18: The tcp group

7.5.7 The UDP group

The udp group reports statistics about the UDP activity of the managed device.

- 1.3.6.1.2.1.7.1 - udpInDatagrams
- 1.3.6.1.2.1.7.2 - udpNoPorts
- 1.3.6.1.2.1.7.3 - udpInErrors
- 1.3.6.1.2.1.7.4 - udpOutDatagrams
- 1.3.6.1.2.1.7.5 - udpTable (3 more)
 - 1.3.6.1.2.1.7.5.1.1 - udpLocalAddress
 - 1.3.6.1.2.1.7.5.1.2 - udpLocalPort

udpInDatagrams - number of UDP datagrams received and delivered (counter)

udpNoPorts - number of UDP datagrams discarded due to no listen on local port (counter)

udpInErrors - number of UDP datagrams discarded due to other errors (counter)

udpOutDatagrams - number of UDP datagrams sent (counter)

The **udpTable** table reports UDP listener information.

udpLocalAddress - local IP address

udpLocalPort - local UDP port

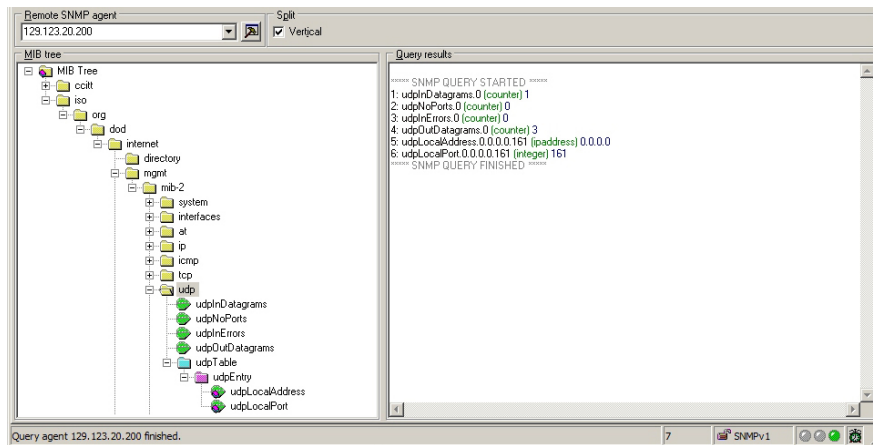


Figure 7.19: The udp group

7.5.8 The EGP group

The `egp` group reports statistics about the EGP activity of the managed device.

- 1.3.6.1.2.1.8.1 - `egpInMsgs`
- 1.3.6.1.2.1.8.2 - `egpInErrors`
- 1.3.6.1.2.1.8.3 - `egpOutMsgs`
- 1.3.6.1.2.1.8.4 - `egpOutErrors`
- 1.3.6.1.2.1.8.5 - `egpNeighTable` (16 more)
 - 1.3.6.1.2.1.8.5.1.1 - `egpNeighState`
 - 1.3.6.1.2.1.8.5.1.2 - `egpNeighAddr`
 - 1.3.6.1.2.1.8.5.1.3 - `egpNeighAs`
 - 1.3.6.1.2.1.8.5.1.4 - `egpNeighInMsgs`
 - 1.3.6.1.2.1.8.5.1.5 - `egpNeighInErrs`
 - 1.3.6.1.2.1.8.5.1.6 - `egpNeighOutMsgs`
 - 1.3.6.1.2.1.8.5.1.7 - `egpNeighOutErrs`
 - 1.3.6.1.2.1.8.5.1.8 - `egpNeighInErrMsgs`
 - 1.3.6.1.2.1.8.5.1.9 - `egpNeighOutErrMsgs`
 - 1.3.6.1.2.1.8.5.1.10 - `egpNeighStateUps`
 - 1.3.6.1.2.1.8.5.1.11 - `egpNeighStateDowns`
 - 1.3.6.1.2.1.8.5.1.12 - `egpNeighIntervalHello`
 - 1.3.6.1.2.1.8.5.1.13 - `egpNeighIntervalPoll`
 - 1.3.6.1.2.1.8.5.1.14 - `egpNeighMode`
 - 1.3.6.1.2.1.8.5.1.15 - `egpNeighEventTrigger`
- 1.3.6.1.2.1.8.6 - `egpAs`

`egpInMsgs` - number of EGP messages received without error (counter)

`egpInErrors` - number of EGP messages received with error (counter)

`egpOutMsgs` - number of locally generated EGP messages (counter)

`egpOutErrors` - number of EGP messages not sent due to errors (counter)

The **`egpNeighTable`** table reports statistics about the EGP neighbor table. The EGP neighbor key is a host IP address in dot notation.

egpNeighState - EGP state of neighbor (int)
egpNeighAddress - IP address of EGP neighbor (netaddress)
egpNeighAs - autonomous system number (int)
egpNeighInMsgs - number of EGP messages received without error (counter)
egpNeighInErrs - number of EGP messages received with error (counter)
egpNeighOutMsgs - number of locally-generated EGP messages (counter)
egpNeighOutErrs - number of locally-generated EGP messages not sent (counter)
egpNeighInErrMsgs - number of EGP error messages received (counter)
egpNeighOutErrMsgs - number of EGP error messages sent (counter)
egpNeighStateUps - number of EGP state transitions to the UP state (counter)
egpNeighStateDowns - number of EGP state transitions from the UP state (counter)
egpNeighIntervalHello - interval (in hundredths of a second) between EGP Hello command retransmissions (int)
egpNeighIntervalPoll - interval (in hundredths of a second) between EGP poll command retransmissions (int)
egpNeighMode - polling mode of EGP entity (int)
egpNeighEventTrigger - trigger for operator-initiated Start and Stop events (int)

egpAs - autonomous system number of entity (int)

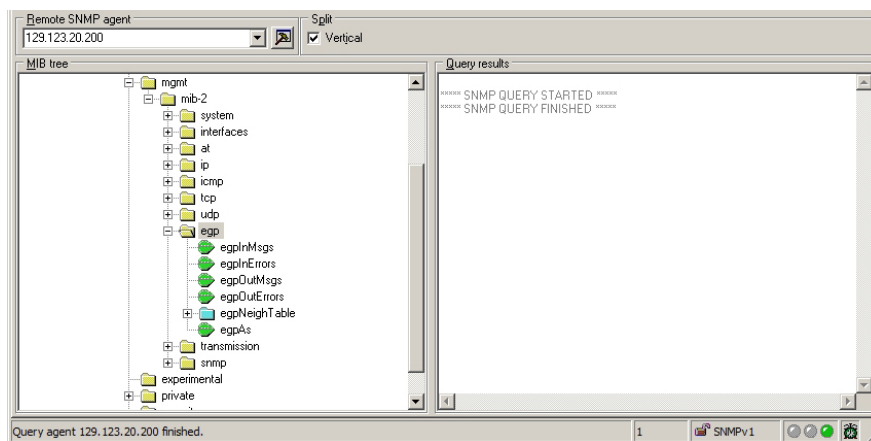


Figure 7.20: The egp group

7.5.9 The Transmission group

The transmission group reports statistics about transmission media. Note that RFC 1213 does not define MIB objects in this group.

The Orinoco AP-500 only supports the dot3 MIB (RFC1398) in the transmission group; it defines objects for managing ethernet-like objects.

- 1.3.6.1.2.1.10.7 - dot3
- 1.3.6.1.2.1.10.7.2 - dot3StatsTable (20 more)
 - 1.3.6.1.2.1.10.7.2.1.1 - dot3StatsIndex
 - 1.3.6.1.2.1.10.7.2.1.2 - dot3StatsAlignmentErrors
 - 1.3.6.1.2.1.10.7.2.1.3 - dot3StatsFCSErrors
 - 1.3.6.1.2.1.10.7.2.1.4 - dot3StatsSingleCollisionFrames
 - 1.3.6.1.2.1.10.7.2.1.5 - dot3StatsMultipleCollisionFrames
 - 1.3.6.1.2.1.10.7.2.1.6 - dot3StatsSQETestErrors
 - 1.3.6.1.2.1.10.7.2.1.7 - dot3StatsDeferredTransmissions
 - 1.3.6.1.2.1.10.7.2.1.8 - dot3StatsLateCollisions
 - 1.3.6.1.2.1.10.7.2.1.9 - dot3StatsExcessiveCollisions
 - 1.3.6.1.2.1.10.7.2.1.10 - dot3StatsInternalMacTransmitErrors
 - 1.3.6.1.2.1.10.7.2.1.11 - dot3StatsCarrierSenseErrors
 - 1.3.6.1.2.1.10.7.2.1.13 - dot3StatsFrameTooLongs
 - 1.3.6.1.2.1.10.7.2.1.16 - dot3StatsInternalMacReceiveErrors
 - 1.3.6.1.2.1.10.7.2.1.17 - dot3StatsEtherChipSet
 - 1.3.6.1.2.1.10.7.2.1.18 - dot3StatsSymbolErrors
 - 1.3.6.1.2.1.10.7.2.1.19 - dot3StatsDuplexStatus
- 1.3.6.1.2.1.10.7.5 - dot3CollTable (3 more)
 - 1.3.6.1.2.1.10.7.5.1.2 - dot3CollCount
 - 1.3.6.1.2.1.10.7.5.1.3 - dot3CollFrequencies
- 1.3.6.1.2.1.10.7.6 - dot3Tests
 - 1.3.6.1.2.1.10.7.6.1 - dot3TestTdr
 - 1.3.6.1.2.1.10.7.6.2 - dot3TestLoopBack

- 1.3.6.1.2.1.10.7.7 - dot3Errors
- 1.3.6.1.2.1.10.7.7.1 - dot3ErrorInitError
- 1.3.6.1.2.1.10.7.7.2 - dot3ErrorLoopbackError
- 1.3.6.1.2.1.10.7.9 - dot3ControlTable (3 more)
 - 1.3.6.1.2.1.10.7.9.1.1 - dot3ControlFunctionsSupported
 - 1.3.6.1.2.1.10.7.9.1.2 - dot3ControlInUnknownOpCodes
- 1.3.6.1.2.1.10.7.10 - dot3PauseTable (4 more)
 - 1.3.6.1.2.1.10.7.10.1.1 - dot3PauseAdminMode
 - 1.3.6.1.2.1.10.7.10.1.2 - dot3PauseOperMode
 - 1.3.6.1.2.1.10.7.10.1.3 - dot3InPauseFrames

The **Ethernet-like Statistics** group defines statistics for a collection of Ethernet-like interfaces attached to a particular system.

The **Ethernet-like Collision Statistics** group defines a collection of collision histograms for a particular set of interfaces.

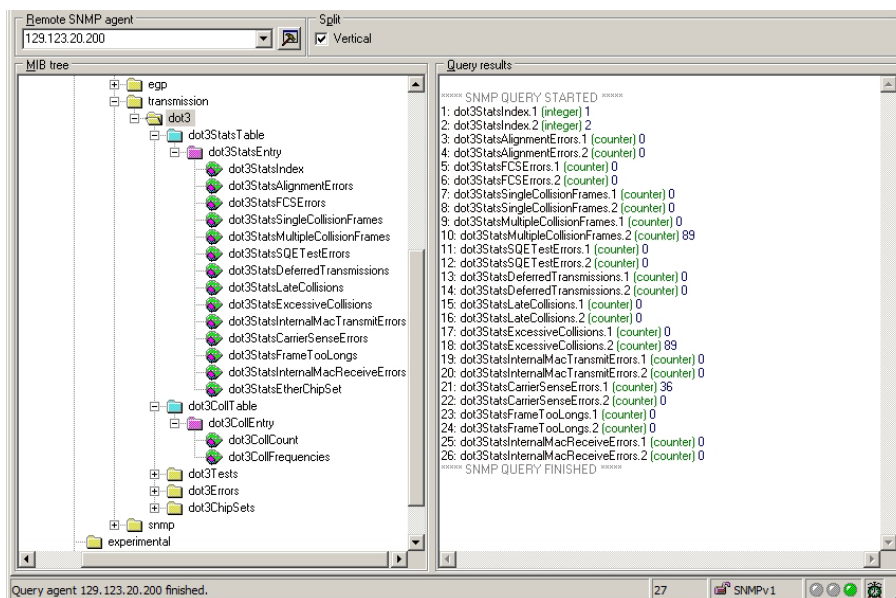


Figure 7.21: The Transmission group

7.5.10 The SNMP group

The snmp group reports statistics about the SNMP activity of the managed device.

- 1.3.6.1.2.1.11.2 - snmpOutPkts
- 1.3.6.1.2.1.11.3 - snmpInBadVersions
- 1.3.6.1.2.1.11.4 - snmpInBadCommunityNames
- 1.3.6.1.2.1.11.5 - snmpInBadCommunityUses
- 1.3.6.1.2.1.11.6 - snmpInASNParseErrs
- 1.3.6.1.2.1.11.8 - snmpInTooBig
- 1.3.6.1.2.1.11.9 - snmpInNoSuchNames
- 1.3.6.1.2.1.11.10 - snmpInBadValues
- 1.3.6.1.2.1.11.11 - snmpInReadOnly
- 1.3.6.1.2.1.11.12 - snmpInGenErrs
- 1.3.6.1.2.1.11.13 - snmpInTotalReqVars
- 1.3.6.1.2.1.11.14 - snmpInTotalSetVars
- 1.3.6.1.2.1.11.15 - snmpInGetRequests
- 1.3.6.1.2.1.11.16 - snmpInGetNexts
- 1.3.6.1.2.1.11.17 - snmpInSetRequests
- 1.3.6.1.2.1.11.18 - snmpInGetResponses
- 1.3.6.1.2.1.11.19 - snmpInTraps
- 1.3.6.1.2.1.11.20 - snmpOutTooBig
- 1.3.6.1.2.1.11.21 - snmpOutNoSuchNames
- 1.3.6.1.2.1.11.22 - snmpOutBadValues
- 1.3.6.1.2.1.11.24 - snmpOutGenErrs
- 1.3.6.1.2.1.11.25 - snmpOutGetRequests
- 1.3.6.1.2.1.11.26 - snmpOutGetNexts
- 1.3.6.1.2.1.11.27 - snmpOutSetRequests
- 1.3.6.1.2.1.11.28 - snmpOutGetResponses
- 1.3.6.1.2.1.11.29 - snmpOutTraps
- 1.3.6.1.2.1.11.30 - snmpEnableAuthenTraps

snmpInPkts - number of messages delivered from transport service (counter)
snmpInTotalReqVars - number of MIB objects retrieved successfully (counter)
snmpInTotalSetVars - number of MIB objects altered successfully (counter)
snmpInGetRequests - number of SNMP Get-Request PDUs accepted and processed (counter)
snmpInGetNexts - number of SNMP Get-Next PDUs accepted and processed (counter)
snmpInSetRequests - number of SNMP Set-Request PDUs accepted and processed (counter)
snmpInGetResponses - number of SNMP Get-Response PDUs accepted and processed (counter)
snmpInTraps - number of SNMP Trap PDUs accepted and processed (counter)
snmpInBadVersions - number of SNMP messages delivered for an unsupported SNMP version (counter)
snmpInBadCommunityNames - number of SNMP messages delivered which used an unknown SNMP community name (counter)
snmpInBadCommunityUses - number of SNMP messages delivered which represented an operation not allowed by the SNMP community (counter)
snmpInASNParseErrs - number of ASN.1 or BER errors encountered when decoding received messages (counter)
snmpInBadTypes - reserved
snmpInTooBigs - number of SNMP PDUs delivered for which the error-status field is 'tooBig' (counter)
snmpInNoSuchNames - number of SNMP PDUs delivered for which the error-status field is 'noSuchName' (counter)
snmpInBadValues - number of SNMP PDUs delivered for which the error-status field is 'badValue' (counter)
snmpInReadOnlys - number of SNMP PDUs delivered for which the error-status field is 'readOnly' (counter)
snmpInGenErrs - number of SNMP PDUs delivered for which the error-status field is 'genErr' (counter)
snmpOutPkts - number of SNMP message passed to transport service (counter)
snmpOutGetRequests - number of SNMP Get-Request PDUS generated (counter)
snmpOutGetNexts - number of SNMP Get-Next PDUs generated (counter)
snmpOutSetRequests - number of SNMP Set-Request PDUs generated (counter)
snmpOutGetResponses - number of SNMP Get-Response PDUS generated (counter)
snmpOutTraps - number of SNMP Trap PDUs generated (counter)
snmpOutTooBigs - number of SNMP PDUS generated for which the error-status field is 'tooBig' (counter)
snmpOutNoSuchNames - number of SNMP PDUs generated for which the error-status field is 'noSuchName' (counter)

snmpOutBadValues - number of SNMP PDUs generated for which the error-status field is 'badValue' (counter)

snmpOutReadOnlys - reserved

snmpOutGenErrs- number of SNMP PDUs generated for which the error-status field is 'genErr' (counter)

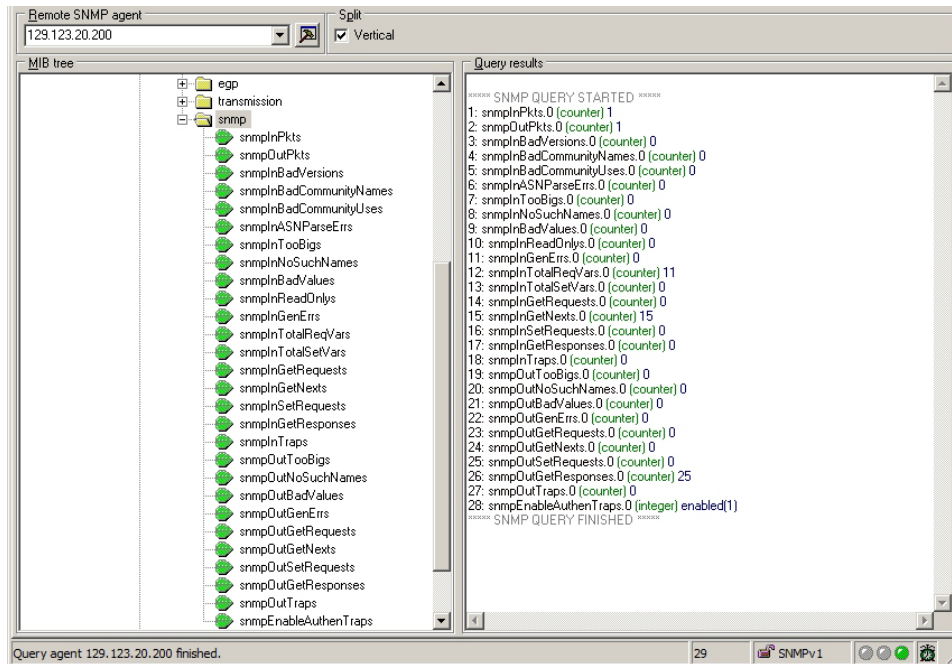


Figure 7.22: The snmp group

The dot1dBridge group

The Orinoco AP-500 also supports this dot1dBridge MIB (RFC1493); it defines objects for managing MAC bridges based on the IEEE 802.1D-1990 standard between Local Area Network (LAN) segment.

- 1.3.6.1.2.1.17 - dot1dBridge
- 1.3.6.1.2.1.17.1 - dot1dBase
- 1.3.6.1.2.1.17.1.1 - dot1dBaseBridgeAddress
- 1.3.6.1.2.1.17.1.2 - dot1dBaseNumPorts
- 1.3.6.1.2.1.17.1.3 - dot1dBaseType

- 1.3.6.1.2.1.17.1.4 - dot1dBasePortTable (6 more)
 - 1.3.6.1.2.1.17.1.4.1.1 - dot1dBasePort
 - 1.3.6.1.2.1.17.1.4.1.2 - dot1dBasePortIfIndex
 - 1.3.6.1.2.1.17.1.4.1.3 - dot1dBasePortCircuit
 - 1.3.6.1.2.1.17.1.4.1.4 - dot1dBasePortDelayExceededDiscards
 - 1.3.6.1.2.1.17.1.4.1.5 - dot1dBasePortMtuExceededDiscards
- 1.3.6.1.2.1.17.2 - dot1dStp
- 1.3.6.1.2.1.17.2.1 - dot1dStpProtocolSpecification
- 1.3.6.1.2.1.17.2.2 - dot1dStpPriority
- 1.3.6.1.2.1.17.2.3 - dot1dStpTimeSinceTopologyChange
- 1.3.6.1.2.1.17.2.4 - dot1dStpTopChanges
- 1.3.6.1.2.1.17.2.5 - dot1dStpDesignatedRoot
- 1.3.6.1.2.1.17.2.6 - dot1dStpRootCost
- 1.3.6.1.2.1.17.2.7 - dot1dStpRootPort
- 1.3.6.1.2.1.17.2.8 - dot1dStpMaxAge
- 1.3.6.1.2.1.17.2.9 - dot1dStpHelloTime
- 1.3.6.1.2.1.17.2.10 - dot1dStpHoldTime
- 1.3.6.1.2.1.17.2.11 - dot1dStpForwardDelay
- 1.3.6.1.2.1.17.2.12 - dot1dStpBridgeMaxAge
- 1.3.6.1.2.1.17.2.13 - dot1dStpBridgeHelloTime
- 1.3.6.1.2.1.17.2.14 - dot1dStpBridgeForwardDelay
- 1.3.6.1.2.1.17.2.15 - dot1dStpPortTable (11 more)
 - 1.3.6.1.2.1.17.2.15.1.1 - dot1dStpPort
 - 1.3.6.1.2.1.17.2.15.1.2 - dot1dStpPortPriority
 - 1.3.6.1.2.1.17.2.15.1.3 - dot1dStpPortState
 - 1.3.6.1.2.1.17.2.15.1.4 - dot1dStpPortEnable
 - 1.3.6.1.2.1.17.2.15.1.5 - dot1dStpPortPathCost
 - 1.3.6.1.2.1.17.2.15.1.6 - dot1dStpPortDesignatedRoot
 - 1.3.6.1.2.1.17.2.15.1.7 - dot1dStpPortDesignatedCost
 - 1.3.6.1.2.1.17.2.15.1.8 - dot1dStpPortDesignatedBridge

- 1.3.6.1.2.1.17.2.15.1.9 - dot1dStpPortDesignatedPort
- 1.3.6.1.2.1.17.2.15.1.10 - dot1dStpPortForwardTransitions
- 1.3.6.1.2.1.17.3 - dot1dSr
- 1.3.6.1.2.1.17.4 - dot1dTp
 - 1.3.6.1.2.1.17.4.1 - dot1dTpLearnedEntryDiscards
 - 1.3.6.1.2.1.17.4.2 - dot1dTpAgingTime
 - 1.3.6.1.2.1.17.4.3 - dot1dTpFdbTable (4 more)
 - 1.3.6.1.2.1.17.4.3.1.1 - dot1dTpFdbAddress
 - 1.3.6.1.2.1.17.4.3.1.2 - dot1dTpFdbPort
 - 1.3.6.1.2.1.17.4.3.1.3 - dot1dTpFdbStatus
 - 1.3.6.1.2.1.17.4.4 - dot1dTpPortTable (6 more)
 - 1.3.6.1.2.1.17.4.4.1.1 - dot1dTpPort
 - 1.3.6.1.2.1.17.4.4.1.2 - dot1dTpPortMaxInfo
 - 1.3.6.1.2.1.17.4.4.1.3 - dot1dTpPortInFrames
 - 1.3.6.1.2.1.17.4.4.1.4 - dot1dTpPortOutFrames
 - 1.3.6.1.2.1.17.4.4.1.5 - dot1dTpPortInDiscards
- 1.3.6.1.2.1.17.5 - dot1dStatic
 - 1.3.6.1.2.1.17.5.1 - dot1dStaticTable (5 more)
 - 1.3.6.1.2.1.17.5.1.1.1 - dot1dStaticAddress
 - 1.3.6.1.2.1.17.5.1.1.2 - dot1dStaticReceivePort
 - 1.3.6.1.2.1.17.5.1.1.3 - dot1dStaticAllowedToGoTo
 - 1.3.6.1.2.1.17.5.1.1.4 - dot1dStaticStatus

The **dot1dBase** group contains the objects which are applicable to all types of bridges.

The **dot1dStp** group contains the objects that's denote the bridge's state with respect to the Spanning Tree Protocol.

The **dot1dSr** group contains the objects that's describe the entity's state with respect to source route bridging.

The **dot1dTp** group contains the objects that's describe the entity's state with respect to transparent bridging.

The **dot1dStatic** group contains the objects that's describe the entity's state with respect to destination-address filtering.

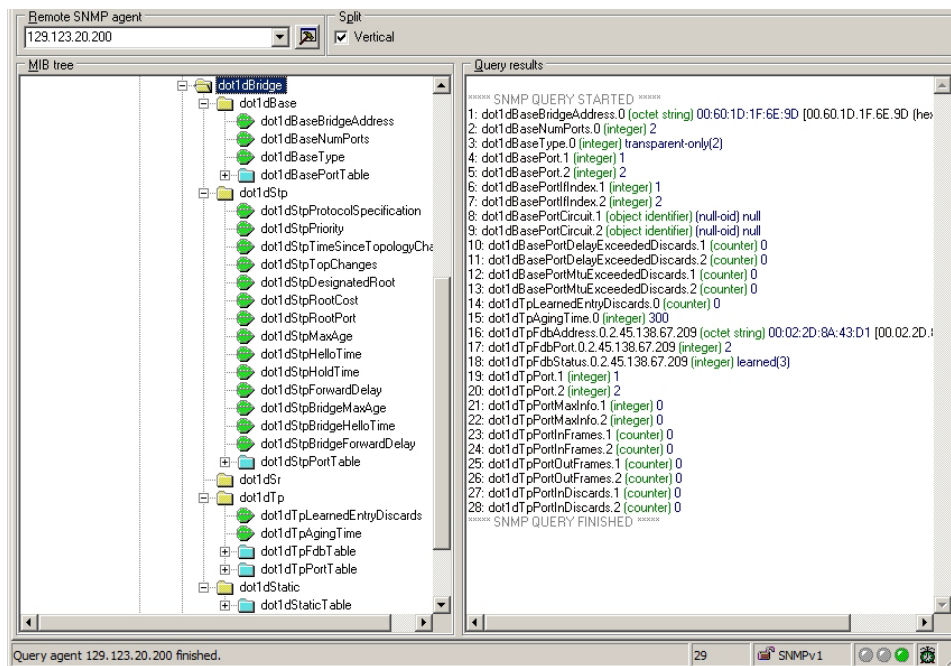


Figure 7.23: The dot1dBridge group

7.5.11 The PRIVATE MIB group

The Orinoco AP-500 extends the standard MIB with a private MIB that defines object for managing and monitoring the Wireless LAN activity; this MIB is property of Lucent Technology Inc. and no public information about it is currently available. Even if we tried to retrieve more details about it, Lucent Orinoco was not able to provide us the required information.

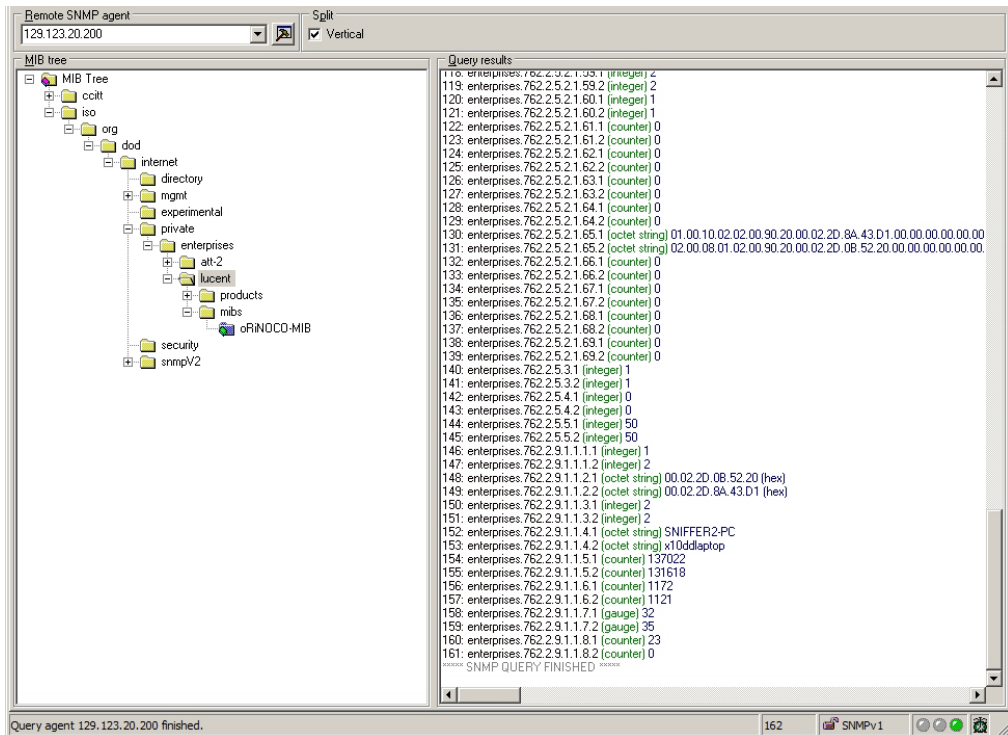


Figure 7.24: The PRIVATE MIB group

The Access Point Manager program retrieves information about the wireless network activity from this part of the MIB; the main group where the AP Manager program gets the values of the OIDs is identified by the following sequence: *1.3.6.1.4.1.762*.

Note: all the snapshots were made using MG-SOFT MIB Browser program (30 days free trial version). They show the real values of the OIDs for the Orinoco AP-500.

Chapter 8

Nortel Networks Contivity 1010 test

8.1 Setup and Configuration

8.1.1 Overview

The Contivity 1010 Gateway is a cost-effective solution delivering secure, comprehensive IP services either in stand-alone mode or in conjunction with an existing router or Internet access device. With a flexible licensing scheme, Contivity can be purchased and installed as an IP access router, IP VPN gateway, or stateful firewall device depending on enterprise need and budget. Supporting up to 30 tunnels, Contivity 1010 is ideal for bringing branch office and partner locations into a secure corporate network.

8.1.2 Devices

Hardware Devices

Here is the list of the used hardware devices:

- 1 desktop PC Fujitsu Siemens SCENIC S ¹
- 1 laptop PC Dell Latitude CPx ²
- 1 sniffer PC TOSHIBA Satellite P20-S303 ³

¹Microsoft Windows 2000 SP4, Intel Pentium III 933MHz, 256MB RAM

²Microsoft Windows 2000 SP4, Intel Pentium III 500MHz, 128MB RAM

³Microsoft Windows XP Professional SP1, Intel Pentium 4 2.66GHz, 512MB RAM

- 1 Pocket PC Dell Axim X5 ⁴
- 1 Nortel Networks Contivity 1010
- 1 Switch AT-FS705LE (5 ports Fast Ethernet Switch)
- 1 Switch AT-8350GB (50 ports Fast Ethernet Switch)
- 1 Lucent AP-500 Access Point
- 2 PCMCIA Wireless Card 802.11b Gold by Lucent
- 1 USB Wireless Adapter 802.11b DWL-120 by D-Link
- 1 Lexmark E323N Wireless Laser Printer ⁵.

Software Programs

Here is the list of used software programs:

- Orinoco AP Manager v2.20, installed on the desktop PC
- Orinoco Client Manager v2.92, installed on the laptop PC
- Nortel IPsec Client v04_65.30, installed on the laptop PC
- movianVPN v3.10 Build 108.39c, installed on the Pocket PC
- Serv-U v4.0 Build 4.0.0.4 (FTP Server), installed on the desktop PC
- FlashFXP v2.1 Build 924 (FTP Client), installed on the laptop PC
- LinkFerret Network Monitor v3.07.0306.0, installed on the sniffer PC

⁴Microsoft Windows CE 4.20, Intel XScale 400MHz, 64MB RAM

⁵with the optional 802.11b interface

8.1.3 Configuration with a Wireless Client

Figure 8.1 shows the front view of the Contivity 1010.

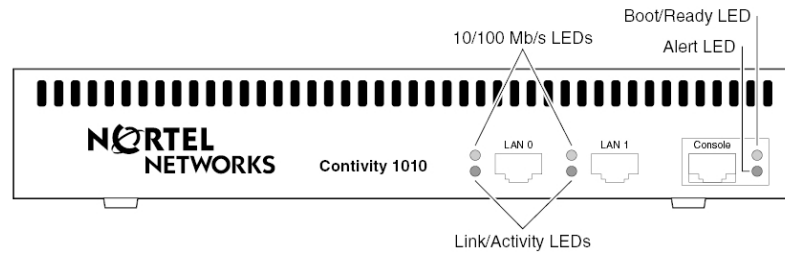


Figure 8.1: The gateway front view

The configuration of figure 8.2 is the one we tried in order to set up correctly the Nortel Contivity 1010 (in the following referred as "gateway").

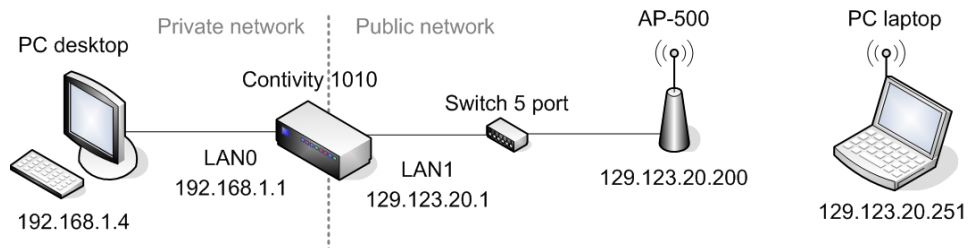


Figure 8.2: The test configuration

First of all we assigned the correct IP address to the LAN0 and LAN1 ports of the gateway: LAN0 is the default interface for the PRIVATE NETWORK while LAN1 is the interface for the PUBLIC NETWORK.

We made this assignment through the serial port using the console interface of the HyperTerminal program as shown in figure 8.3 on page 154.

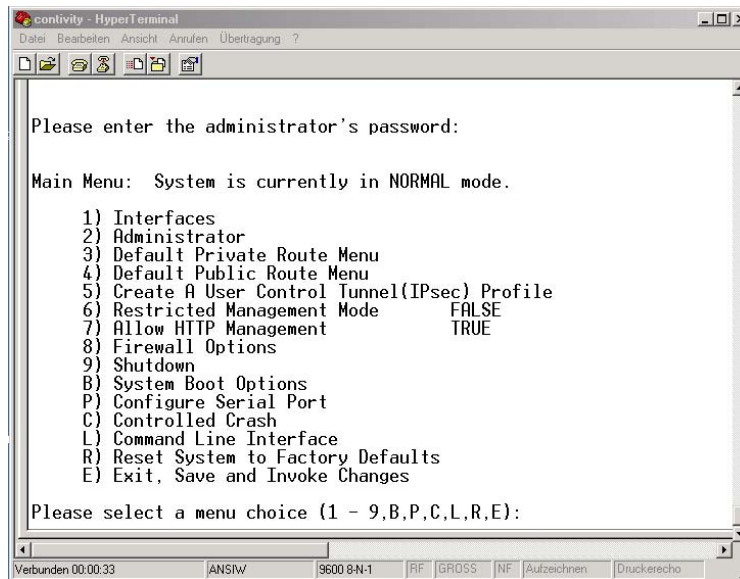


Figure 8.3: The HyperTerminal program manage the gateway via serial

Once setup these ports, we made the further configurations through the Web Management interface of the gateway; in figure 8.4 is presented the welcome screen.

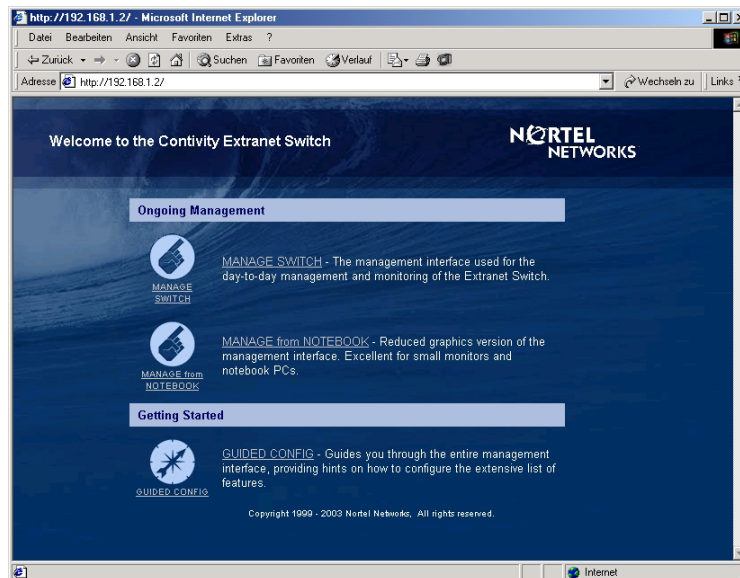


Figure 8.4: The Web Management interface

The gateway associates all remote users with a group, which dictates the attributes that are assigned to a remote user session. The gateway organizes groups in a hierarchical manner; at the top of the hierarchy there's the base group. The base group **\Base** contains the default characteristics that each new group inherits. We add an additional group to the hierarchy named **\Control Tunnel** as child of the base group. In this group we have setup two users (desktop and laptop) to build up the IPsec user tunnel.

Groups are collection of users with the same access attributes and rights. Our two users have identical characteristics, than only one group is necessary. The gateway authenticates each user that tries to connect by checking the user ID and password against a database.

Idle Timeout feature

One of the most useful settings is the *Idle Timeout* feature; it permits to set the amount of time the tunnel remains established even if the connection is lost as set in figure 8.5.

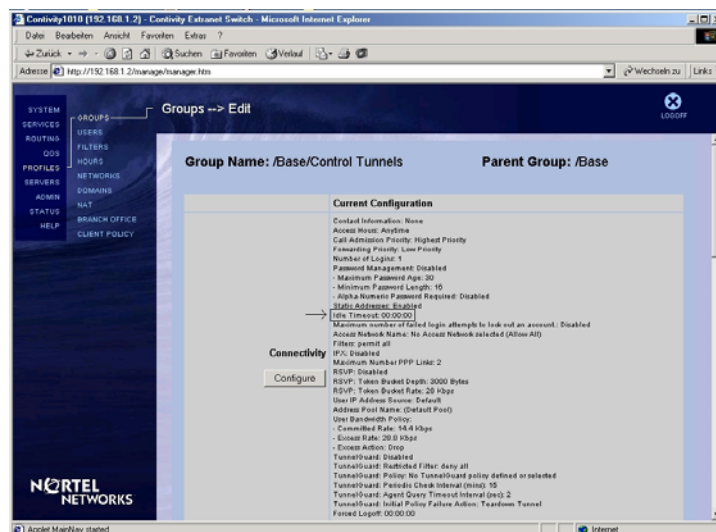


Figure 8.5: The Connectivity property of the `\Base\Control Tunnel` group

This feature is particularly important while operating in wireless environments, where the connection could be easily lost by the client; consequently it's not necessary a tunnel to be established every time.

User Bandwidth Policy feature

Setting up the Connectivity options as shown in previous figure 8.5, we realized that the changes made in the User Bandwidth Policy seems not to take effect. We contacted CMS (our official Nortel reseller) about this problem; they told us that before we can enable this option, we must install an Advanced Routing licence to the Contivity. Otherwise the bandwidth management won't work.

Tunnel Settings

To implement user tunnel we finally must configure tunnelling protocol settings as shown in figure 8.6.

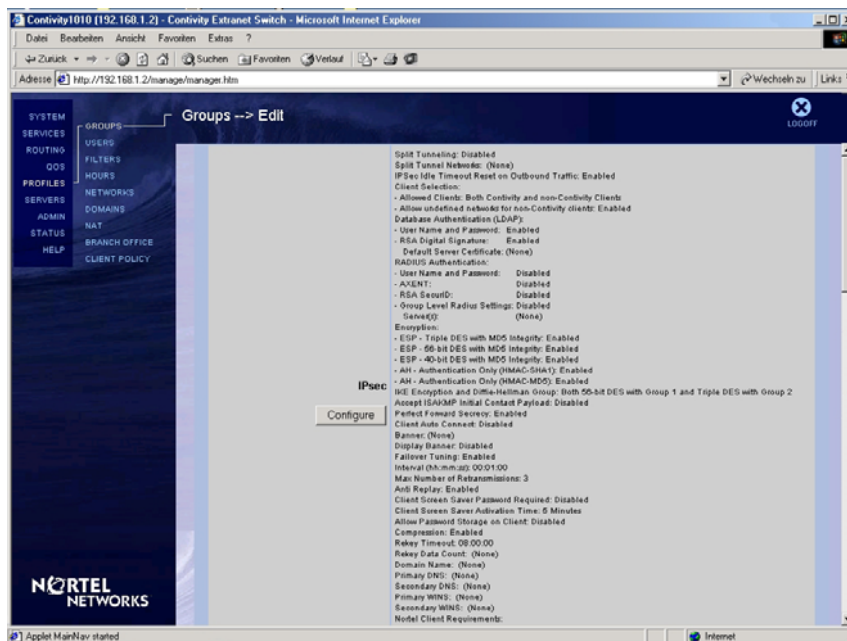


Figure 8.6: The IPsec property of the `\Base\Control Tunnel` group

Contivity Nortel VPN Client installation

Once we finished the gateway configuration, we installed the Nortel IPsec client on both the desktop and the laptop in order to establish an IPsec tunnel through the gateway.

In figure 8.7 it's shown the client mask where to put the *User Name* and *Password* and insert the *IP Destination* gateway.

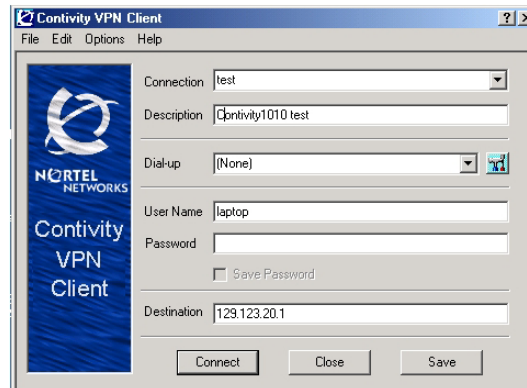


Figure 8.7: The Contivity VPN Client

Both the desktop and the laptop clients must connect to the same *IP Destination*, in our case we chose 129.123.20.1 (i.e. the LAN1 public interface). Once the clients succeed to connect to the gateway, a new virtual IP address is assigned to each PC as we can see in the VPN Client Monitor of figure 8.8.

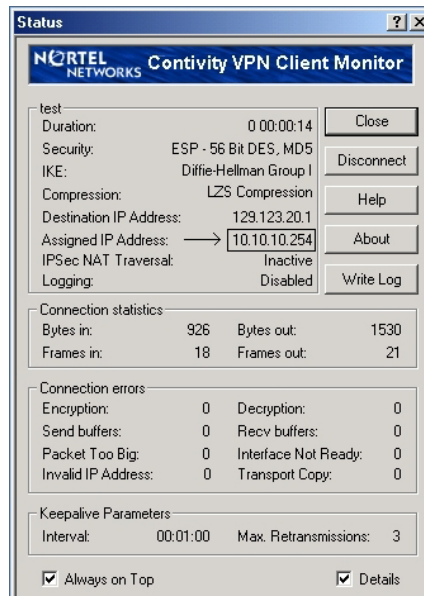


Figure 8.8: The Contivity VPN Client Monitor

Address Pool

Every PC requesting an IPSec connection receives a virtual IP address that is chosen from a so called "pool" of IP addresses set in the gateway by the administrator (us!) as shown in figure 8.9.

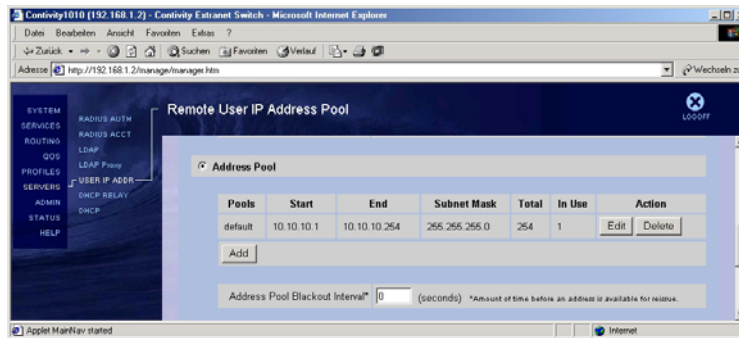


Figure 8.9: The Address Pool assignment window

Backup feature

The gateway gives the administrator the opportunity to backup all the files stored in the built in flash memory (/ide0/). This feature is available through the web interface and requires the presence of an FTP-Server running and reachable on the private network.

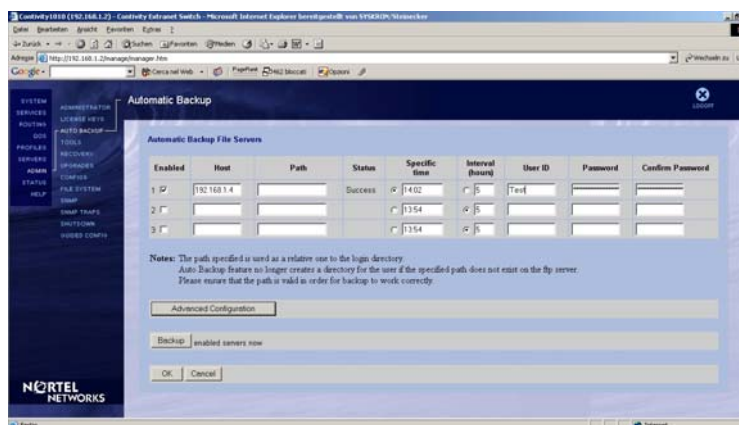


Figure 8.10: The Automatic Backup File Servers window

As shown in figure 8.10, the administrator must specify the IP address of the FTP-Server, giving the related *USER ID* and *PASSWORD*. It's also possible to specify the time interval (in hours) when the automatic backup is supposed to be performed.

When needed, the administrator can upload the backup files and restore a previous configuration enabling the FTP-Server on the gateway; this procedure is possible only through the serial management interface.

Auto Connect feature

One useful feature available in the client is the *Auto Connect*; it enables the VPN client to run like a Windows service at startup so that the mobile device is automatically connected to the VPN network without any further login/password request every time an application requires a connection.

Filtering

It's important to notice that, in order to protect the private network, we must configure the gateway with an appropriate filter, as shown in figure 8.11. Different filter configurations will be used depending on which network activity the administrator decides to allow.

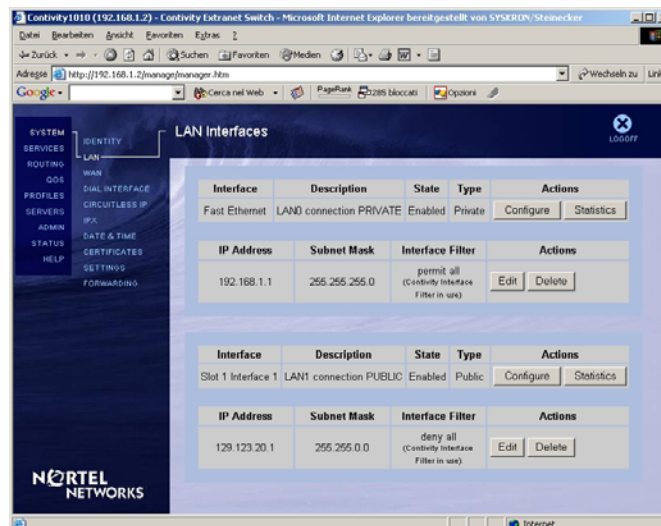


Figure 8.11: The interface LAN1 filters

8.1.4 Configuration with a Pocket PC

We had also the opportunity to test and verify the compatibility of a Pocket PC with IPsec tunneled connections in the configuration shown in figure 8.12. Following the Nortel suggestions, we evaluated the movianVPN client to be installed on the Pocket PC.

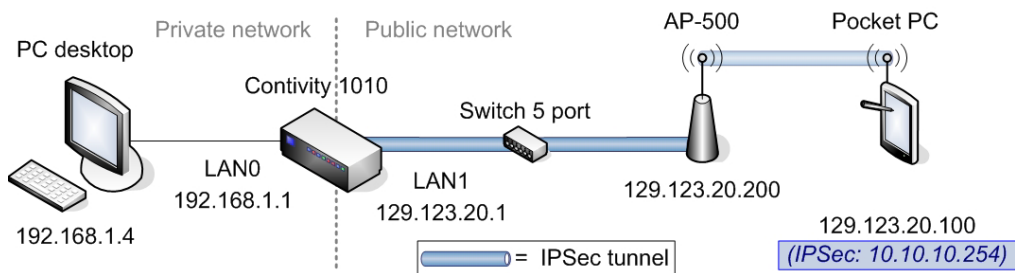


Figure 8.12: The wireless test configuration with Pocket PC

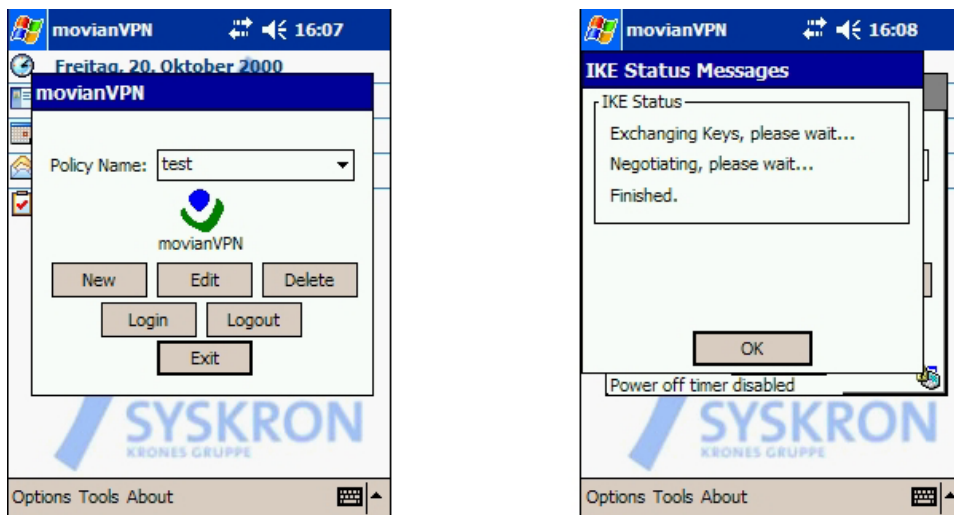


Figure 8.13: The movianVPN client welcome and connection screen

As shown in figure 8.14, the movianVPN client functionality is the same as the one installed on the laptop side.

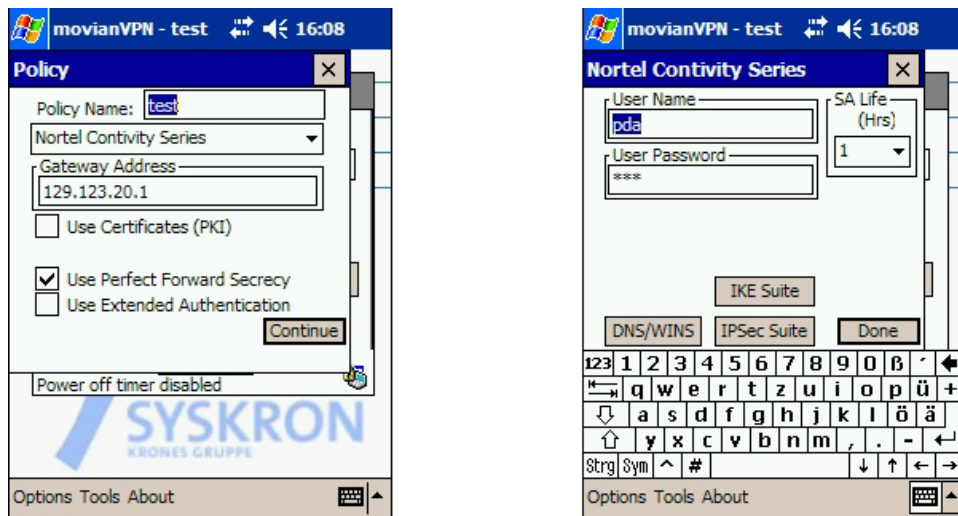


Figure 8.14: The movianVPN client basic settings screens

In order to properly work the IKE and IPsec settings must match the one configured in the Contivity gateway (see figure 8.15)

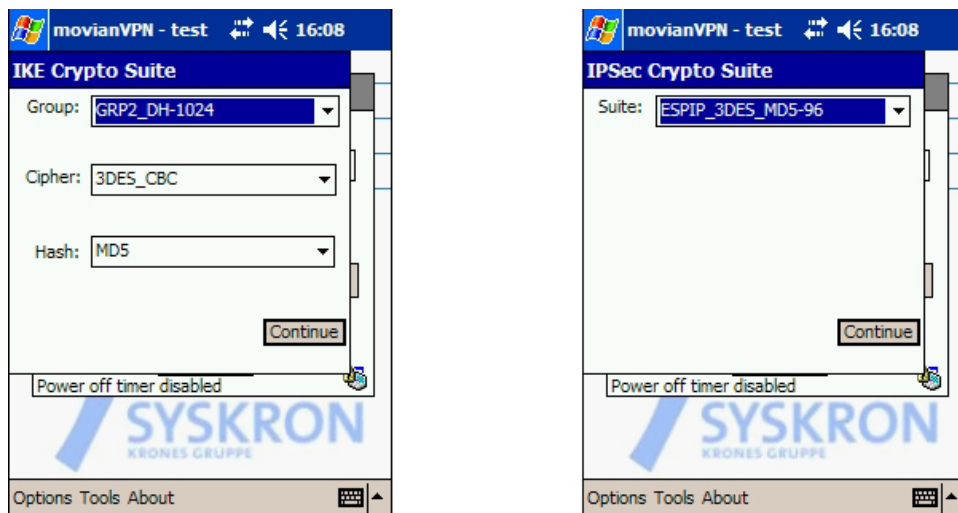


Figure 8.15: The movianVPN client IKE and IPsec settings screens

We have noticed that the *Idle Timeout* option doesn't work properly with the movianVPN client so that every time the wireless connection is lost, the IPsec tunnel must be established again.

8.1.5 Configuration with a Wireless Printer

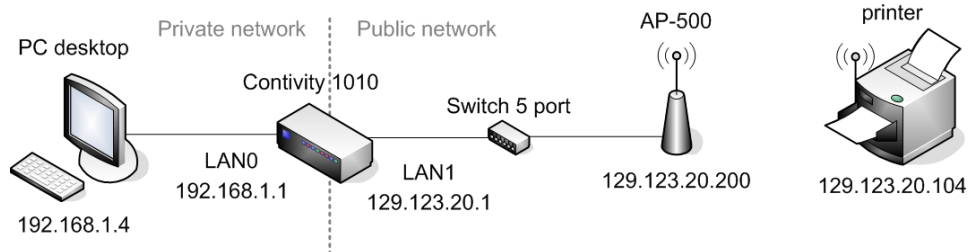


Figure 8.16: The wireless printer configuration test

In figure 8.16 it's shown the configuration used to test the use of a wireless LAN printer in an IPsec environment. In the following section we're going to analyze the impact of its *non-IPSec traffic* on the overall network security.

8.2 Test and Analysis

This section is fully developed in the companion thesis [24] treated by Allegro Matteo. It talks about the test and analysis made on this device in order to get a global overview of the product characteristics and functionalities.

Chapter 9

NCP Secure Communications test

9.1 Setup and Configuration

9.1.1 Overview

NCPs Secure Communications provides a comprehensive portfolio of products for implementing total solutions for high-security remote access. These software-based products comply fully with all current major technology standards for communication and encryption, as defined by the **IETF** (*Internet Engineering Task Force*) and **ITU** (*International Telecommunication Union*). Consequently all products can be smoothly integrated into any existing network and communication architectures; any Internet infrastructure, which may already consist of third-party security and access components, can be further used without changes. The network configuration we took into consideration, is shown in the following figure:

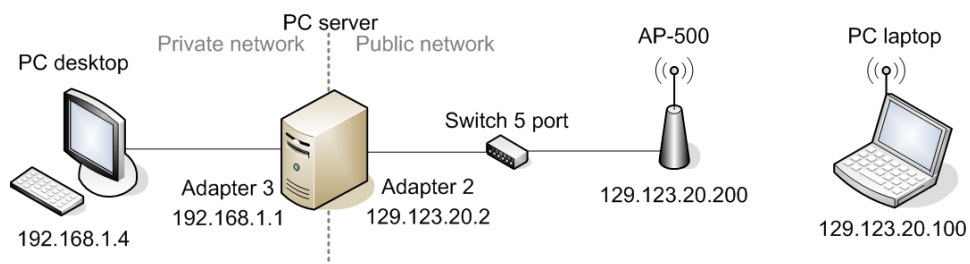


Figure 9.1: The test configuration

NCPs Secure Communications is based on software solutions. All Client

and Server-related products are available for computers based on today's standard operating systems (Windows, Linux, Unix, etc.).

Brief overview of features:

- Flexible access to central site data and IT resources via any public network from any location worldwide
- Intelligent line management for fast and cost effective data communications
- Strong encryption: with key lengths up to 448 bit (symmetric) and 2048 bits (asymmetric)
- Interoperability with all third party products according to international standards
- Powerful administration tools for configuration, deployment, remote control and automatic updating
- Support of all standard VPN tunneling methods: IPSec, L2TP
- Support of VPN tunneling methods optimized for Remote Access: L2Sec, IPSec over L2TP
- Extensive PKI support: X.509 certificates, Smart Cards, Card Readers, Tokens, and CAs.

9.1.2 Devices

Hardware Devices

Here is the list of the used hardware devices:

- 2 desktop PCs Fujitsu Siemens SCENIC S ¹
- 1 laptop PC Dell Latitude CPx ²
- 1 sniffer PC TOSHIBA Satellite P20-S303 ³
- 1 Pocket PC Dell Axim X5 ⁴
- 2 NICs (Network Interface Card) 10/100Mbits installed on 1 server PC

¹Microsoft Windows 2000 SP4, Intel Pentium III 933MHz, 256MB RAM

²Microsoft Windows 2000 SP4, Intel Pentium III 500MHz, 128MB RAM

³Microsoft Windows XP Professional SP1, Intel Pentium 4 2.66GHz, 512MB RAM

⁴Microsoft Windows CE 4.20, Intel XScale 400MHz, 64MB RAM

- 1 Switch AT-FS705LE (5 ports Fast Ethernet Switch)
- 1 Lucent AP-500 Access Point
- 2 PCMCIA Wireless Card 802.11b Gold by Lucent
- 2 USB Wireless Adapter 802.11b DWL-120 by D-Link.

Software Programs

Here is the list of used software programs:

- Orinoco AP Manager v2.20, installed on the desktop PC
- Orinoco Client Manager v2.92, installed on the laptop PC
- NCP Secure Server Manager v6.00 Build 13
- NCP Secure Server v6.0 Build 48
- NCP Secure Client Enterprise v8.00 SP1 Build 3
- NCP Secure Client Monitor v8.0 SP1 Build 3
- NCP Secure Client Manager v8.01 Build 86
- NCP Secure CE Client v2.01
- Serv-U v4.0 (FTP Server) installed on the desktop PC
- FlashFXP v2.1 (FTP Client) installed on the laptop PC
- LinkFerret Network Monitor v3.07.0306.0 installed on the sniffer PC

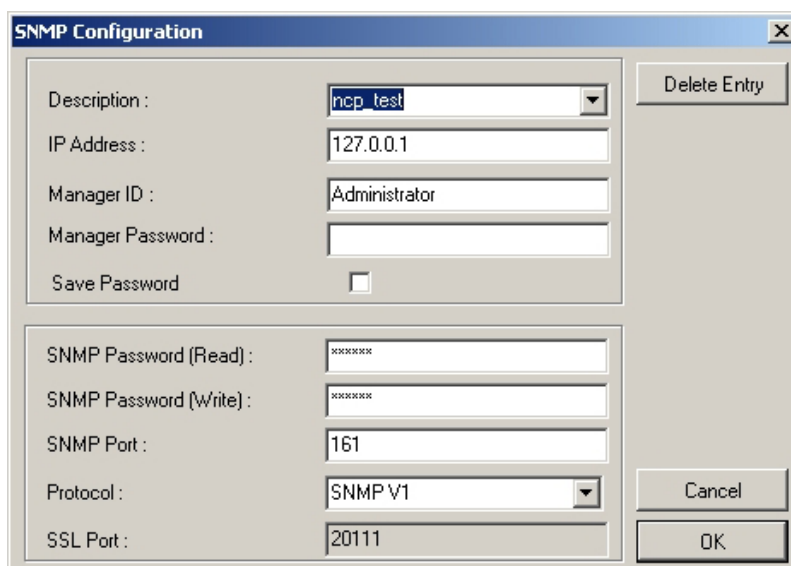
9.1.3 NCP Secure Server Installation

A Setup program performs the installation of the Server Software quickly and smoothly. The only requirement is that the "SNMP service" must be started and running before performing the installation procedure. This is necessary to enable the server configuration through an SNMP based client manager (NCP Secure Server Manager); in fact the server runs completely hidden in background.

9.1.4 NCP Secure Server Configuration

Accessing the server configuration

Once installed the server, the NCP Secure Server Manager must be started in order to configure all the settings. The window shown in figure 9.2 is prompted to the administrator who decides the target server to manage; in fact this program allows also remote configuration. In order to connect to the target computer, the administrator must enter the correct IP address and passwords. In our test, we installed it locally, so we entered the local loop IP address (127.0.0.1); the requested SNMP passwords are the ones that must be set when activating the SNMP service.



The screenshot shows a dialog box titled "SNMP Configuration". It contains the following fields and controls:

- Description: ncp_test (dropdown menu)
- IP Address: 127.0.0.1 (text box)
- Manager ID: Administrator (text box)
- Manager Password: (empty text box)
- Save Password:
- SNMP Password (Read): ***** (text box)
- SNMP Password (Write): ***** (text box)
- SNMP Port: 161 (text box)
- Protocol: SNMP V1 (dropdown menu)
- SSL Port: 20111 (text box)
- Buttons: Delete Entry, Cancel, OK

Figure 9.2: The SNMP Configuration window

Local System configuration

General folder

The "General" folder is used for defining authentication parameters and for setting basic VPN configuration.

The first settings required to fill in were the *LocalUserID* and *LocalPassword*; they are valid for all outgoing calls initiated by the Secure Server when no other link specific ID has been specified in the parameter folder "Link Profiles". This User ID is necessary only for bi-directional authentication.

In our settings, as shown in figure 9.3, the field *Deny incoming call* is disabled; this should be activated prior to or during system maintenance.

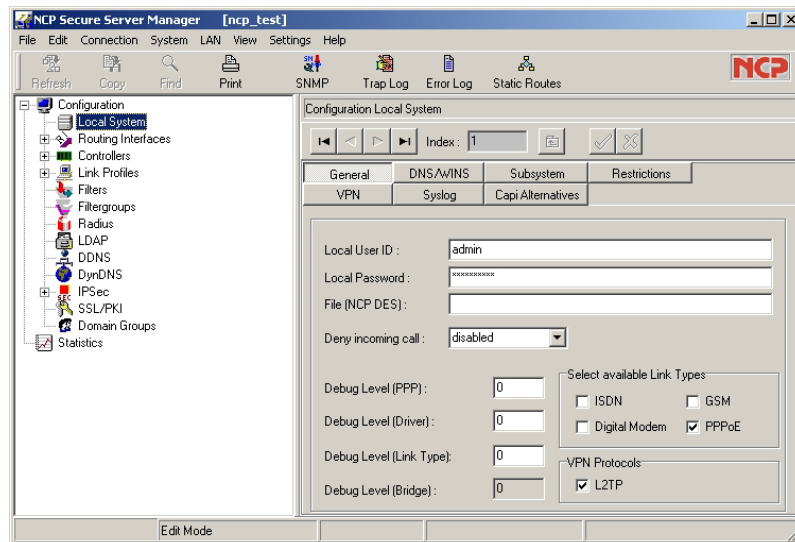


Figure 9.3: The "General" folder

The *Select Available Link Types* feature enables the Link Type, which will be used by the Secure Server for making an outbound connection, to be selected. The corresponding adapter/device and the appropriate drivers must first be installed and loaded. Once selected, the Secure Server loads the respective interface modules. In our settings we checked PPPoE in order to enable the two Ethernet NICs installed on the server PC.

The *VPN Protocols* feature defines which tunneling protocol can be used for outgoing calls. It is essential to select the VPN Protocol in order to build tunnels between the VPN Gateway and the remote destination. A virtual controller is generated for each selected protocol. We checked L2TP (*Layer 2 Tunneling Protocol*) as the default tunneling protocol.

VPN folder

The VPN parameters are important because in our configuration the Secure Server is employed as a VPN Gateway and VPN tunneling is used between the VPN Gateway and the remote destination. In this case the remote destination must also support the same VPN Tunnel protocol.

The *Local Tunnel Endpoint IP Address* field is the official IP Address of the Tunnel End Point in the VPN Gateway. The Client uses this of-

ficial IP Address to access the Secure Server. The IP Address we chose (129.123.20.2) is shown in figure 9.4 and belongs to the public network (129.123.0.0) where the tunnel will be created.

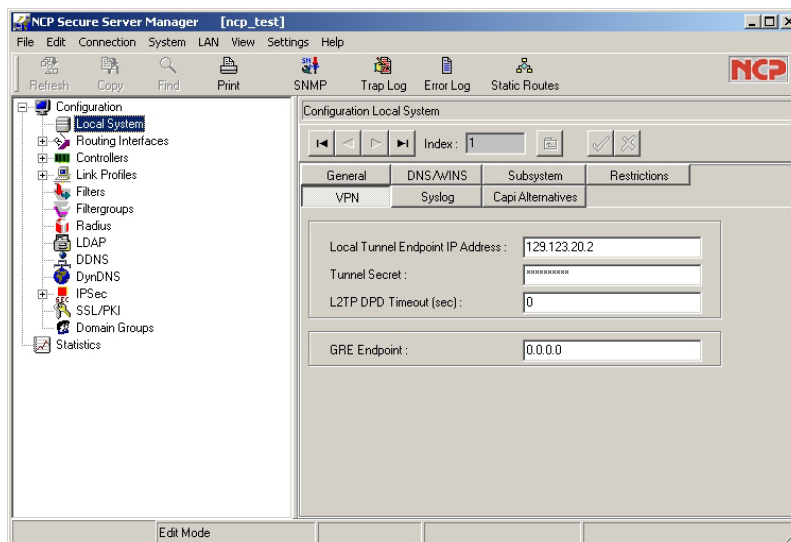


Figure 9.4: The "VPN" folder

The *Tunnel Secret* is a password that is necessary for building a tunnel to a VPN Gateway. The tunnel will only be built when the password set in the VPN Gateway coincides with the password set at remote destination (e.g. Secure Client).

Routing Interfaces configuration

The Routing Interfaces section describes each routing interface that is configured on the Secure Server.

- **Secure Server Adapter**

The Secure Server Adapter is the virtual device created by the Secure Server to manage the tunnel connections. In the "General" folder (figure 9.5) are shown its basic settings (*IP Address*, *IP Netmask*, etc.). The "Pools" folder is used for defining an IP Address Pool for the Secure Clients within the realm of the IP Network. The Secure Server automatically assigns each Client that dials into the system an available IP address from the IP Pool for each session. The ones we used are shown in figure 9.6.

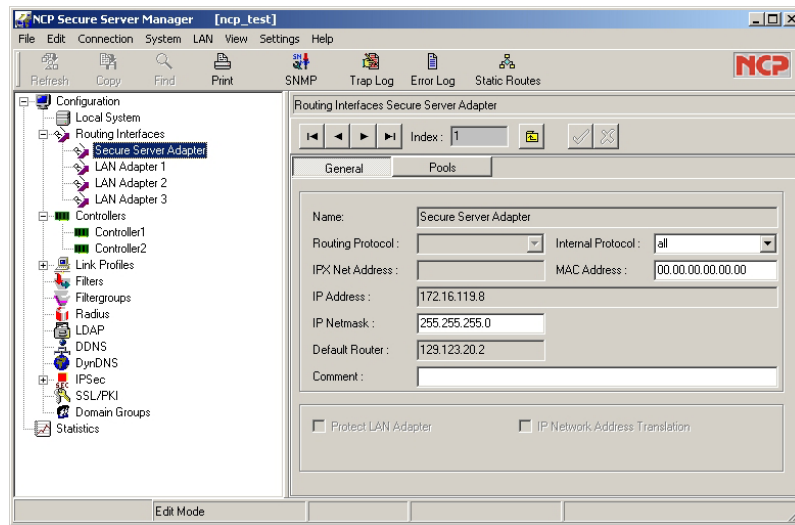


Figure 9.5: The Secure Server Adapter "General" folder

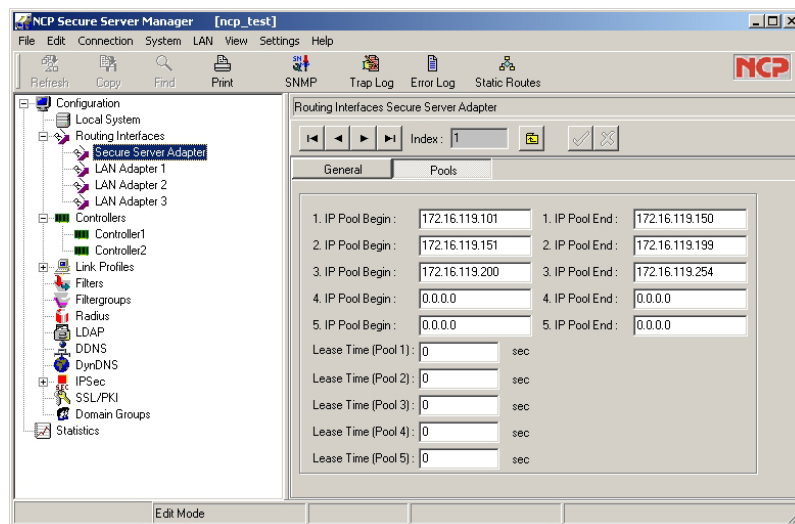


Figure 9.6: The Secure Server Adapter "Pools" folder

- **LAN Adapter 2**

The LAN Adapter 2 is the interface designed for the public network. In the "General" folder (figure 9.7) are shown its basic settings (*IP Address, IP Netmask*, etc.). To secure the private network the **Protect LAN Adapter** must be checked; this setting authorizes only authenticated tunnelled connections refusing all the others.

9.1.4 NCP Secure Server Configuration NCP Secure Communications test

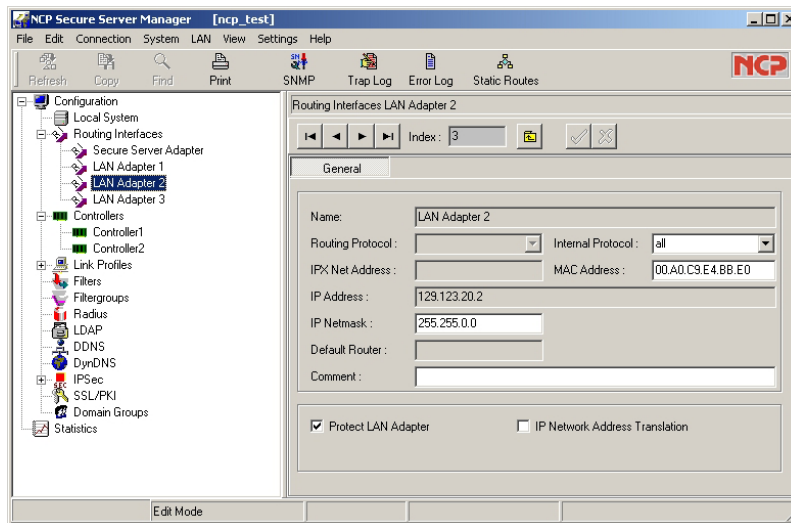


Figure 9.7: The LAN Adapter 2 "General" folder

- **LAN Adapter 3**

The LAN Adapter 3 is the interface designed for the private network. In the "General" folder (figure 9.8) are shown its basic settings (*IP Address, IP Netmask, etc.*).

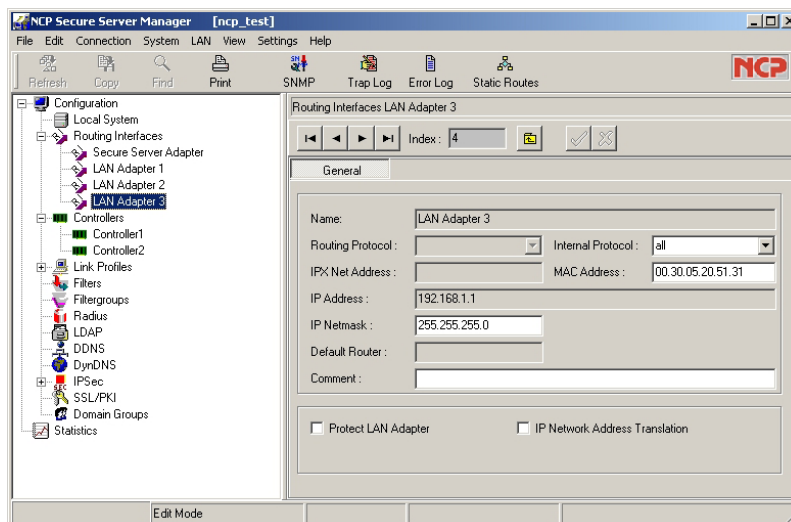


Figure 9.8: The LAN Adapter 3 "General" folder

Link Profiles configuration

This section contains the profiles of all the links the Secure Server can manage. We created a new link profile named *IPsec User* to handle the VPN tunnel connections.

Basic Setting folder

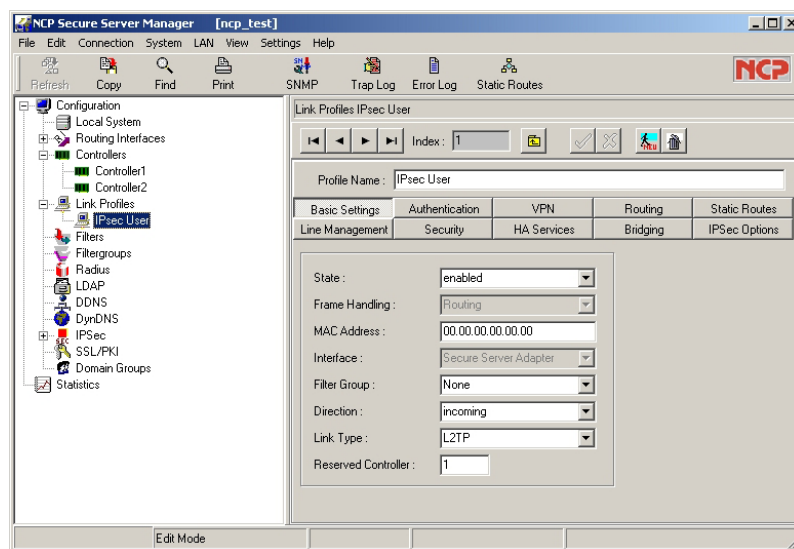


Figure 9.9: The IPsec User "Basic Setting" folder

As shown in figure 9.9, in this folder we **enabled** the *State* field to activate the profile; we set the *Direction* field as **incoming** and the *Link Type* as **L2TP**; finally we reserved the controller n.1 for this user.

Line Management folder

In this folder the only setting we applied during the test was the *Compression* field; we set it to **STAC** or **disable** in order to compare the performances of the tunnelled connection.

Authentication folder

The settings in this folder define the NCP security features for this link

(see figure 9.10). The negotiation is automatic and dependent upon the authentication method supported by the remote destination.

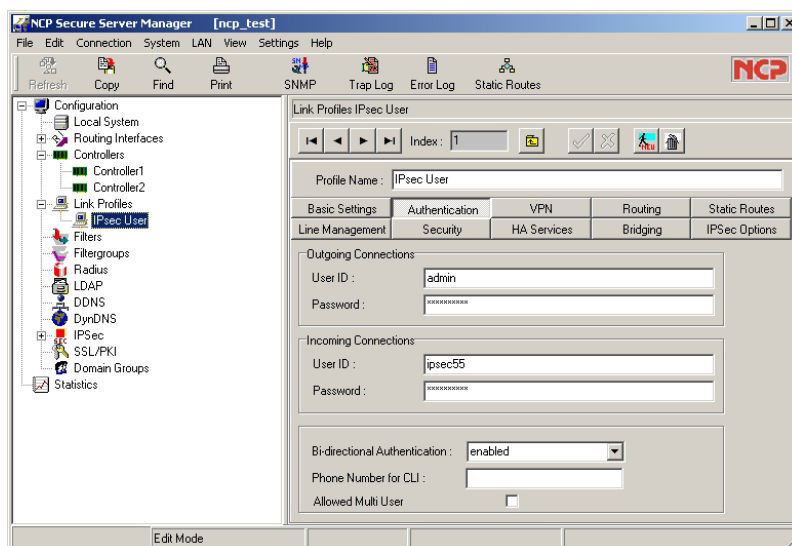


Figure 9.10: The IPsec User "Authentication" folder

The **Outgoing Connections** settings are optional; if no *User ID* and *Password* are specified here, the system will use the ID specified in the "User System" folder under the settings at "Local System, General". This parameter assigns a PPP user ID to the server for calls over this link. If the remote unit is a NCP Client software, this ID must be identical with the ID entered under "Local User ID" in the "Incoming Call" folder.

The **Incoming Connections** settings are absolutely required for incoming calls by using this special link configuration. This parameter assigns a PPP *User ID* and *Password* for this link to the remote destination. If the remote unit wants to connect the Secure Server via this special link, it has to verify itself with this user ID, otherwise no access can be granted. If the remote site is using the NCP Client, this ID has to be identical with the "User-ID" entered in the parameter folder "Dial-Up Network".

Finally we **enabled** the *Bidirectional Authentication* for increasing security and access control. First, the part that tries to establish the connection has to identify itself. After this has been done successfully, the other side has to identify itself as well. The connection is established only after both sides have been successfully authenticated to one another.

The *Allowed Multi User* parameter makes it possible for multiple users to access the VPN Gateway with the same User-ID and password. For security

reasons we didn't check this option.

Security folder

In this folder the configuration parameters pertaining to L2Sec and IPsec are collected for implementation in the remote access environment.

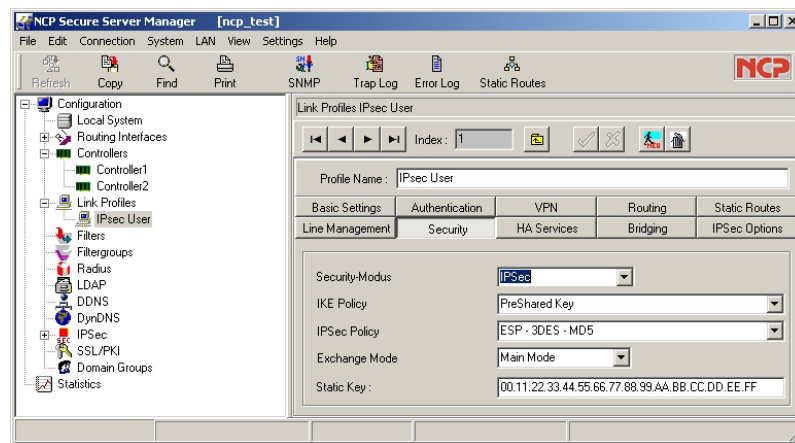


Figure 9.11: The IPsec User "Security" folder

In the *Security-Modus* of figure 9.11 it's possible to determine with which security standard a connection will be permitted, either L2Sec or IPsec. We chose IPsec. Moreover under *IKE Policy* we selected PreShared Key in combination with ESP-3DES-MD5 as configured in the branch "IPsec-IPsec Policy".

The *Exchange Mode* determines how the **IKE** (Internet Key Exchange) should proceed. Two different modes are available; Main Mode also referred to as Identity Protection Mode and the Aggressive Mode. These modes are differentiated by the number of messages and by their encryption. We chose the **Main Mode** as standard setting.

The last step was the setting of the *Static Key* which must be identical to the one entered on the tunnel's other side.

IPsec configuration

The parameters for IPsec are subdivided in three branches of the Configuration tree including the parameter necessary for the configuration of a static Secure Policy Database.

9.1.4 NCP Secure Server Configuration *NCP Secure Communications test*

The policies (IKE / IPSec Policy) are required for every IPSec configuration; the authentication is negotiated between the IPSec initiator and the destination according to the IKE policy. An encrypted control channel is created between them. The IPSec Policy determines how data are to be processed according to IPSec.

The parameters in IKE Policy field relate to Phase 1 of IKE with which the control channel for the SA negotiation was established. Our settings are shown in figure 9.12.

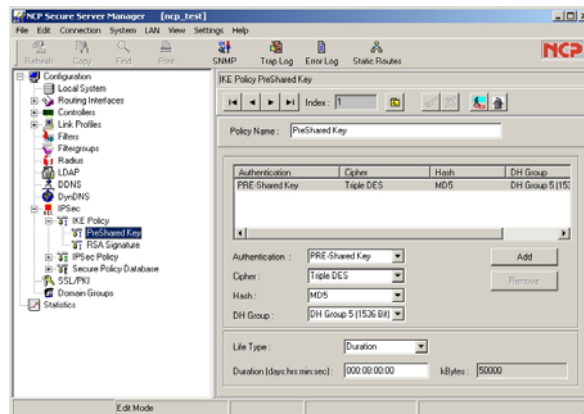


Figure 9.12: The IKE Policy

The parameters in IPSec Policy field relate to Phase 2 of the negotiation. Our settings are shown in figure 9.13.

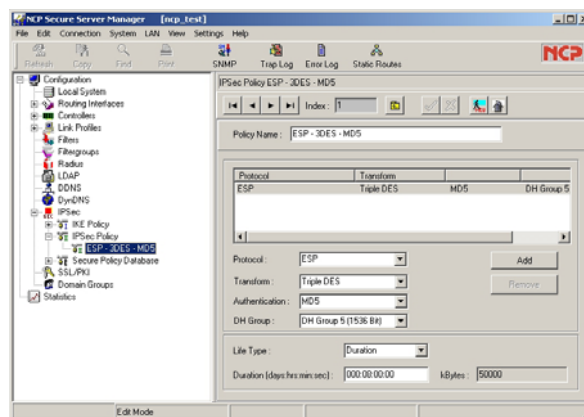


Figure 9.13: The IPSec Policy

9.1.5 NCP Secure Client Configuration

Once installed the Secure Client, the Monitor appears automatically allowing the client configuration.

The Client Monitor serves 4 important features:

- to display the current communications status. This happens in the graphic status window by displaying colored symbols.
- for selection of Link Type. In the phonebook under "Line Management".
- for definition of Call Control parameters. They are defined in the pull-down menu "Configuration → Line Management".
- for definition of phonebook and associated Destination parameters. The settings of the Phonebook are made in the pull-down menu under "Configuration → Phonebook".

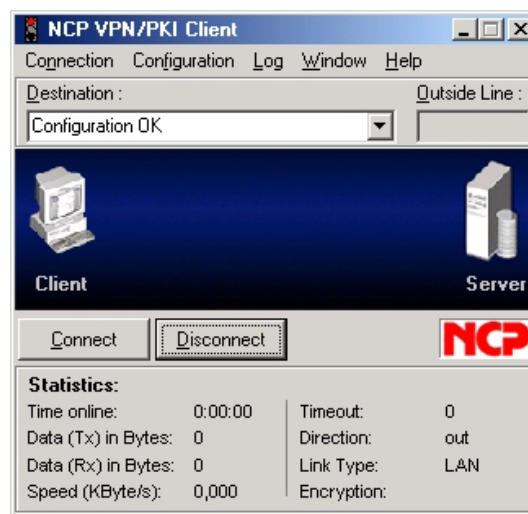


Figure 9.14: The Client Monitor window

When starting the Monitor it's possible to enter a PIN before a connection is established or to insert a Smart Card to enable the secure access.

Phonebook

A very important feature of the Client Monitor is the Phonebook, which provides the basis for defining and configuring Destinations.

Line Management

In the *Line Management* section the most important feature is the **Connection at Boot** (see figure 9.15); it allows the L2TP establishment connection before Windows logon, before the IP stack is loaded. In this way all the communication are encrypted at the very first time, ensuring a safe IPsec key exchange while building up the tunnel.

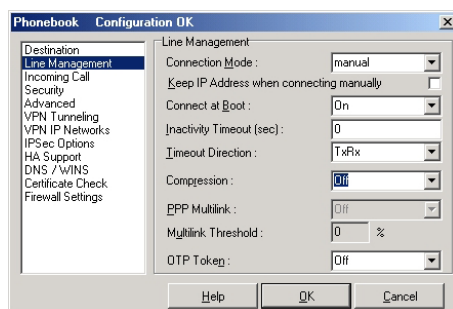


Figure 9.15: The Line Management window

Incoming Calls

In this folder we defined which *Incoming Calls* are permitted and if these should be authenticated (see figure 9.16).

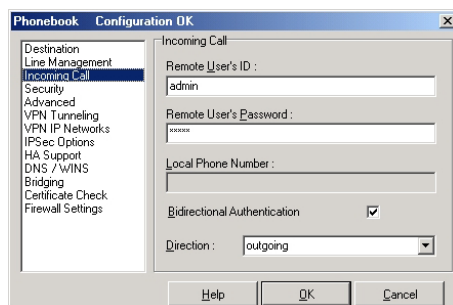


Figure 9.16: The Incoming Calls window

Security

In the parameter field *Security* the configuration parameters pertaining to L2Sec and IPsec are collected for implementation in the remote access environment (see figure 9.17).

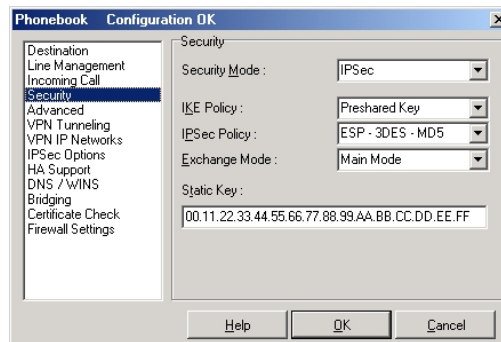


Figure 9.17: The Security window

Advanced

In this field we checked **Permit Incoming IP Traffic** so that it would be possible to initiate an IP connection to the Client (see figure 9.18).

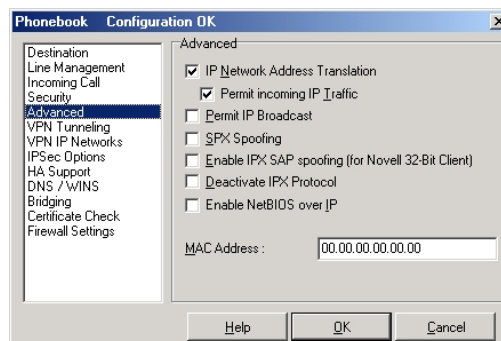


Figure 9.18: The Advanced window

VPN Tunneling

This folder is used to define parameters associated with VPN Tunnelling. These parameters must match and be supported by the Destination that the Client communicates with (see figure 9.19).

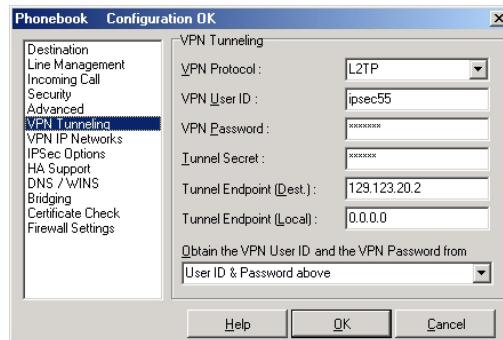


Figure 9.19: The VPN Tunnelling window

Firewall Settings

In this field we chose the *Activation when connected* option so that the PC is not vulnerable if a connection exists. Moreover we checked the *Only communication within the tunnel permitted* field to additionally filter IP packets so that only VPN connections are possible (see figure 9.20).

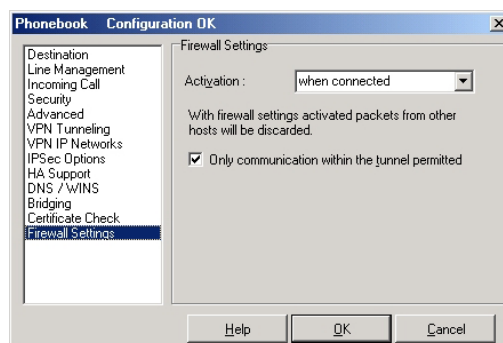


Figure 9.20: The Firewall Settings window

Connection test

When the connection is established (see figure 9.21), the monitor displays a thick green bar from the Client to the Server under which the text *connection is established* is displayed. At the same time, the traffic lights change from red to green. The green traffic light denotes an established connection and occurring costs.

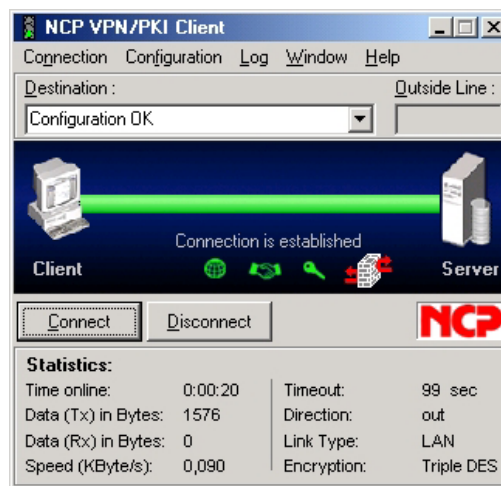


Figure 9.21: The Connection established window

In the Statistics details of figure 9.21, the *timeout* field shows the remaining time till the tunnelled connection remains established even if the mobile client loses the signal. In this way when the mobile station loses the connection, the tunnel remains up and it's not needed a new renegotiation with the server. The timeout value can be setup in the Phonebook settings according to the user's needs.

9.1.6 NCP Secure Client Manager

NCP's Client Manager has the tools for monitoring remote workstations from one central office. The software tool supports display and changes of configuration, display and take-over of log files for error analysis, determination of limits for connection control, as well as display of general information, e.g. software status, CAPI version, operating system, etc. In this way the central office is always up-to-date as to all remote implementations. Thus, updates and elimination of possible faults can quickly be executed. All data transferred between remote PC and central office is, of course, encrypted.

All PC workstations must be pre-configured in an uncomplicated way. NCP's Client Manager contains a comfortable configuration tool that can define all parameters relevant to each workstation (e.g. IP Address, user authorization, user ID, password, presetting of configuration values at the Client Monitor). Particularly handy: If configuration data already exist in databases e.g. as a text file they can be read into the client configuration manager.

For automatic Update/Upgrade purposes the NCP Client Manager includes its own Update Manager, which when applied assures that the Secure Clients always have the most current configuration settings and software release. The entire process is fully automated. Each time the Client connects to the central office the User's profile will be compared with his profile at the central office. Any new items or changes will be automatically downloaded and implemented in the Client.

Remote Monitoring

In order to monitor a remote client, the administrator must create a new entry as shown in figure 9.22.

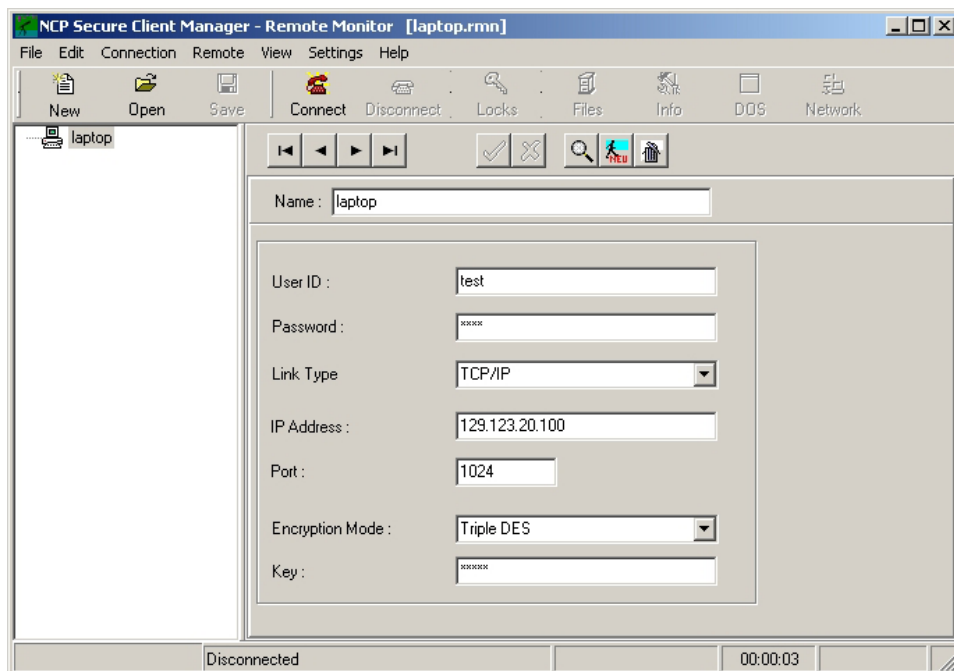


Figure 9.22: The user profile configuration

If the user wants to permit remote access to the client, then he clicks on the menu item "Permit Access" in the NCP Secure Client. Thereafter access is permitted only for a single dial-in procedure from the remote side. This means that once the remote administrator terminates the connection to the client, or if the connection is otherwise terminated (through booting or by turning off the PC), then access is no longer permitted (the check mark disappears). If a renewed remote administration is desired then access must once again be allowed.

If the remote user has "Allowed Access" and the settings shown in figure 9.22 agree, then the administrator can dial-in to the client.

In the Client Monitor the user is directed to the remote access by a graphic (see in figure 9.23 the Remote Administration in yellow). During this time no configuration can be executed on the client. The parameter fields can indeed be opened during this procedure, however they cannot be edited (they are grayed out). The remote user cannot see the actions (mouse movements) the administrator makes.

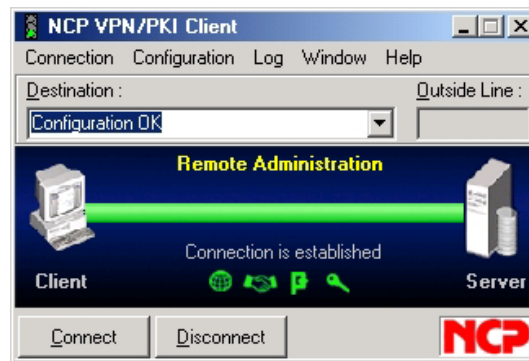


Figure 9.23: The Remote Administration (client side)

On the administrator's side, the Client Monitor is visible as shown in figure 9.24.

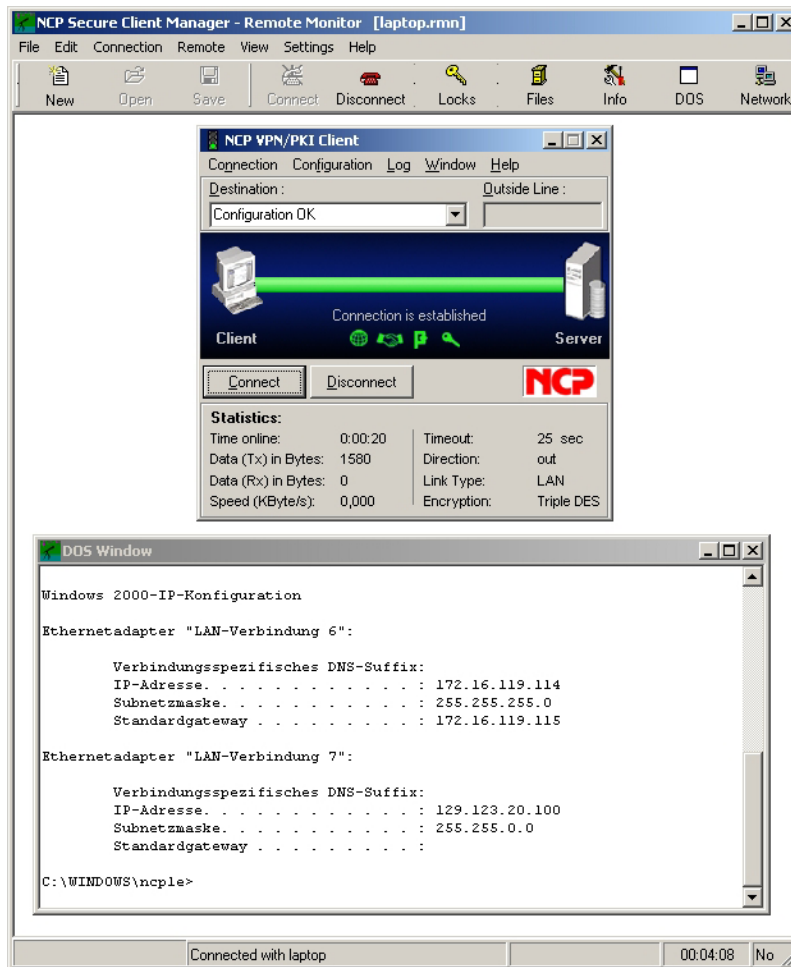


Figure 9.24: The Remote Administration (administrator side) example 1

In this screen shot it's also present a DOS terminal window that's emulate the DOS terminal window on the client side; in this way the administrator is allowed to run any type of command like he was operating locally.

There's also the possibility to upload/download file through this monitor connection as shows in figure 9.25; this is particularly useful when uploading configuration files which can be created through the NCP Secure Client Manager - Configurator.

All these features gives the administrator a complete control over the client station.

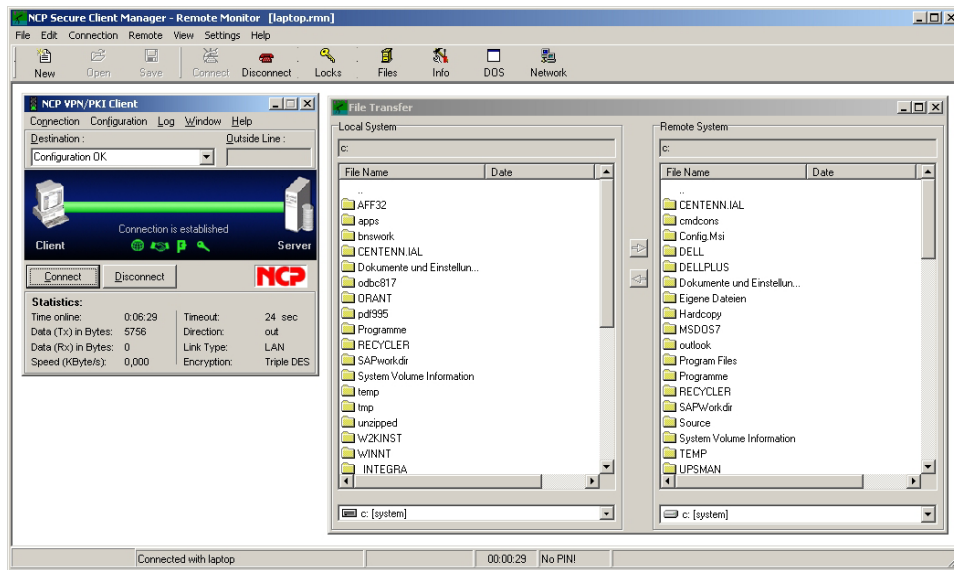


Figure 9.25: The Remote Administration (administrator side) example 2

9.1.7 NCP Secure CE Client

We test also the NCP Secure CE Client installed on the Dell Axim X5. It's almost similar to the standard version of the NCP Secure Client (see figure 9.26).

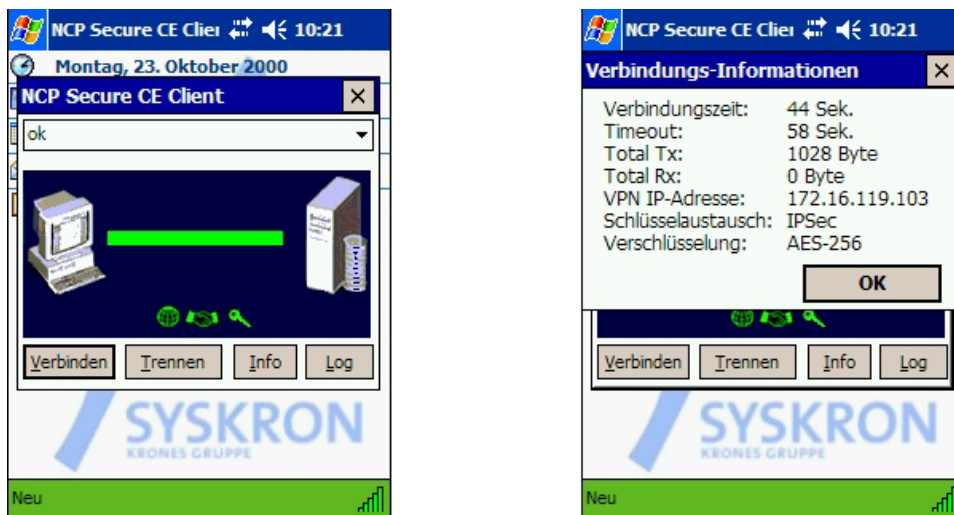


Figure 9.26: The NCP Secure CE Client

The only difference is that the configuration must be prepared with the NCP Secure Client Monitor and then uploaded to the Pocket PC via USB using the cradle (see figure 9.27).



Figure 9.27: The NCP Secure CE Client Monitor

One important feature we noticed is that the Idle Timeout works properly also on the Pocket PC. In this way when the mobile device loses the wireless connection, the L2TP/IPSec tunnel remains established avoiding the user not to renegotiate a new session.

9.2 Test and Analysis

This section is fully developed in the companion thesis [24] treated by Allegro Matteo. It talks about the test and analysis made on this device in order to get a global overview of the product characteristics and functionalities.

Chapter 10

SonicWALL TZ 170 test

10.1 Setup and Configuration

10.1.1 Overview

SonicWALL Internet firewall/VPN security appliances support an array of security applications and deliver powerful firewall and VPN performance. SonicWALL appliances are built on stateful inspection firewall technology, and a dedicated security ASIC designed to ensure maximum performance for VPN enabled applications. With integrated support for firewall, VPN, Anti Virus, content filtering, and an award-winning Global Management System (GMS), IT administrators can trust SonicWALL to protect their network while securely and reliably connecting their remote businesses or personnel.

10.1.2 Devices

Hardware Devices

Here is the list of the used hardware devices:

- 1 desktop PC Fujitsu Siemens SCENIC S ¹
- 1 laptop PC Dell Latitude CPx ²
- 1 sniffer PC TOSHIBA Satellite P20-S303 ³
- 1 SonicWALL TZ 170

¹Microsoft Windows 2000 SP4, Intel Pentium III 933MHz, 256MB RAM

²Microsoft Windows 2000 SP4, Intel Pentium III 500MHz, 128MB RAM

³Microsoft Windows XP Professional SP1, Intel Pentium 4 2.66GHz, 512MB RAM

- 1 Switch AT-FS705LE (5 ports Fast Ethernet Switch)
- 1 Lucent AP-500 Access Point
- 2 PCMCIA Wireless Card 802.11b Gold by Lucent
- 1 Lexmark E323N Wireless Laser Printer ⁴.

Software Programs

Here is the list of used software programs:

- Orinoco AP Manager v2.20, installed on the desktop PC
- Orinoco Client Manager v2.92, installed on the laptop PC
- SonicWALL Global VPN Client v2.0.0.0, installed on the laptop PC
- Serv-U v4.0 Build 4.0.0.4 (FTP Server), installed on the desktop PC
- FlashFXP v2.1 Build 924 (FTP Client), installed on the laptop PC
- LinkFerret Network Monitor v3.07.0306.0, installed on the sniffer PC

⁴with the optional 802.11b interface

10.1.3 Configuration with a Wireless Client

Figure 10.1 shows the front and rear view of the SonicWALL TZ 170.



Figure 10.1: The SonicWALL front and rear view

The configuration of figure 10.2 is the one we tried in order to set up correctly the SonicWALL TZ 170 (in the following referred as "gateway").

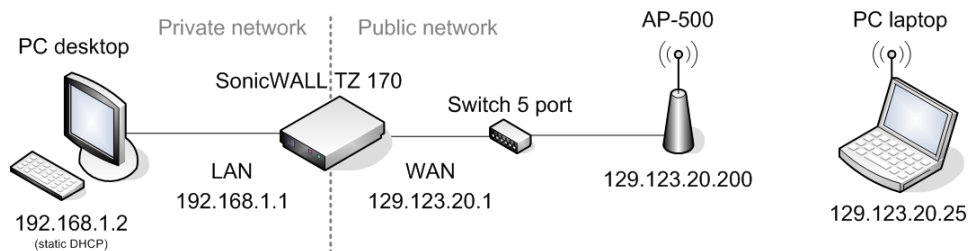


Figure 10.2: The test configuration

Network settings

This device is fully manageable and configurable through the WEB interface; the administrator logs in through the LAN port via HTTPS and he's prompted to insert login and password.

First of all we assigned the correct IP address to the LAN and WAN ports of the gateway; we assigned the following IPs: 192.168.1.1 to the LAN and 129.123.20.1 to the WAN.

LAN is the default interface for the PRIVATE NETWORK, while WAN is the default port for the PUBLIC NETWORK. In order to allow communication between the two different networks, *NAT*⁵ (**Network Address**

⁵One-to-One NAT option enabled

Translation) was enabled (as shown in figure 10.3).

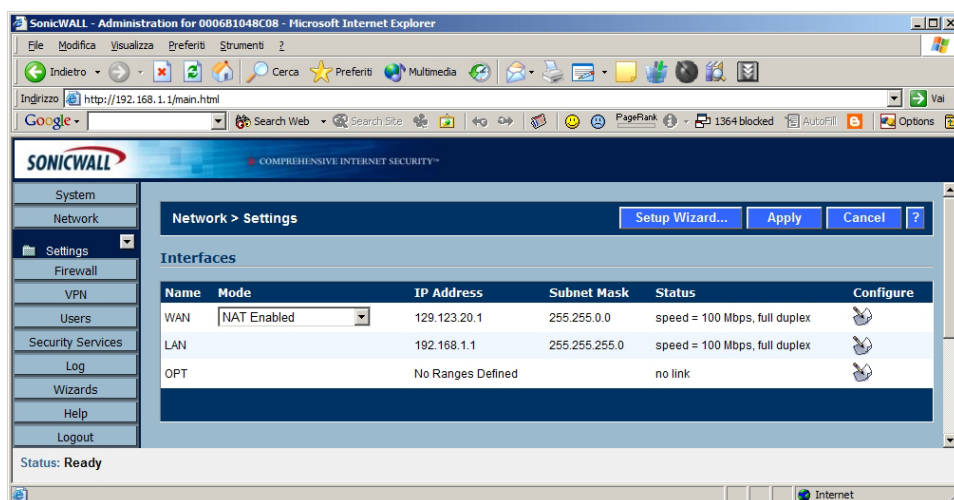


Figure 10.3: The Network settings

Then we proceed enabling the internal DHCP Server under the Network properties so that any computer connected to the gateway in the private network receives dynamically an IP address (the DHCP Server settings are shown in figure 10.4).

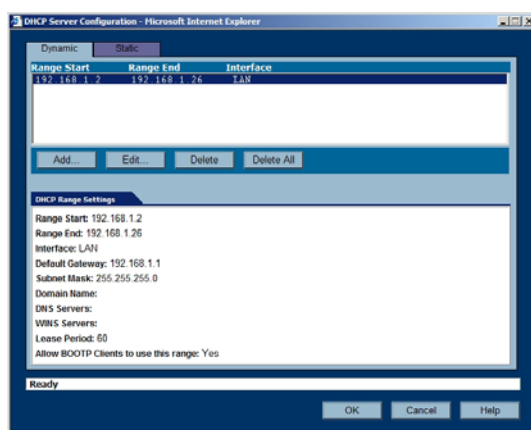


Figure 10.4: The DHCP Server settings

The gateway can be configured as an Intranet firewall to prevent network users from accessing sensitive servers. By default, users on the LAN cannot

access devices connected to the WAN port. To enable access to this area, we must configure the Intranet settings on the gateway as shown in figure 10.5.

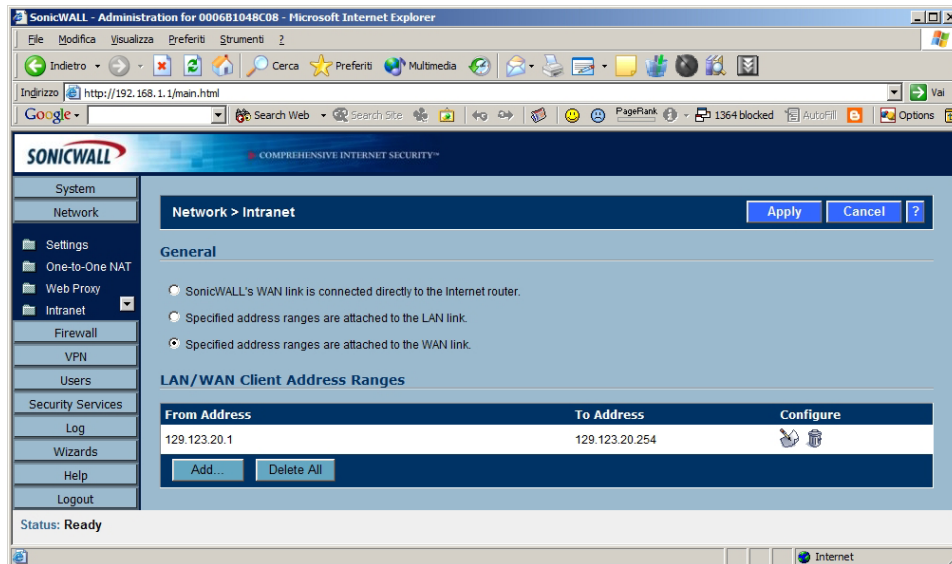


Figure 10.5: The Intranet settings

Firewall settings

Network Access Rules are management tools that allow to define inbound and outbound access policy, configure user authentication, and enable remote management of the gateway. In figure 10.6 are shown the basic rules that permit HTTP management (rule 1 and 2 enabled) and block any non authenticated traffic (rule 3 disabled).

Network Access Rules can be defined by the administrator; these custom rules evaluate network traffic source IP addresses, destination IP addresses, IP protocol types through the SonicWALL Stateful Firewall Engine.

Bandwidth management allows the administrator to assign guaranteed and maximum bandwidth to services and also prioritize the outbound traffic. Bandwidth management only applies to outbound traffic from the gateway to the WAN or any other destination.

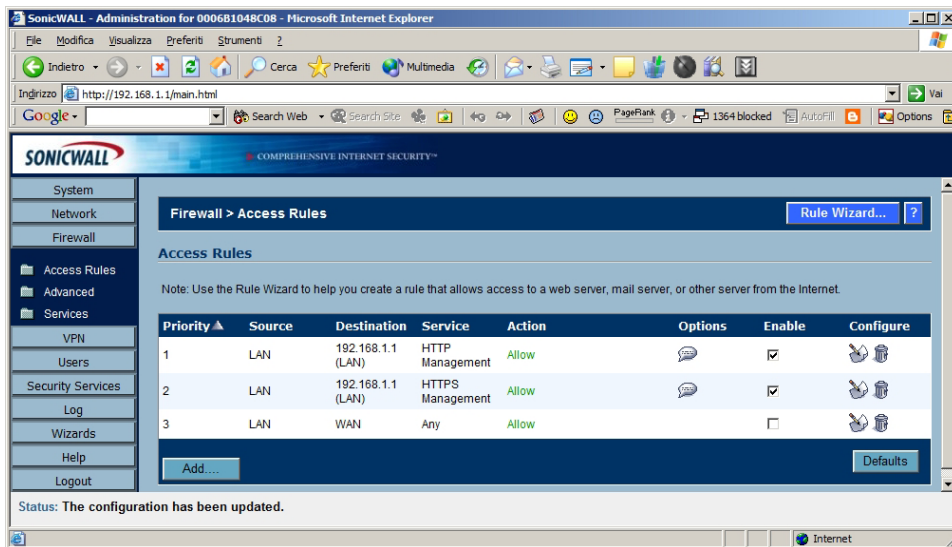


Figure 10.6: The Firewall rules

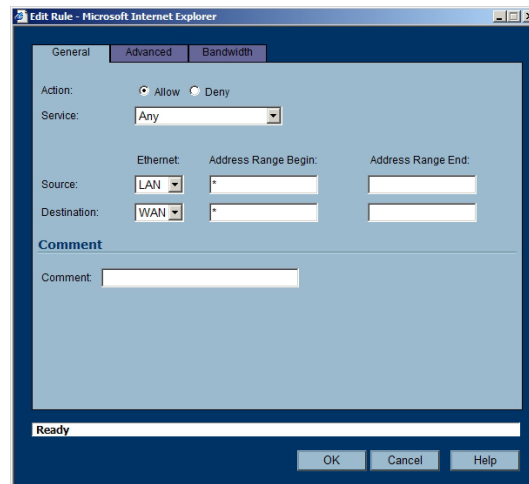


Figure 10.7: The Firewall configuration

VPN settings

SonicWALL VPN provides secure and encrypted communication; using the intuitive Web Management Interface, it's possible to create VPN Security Association to remote hosts.

The gateway automatically encrypts and decrypts data, forwarding the traffic to the intended destination. SonicWALL VPN is based on the industry-standard IPsec VPN implementation.

The Global VPN Settings section shown in figure 10.8 displays the *Unique Firewall Identifier* and *Enable VPN check* that must be selected to allow VPN policies through the gateway.

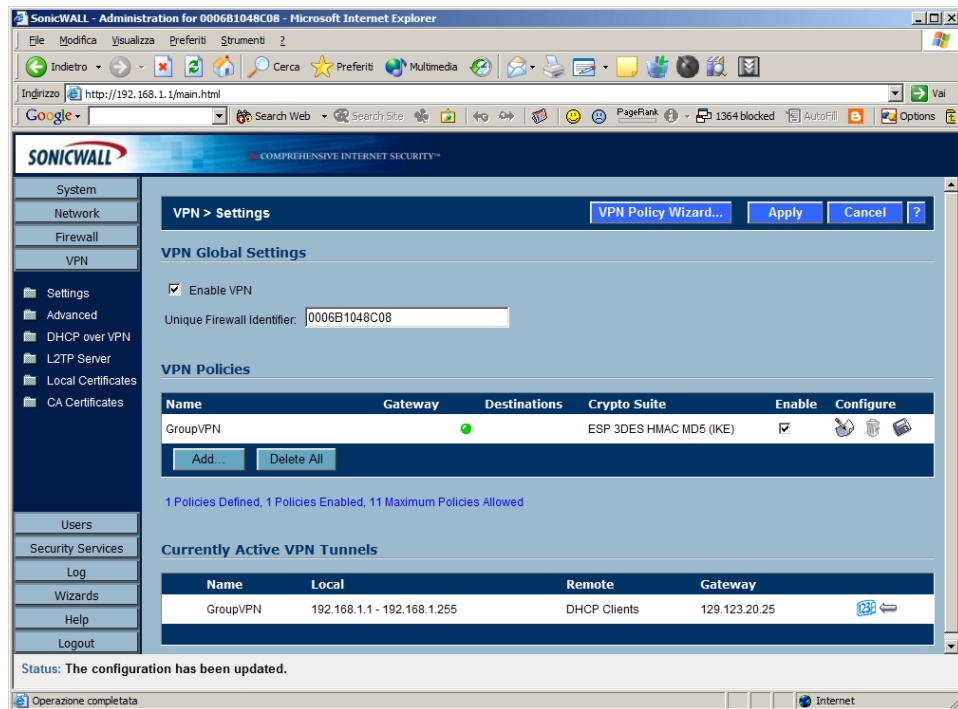


Figure 10.8: The VPN configuration

All existing VPN Security Associations are displayed in the VPN Policy table. Each entry displays the following information:

- Name - user-defined name to identify the Security Association.
- Gateway - the IP address of the remote SonicWALL. If 0.0.0.0 is used, no Gateway is displayed.
- Destinations - the IP addresses of the destination networks.

SonicWALL VPN defaults to a Group VPN setting. This feature facilitates the set up and deployment of multiple VPN clients by the administrator of the gateway. Security settings can now be exported to the remote client and imported into the remote VPN client settings.

Group VPN allows for easy deployment of multiple VPN clients making it unnecessary to individually configure remote VPN clients. Group VPN is only available for VPN clients; its tab settings are shown in figure 10.9.

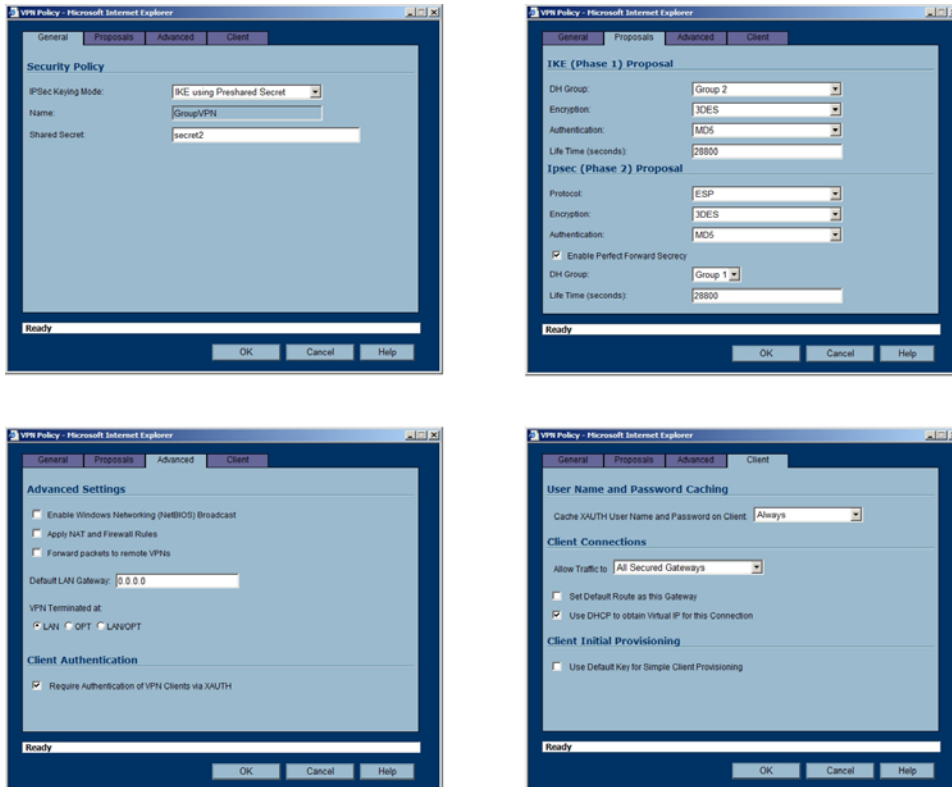


Figure 10.9: The Group VPN tab setting windows

In the Advance VPN settings, there's an important feature: **Enable IKE Dead Peer Detection**. Selecting this option, inactive VPN tunnels can be dropped by the gateway after an idle timeout expires. It works with the use of "heartbeats" in the Dead peer detection and with a "Failure Trigger Level" (missed heartbeats). If the trigger level is reached, the VPN connection is dropped by the gateway, otherwise the tunnelled connection remains established. The SonicWALL uses a UDP packet protected by Phase 1 Encryption as the heartbeat.

DHCP over VPN allows a host (DHCP Client) behind the gateway to obtain an IP address lease from a DHCP server at the other end of a VPN tunnel (the gateway itself acts like a DHCP Server in the LAN network). In order to assign a Virtual IP to the IPsec client host connecting from the WAN network, the "Apply NAT and Firewall Rules" option, under the

Advanced Group VPN policy, must be disabled.

The VPN users must be configured under the Users settings section as shown in figure

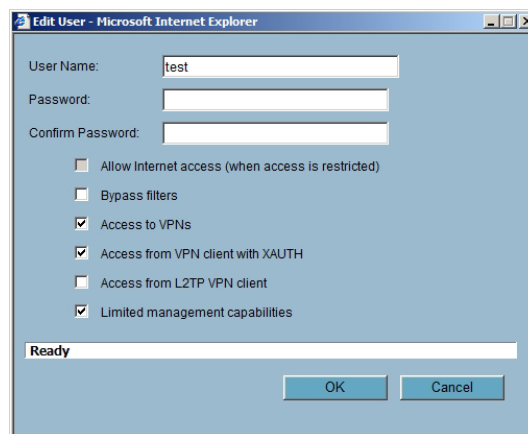


Figure 10.10: The User setting windows

SonicWALL Global VPN Client

The SonicWALL Global VPN Client delivers a robust IPsec VPN solution with these features:

- **Client Policy Provisioning** - Using only the IP address or Fully Qualified Domain Name (FQDN) of the SonicWALL VPN gateway, the VPN configuration data is automatically downloaded from the SonicWALL VPN gateway via a secure IPsec tunnel, removing the burden from the remote user of provisioning VPN connections.
- **Tunnel All Support** - Provides enhanced security by blocking all traffic not directed to the VPN tunnel to prevent Internet attacks from entering the corporate network through a VPN connection.
- **Secure VPN Configuration** - Critical Global VPN Client configuration information is locked from the user to prevent tampering.
- **Automatic Reconnect When Error Occurs** - Allows the Global VPN Client to keep retrying a connection if it encounters a problem connecting to a peer. This feature allows the Global VPN Client to automatically make a connection to a SonicWALL VPN gateway that is temporarily disabled, without manual intervention.

- **Start this program at login** - Launches the SonicWALL Global VPN Client when logging into Windows.

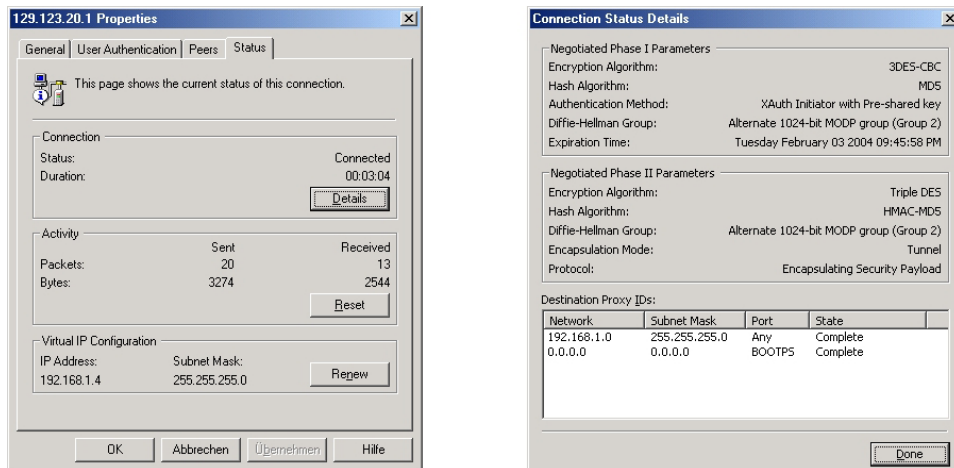


Figure 10.11: The Client "Status" and "Connection Status Details"

Backup/Restore feature

It's possible to export the current configuration of the gateway to a file.

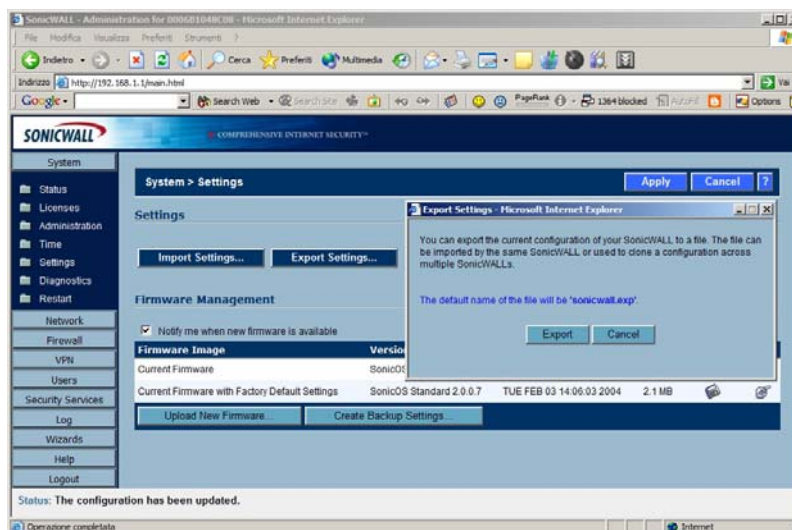


Figure 10.12: The Backup/Restore feature

The file can be easily imported by the same gateway or used to clone a configuration across multiple SonicWALLs. The import/export procedure is shown in figure 10.12.

10.1.4 Configuration with a Pocket PC

We didn't have the possibility to test the SonicWALL TZ 170 with a Pocket PC, so this section is empty.

10.1.5 Configuration with a Wireless Printer

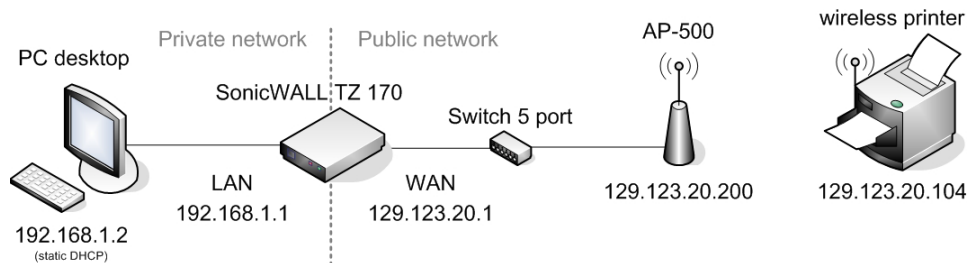


Figure 10.13: The wireless printer configuration test

In figure 10.13 it's shown the configuration used to test the use of a wireless LAN printer in an IPsec environment. In the following section we're going to analyze the impact of its *non-IPSec traffic* on the overall network security.

10.2 Test and Analysis

This section is fully developed in the companion thesis [24] treated by Allegro Matteo. It talks about the test and analysis made on this device in order to get a global overview of the product characteristics and functionalities.

Chapter 11

Conclusions

11.1 Components

In table 11.1 are present the necessary components for each tested solution.

Solution	What do you need?
Contivity 1010	<ul style="list-style-type: none">• 1 Nortel Contivity 1010 box• 1 Nortel IPSec Client (free) for each mobile device• 1 movianVPN client for each PDA
NCP	<ul style="list-style-type: none">• 1 NCP Secure Server on the gateway PC• 1 NCP Secure Server Manager on the gateway PC• 1 NCP Secure Client for each mobile device• 1 NCP Secure CE Client for each PDA
SonicWALL TZ 170	<ul style="list-style-type: none">• 1 SonicWALL TZ 170 box• 1 SonicWALL Global VPN Client (free) for each mobile device (the desired number of concurrent VPN users must be licensed)

Table 11.1: The necessary components

11.2 Best Product Comparison

In the following tables the three tested devices are compared; the most relevant features, highlighted during the tests, have been estimated grouping them in different categories.

Gateway Features	Vote	Description
Contivity 1010	+	the standard gateway has only basic features; additional options can be licensed later
NCP	++	fully configurable with advanced features; available remote management
SonicWALL TZ 170	++	fully configurable with advanced features

Management interface	Vote	Description
Contivity 1010	–	the WEB interface is functional but really slow. To access some sections you need the complete JVM (Java Virtual Machine), making the management really heavy
NCP	++	complete management through SNMP, granting the best and the fastest interface ever test
SonicWALL TZ 170	+	complete management through HTTPS, granting an intuitive and fast interface

Client features	Vote	Description
Contivity 1010	--	the free client embedded with the box has only login options without any firewall settings
NCP	++	fully configurable with advanced features (firewall); available remote management
SonicWALL TZ 170	+	the client embedded with the box has only login options since all the VPN settings are stored in the gateway; firewall rules are also available

Performance with IPSec	Vote	Description
Contivity 1010	+	the benchmark test show excellent throughput performance with a low encryption
NCP	+	the benchmark test show excellent throughput performance with the highest encryption
SonicWALL TZ 170	+	the benchmark test show excellent throughput performance with high encryption

Tunnel Idle Timeout	Vote	Description
Contivity 1010	+	this important feature is available
NCP	+	this important feature is available
SonicWALL TZ 170	+	this important feature is available

Startup	Vote	Description
Contivity 1010	+	this solution provides the automatic VPN client startup and tunnel establishment at windows logon
NCP	++	this solution provides the automatic VPN client startup and tunnel establishment at windows boot
SonicWALL TZ 170	+	this solution provides the automatic VPN client startup and tunnel establishment at windows logon

Encryption	Vote	Description
Contivity 1010	-	with the free client is possible to use only 56bit DES under IPSec tunnel (L2TP not supported by the free client)
NCP	++	provides the highest encryption level with IPSec over L2TP
SonicWALL TZ 170	+	provides an high encryption level with IPSec (L2TP not supported)

Firewall	Vote	Description
Contivity 1010	<i>n.a.</i>	not present in the demo box version
NCP	+	stateful firewall available
SonicWALL TZ 170	+	stateful firewall available

Customer support	Vote	Description
Contivity 1010	+	normally supported by CMS
NCP	+	normally supported by APE
SonicWALL TZ 170	+++	even if the registration procedure is quite annoying (compulsory internet registration), the WEB support we tried was excellent: in fact it provides live chat interaction with company experts

Price	Vote	Description
Contivity 1010	++	the solution with 25 VPN tunnels costs: 1954 €
NCP	---	the solution with 25 VPN tunnels costs: 12561 €. The highest price of the 3 solutions
SonicWALL TZ 170	++	the solution with 25 VPN tunnels costs about: 1800 €

Final considerations	Description
Contivity 1010	This solution is the cheapest one since it provides free VPN clients; although it doesn't offer many features and sufficient security level
NCP	This solution is the most expensive one; although it offers the highest security level and the highest number of features.
SonicWALL TZ 170	This solution has the same price of the Nortel's one, but it offers approximately the same NCP's number of features. In our opinion, it's the Best Choice for the Project.

11.3 Final Considerations

After having analyzed the network topology and having found the ideal structure for the planned usage, we tested this environment using three different solutions that assure a wireless secure communication system.

While, at first glance, the Contivity solution seems to be the optimal choice, after a deep analysis it shows some significant lacks. For example the slow management interface makes its setup and configuration annoying; moreover the embedded client presents only basic features without the integration of a firewall system. In this way the client PC is still vulnerable even if it belongs to the VPN, since it also allows incoming non tunneled traffic.

In our opinion this weakness heavily affects the overall security since it compromises the global network secure communication; for this reason we discouraged the employment of this solution.

NCP provides the most efficient solution, regarding both the gateway management and the encryption level. The most appreciable feature is the possibility to use L2TP in conjunction with IPSec, granting an excellent protection level; moreover this system is fully remote configurable by the administrator through SNMP. Even if this software suite offers superlative performances (probably beyond the company needs), its extremely high price discourages the usage of this solution. We have no doubt that it would have been our *Best Choice for the Project* but it wasn't chosen by the company just because of its high price.

Finally, our choice was the SonicWALL. It grants a reasonable tradeoff between security and price. In fact the security level provided is adequate to the company needs since it offers client with firewall settings and an high IPSec encryption. Moreover its price satisfies the company request to keep costs low since it's affordable for their customers.

In conclusion, we believe that every company implementing the IEEE 802.11 protocol should seriously take into consideration the WLAN security flaws; in fact the WEP standard doesn't offer any security warranty.

For this reason, our thesis proposes a valid and concrete security solution that can be easily employed in an industrial environment.

And remember:

wireless's greatest strengths, in many ways, are also its greatest weaknesses... so don't be weak!

Bibliography

- [1] Stallings, W. *Wireless Communications And Networks*, Prentice Hall, 2002, ISBN 0-13-040864-6, Chapter 14.
- [2] Kent, S., and Atkinson, R. *Security architecture for the Internet protocol*, RFC 2401, Internet Engineering Task Force, November 1998.
- [3] L.M.S.C. of the IEEE Computer Society. *Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, IEEE Standard 802.11, 1999 Edition (1999).
- [4] Postel, J., and Reynolds, J.K. *Standard for the transmission of IP datagrams over IEEE 802 networks*, RFC 1042, Internet Engineering Task Force, February 1988.
- [5] Vocal Technologies Ltd. *IEEE 802.11b White Paper*.
- [6] IEEE Standard 802. *IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture*. 1990, ISBN 1-55937-052-1.
- [7] IEEE Standard 802.11b. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (2.4GHz)*, 1999.
- [8] Wireless Ethernet Compatibility Alliance (WECA).
IEEE 802.11b Wireless Equivalent Privacy (WEP) Security, February 19, 2001.
- [9] Walker, J.R. *Unsafe at Any Key Size; An Analysis of the WEP Encapsulation*, IEEE Document 801.11-01/362.
- [10] Fluhrer, S., Mantin, I., and Shamir, A. *Weaknesses in the Key Scheduling Algorithm of RC4, Eighth Annual Workshop on Selected Area in Cryptography*, August 2001.

- [11] Stubblefield, A., Ioannidis, J., and Rubin, A. *Using the Fluhrer, Mantin, and Shamir Attack to break WEP, Revision 2*, AT&T Laboratories Technical Report TD-4ZCPZZ, August 21, 2001.
- [12] Borisov, N., Goldberg, I., and Wagner, D. *Intercepting Mobile Communications: The Insecurity of 802.11*. MOBICOM 2001 (2001).
- [13] McCloghrie, K., and Rose, M. *RFC 1213 - Management Information Base for Network Management of TCP/IP based internets: MIB-II*, March 1991.
- [14] Kastenholz, F. *RFC 1398 - Definitions of Managed Objects for the Ethernet-Like Interface Types*, January 1993.
- [15] Decker, E., Langille, P., Rijssinghani, A., and McCloghrie, K. *RFC 1493 - Definitions of Managed Objects for Bridges*, July 1993.
- [16] Kent, S. and Atkinson, R. *RFC 2401 - Security Architecture for the Internet Protocol*, November 1998.
- [17] Kent, S. and Atkinson, R. *RFC 2402 - IP Authentication Header*, November 1998.
- [18] Kent, S. and Atkinson, R. *RFC 2406 - IP Encapsulating Security Payload (ESP)*, November 1998.
- [19] Harkins, D. and Carrel, D. *RFC 2409 - The Internet Key Exchange (IKE)*, November 1998.
- [20] Srisuresh, P. *RFC 2888 - Secure Remote Access with L2TP*, August 2000.
- [21] Boden-Cummings, C. and Viney, S. *QuinetiQ White Paper - IPsec Over WLAN: Residual Vulnerabilities*, 2002.
- [22] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G. and Palter, B., *RFC 2661 - Layer two Tunneling Protocol "L2TP"*, August 1999.
- [23] Chokhani, S., Ford, W., Sabett, R., Merrill, C. and Wu, S. *RFC 2527 - Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, November 2003.
- [24] Allegro Matteo, *Monitoring, Management and Test of an IPsec based Industrial WLAN*, companion thesis, April 2004, Padova.