

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

**INFORMATION SECURITY AND WIRELESS
NETWORKS**

**Graduation Project
COM – 400**

Student: İbrahim Aköl

Supervisor: Prof. Dr Fahreddin M. Sadıkoğlu

Nicosia - 2004

"I would like to thank my supervisor Prof. DR. Fahreddin M. Sadıkoğlu for his advices and support in my work and myself during the preparation of this graduation project."

ABSTRACT

Everyday, round the globe, many computer networks and hosts are being broken into and are being compromised by the hackers. As technology continues to modify the ways in which information of all type is stored, analyzed, and exchanged, concerns related to privacy and information security is growing. This project describes how hackers break into systems and manage to compromise the systems, ways to secure our systems from attackers and also various protection/encryption techniques. I discussed all of those in detail with examples where necessary.

The number of computing and telecommunications devices is increasing and consequently, the focus on how to connect them to each other. The cable solution is often complicated since it may require a cable specific to the devices being connected as well as configuration software. To solve these problems wireless technology has been developed. With wireless devices, users will be able to connect a wide range of computing and telecommunications devices easily and simply, without the need for connecting cables. The second part of this project describes two wireless technologies; Bluetooth and IEEE 802.11 standards, and also security concerns of wireless devices.

THE ONLY SAFE COMPUTER IS A DISCONNECTED OR DEAD ONE.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	I
ABSTRACT	II
SAFE COMPUTER	III
TABLE OF CONTENTS	IV
INTRODUCTION	1
CHAPTER ONE: INFORMATION SECURITY	
• PART ONE: INTRODUCTION	2
1.1 Introduction	2
1.2 Why networks must be secured?	2
1.3 Why people try to attack?	4
• PART TWO: CRYPTOGRAPHY & ENCRYPTION	7
2.1 Introduction	7
2.2 Conventional encryption	8
2.3 Public-key encryption	10
2.4 Digital Signatures	12
2.5 Reasons for using digital signatures	13
2.6 The process of checking the validity of digital signatures	14
• PART THREE: FIREWALLS	15
3.1 What does it do?	15
3.2 Packet filter firewalls	16
3.3 Application proxy firewalls	18
3.4 Packet inspection firewalls	19
3.5 Stateful multilayer inspection firewall	20
3.6 Hardware vs. Software firewalls	21
3.7 Do firewalls provide enough security?	22
• PART FOUR: PGP	24
4.1 What is PGP?	24
4.2 How does PGP work?	24
4.3 Hash Functions	27
4.4 Keys	28
• PART FIVE: SSL	29
5.1 What is SSL?	29
5.2 SSL objectives and architecture	29
5.3 SSL session and connection	31
5.4 How does it work?	32
5.5 The SSL record protocol	33
5.6 SSL handshake protocol	36
• PART SIX: KERBEROS	42
6.1 What is Kerberos?	42
6.2 How it works?	43
6.3 Problems with Kerberos	48

• PART SEVEN: IP SECURITY	50
7.1 Goals of IP security	50
7.2 How IP security works?	51
7.3 IP version 6	57
• PART EIGHT: SSH	60
8.1 The secure shell protocol	60
• PART NINE: HACKERS & ATTACKS	63
9.1 Meaning of being a hacker	63
9.2 Common attacks	64
9.3 Denial of Service attacks	64
9.4 Social engineering	67
9.5 Ping/Tracert/Netstat	69
9.6 Port scanning	70
9.7 IP spoofing	72
9.8 Buffer Overflow	73
9.9 Brute Force attacks	75
9.10 Sniffing	76
9.11 Intrusion detection systems	76
9.12 Cyberterrorism	79
9.13 Hacker incidents	80
• PART TEN: SPAM & SPYWARE	83
10.1 Spam emails	83
10.2 How do they get my address?	84
10.3 Stopping Spam	85
10.4 About Spyware	91
10.5 How Spyware operates	92
• PART ELEVEN: VIRUSES & TROJANS	95
11.1 All about viruses	95
11.2 Simple viruses	96
11.3 Encrypted viruses	97
11.4 Polymorphic viruses	99
11.5 Metamorphic viruses	104
11.6 How antivirus programs work?	104
11.7 Worms	106
11.8 What is a Trojan?	108
11.9 How do Trojans work?	109
11.10 Most common Trojans	110
11.11 In what ways could I be infected?	112
• PART TWELVE: BIOMETRIC SECURITY	114
12.1 What is a Biometric security?	114
12.2 Fingerprints	115
12.3 Hand geometry	115
12.4 Retina and Iris scanners	116

12.5 Face recognition	116
12.6 Signature scanning	117
12.7 Voice recognition	117
12.8 Uses for Biometrics	118
CHAPTER TWO: WIRELESS NETWORKS	
• PART ONE: BLUETOOTH	121
1.1 Introduction	121
1.2 How Bluetooth works?	123
1.3 Connection establishment and Bluetooth profiles	126
1.4 Bluetooth protocols	132
1.5 Bluetooth strengths and future	135
• PART TWO: IEEE 802.11 STANDARDS	137
2.1 The IEEE 802.11 standard	137
2.2 The IEEE 802.11a/802.11b standards	144
2.3 The IEEE 802.11g standard	148
2.4 Comparison of Bluetooth and IEEE 802.11 standards	150
• PART THREE: WIRELESS SECURITY	153
3.1 Security risks of 802.11	153
3.2 Why is 802.11 wireless networking technology insecure?	160
3.3 Ways to secure an 802.11 network	161
3.4 Security risks of Bluetooth	162
CONCLUSION	173
REFERENCES	174

INTRODUCTION

This project consists of two chapters: *first chapter* is about Information Security. In today's modern technology it is necessary and also very important to keep information secure. Hackers use several methods to attack, such as; Denial of Service Attacks (DoS), Brute Force attacks or Social Engineering based attacks. In order to keep a computer safe from hackers it is possible to use firewalls or different encryption techniques like PGP, SSL, etc. But we shouldn't forget that we are not 100% safe. However by using a strong encryption standard we can reduce hacker attacks. People try to attack in order to gain unauthorized access to the systems and cause damage in the system. Without proper protection information/data can be altered or stolen. Another big threat of Information security is Viruses&Trojans. Hackers write viruses to cause damage to computers and they use trojans to gain access to a systems and control their victim's computer remotely. Antivirus programs can be used to deal with Viruses&Trojans. Spam mails are another big problem, many people receive unwanted emails every day that we call them Spam mails. They contain unwanted information or advertisement of something (e.g. porn web site, product). Anti-Spam programs fight against spam mails by blocking them before they reach to our inbox.

Second chapter of my project is about Wireless Networks and their security issues. Wireless devices provide access to another device using radio waves rather than a cable infrastructure. There are two commonly used wireless technologies. One of them is Bluetooth, that enables connectivity between mobile devices with short range radio technology. Ericsson invented Bluetooth in 1994 and it operates in the 2.4 GHz radio frequency band, has a range of approximately 10 meters. Second wireless technology is invented by IEEE, named 802.11 standard. There are various kinds of 802.11 standard but most widely used ones are 802.11a/b/g. 802.11a operates in the frequency range of 5 GHz, 802.11b provides 11Mbps transmission in the 2.4 GHz band and 802.11g just like 802.11b devices operates in 2.4 GHz band. Wireless LANs brings security issues with them. Because they are easy to find and locate, there is high risk of danger in wireless LANs. Administrators should consider using Virtual Private Networks (VPN) or Wireless Encryption Privacy (WEP) to keep wireless devices secure. 802.11x also improves data security in wireless devices. Bluetooth devices can be kept secure if we correctly implement security policies inside the application profile. To minimize risks, IT administrators should implement wireless security policies and practices.

CHAPTER ONE: INFORMATION SECURITY

PART ONE: INTRODUCTION

1.1 INTRODUCTION:

Over the past few years, Internet enabled business, E-Business applications such as E-Commerce, supply-chain management and remote access allow companies to streamline processes, lower operating costs and increase customer satisfaction. Such applications require critical Networks that accommodate voice, video and data traffic and these Networks must be scalable to support increasing number of users and the need for greater capacity and performance. However as Networks enable more and more applications and are available to more and more users, they become even more vulnerable to a wider range of security threats. To combat those threats and ensure safety, security technology must play a major role in today's Networks. The objective of *Information Security* is to protect computers and their applications against attacks, ensuring information availability, confidentiality and integrity.

1.2 WHY NETWORKS MUST BE SECURED?

Without proper protection, any part of any Network can be susceptible to attacks or unauthorized activity. Routers, switches and hosts can all be violated by professional Hackers, company competitors or even internal employees. Network attacks can cause organizations several hours or days of downtime and serious breaches in data confidentiality and integrity. Depending on the level of the attack and the type of information that has been compromised, the consequences of Network attacks vary in degree from mildly annoying from to completely debilitating and the cost of recovery from attacks can range from hundreds to millions of dollars. For example; Companies that run E-Commerce web sites lose revenue as customers 'shop' elsewhere for their products and services. Informational web sites can lose precious advertising time and manufacturing organizations can be forced to shut down their lines because they cannot access information regarding their raw materials. If a Hacker gains access to an organizations E-Mail systems, information that is special for that company can be

stolen, resulting in a loss of research and development dollars spent in gaining that information. A Hacker may modify a web site replacing relevant information with nonsensical or offensive content. This results the proprietor of the site to spend money not only to fix the site but also to counter the resulting bad public relations.

Despite of large investments, the number of Network security threats are still increasing. There are well known reasons for that;

- New business requirements are making it more difficult to secure their assets. As new security devices are put in place in enterprise Networks, managing them becomes harder.
- Software packages and operating systems are becoming extremely complex feature to rich. They require keeping up with patches, a hard task at large enterprises as well as for home users.
- New type of technologies such as peer-to-peer, instant messaging and video conferencing involve complex Networking techniques that can be difficult to control.
- Networks increasingly have multiple entry points (ports) – for example; wireless or ftp access points. This exposes Networks to threats from unknown software and unprotected connections.
- Networks and applications have grown more complex and difficult to manage, even as qualified security professionals are scarce and IT budgets have become under pressure.
- Software development lifecycles result in flawed or poorly tested releases. As a result, the number of newly discovered and exploitable vulnerabilities has grown in the past five years.
- Hacking tools have become automated and require less skill to use, increasing the ranks of the Hackers. And because these tools are automated and designed for large scale attacks, a single hacker can rapidly inflict widespread damage.
- Worms, Viruses and Trojans boost damage through a multiplier effect. They keep on giving damage long after the initial incident.
- The lifecycle for Network attacks is shorter. Therefore companies have less time to identify and correct vulnerabilities before they are exploited by hackers and worms.

Table 1.1. Some Information Security reasons

Privacy or Confidentiality	→	Keeping information secret from all but those who are authorized to see it.
Data Integrity	→	Ensuring information has not been altered by unauthorized or unknown means.
Entity Authentication or Identification	→	Validation of the identity of a person, an entity or credit card.
Message Authentication	→	Validation of the source of information. Also known as data origin authentication.
Signature	→	A means to bind information to an entity.
Authorization	→	Authorizing validity of data.
Validation	→	A means to provide authorization to use or manipulate information or resources.
Access Control	→	Restricting access to resources to unauthorized person.
Certification	→	Endorsement of information by a trusted entity.
Timestamping	→	Recording the time of creation or existence of information.
Witnessing	→	Verifying the creation or existence of information by entity other than the creator.
Receipt	→	Acknowledgement that the information has been received.
Confirmation	→	Acknowledgement that services have been provided.
Ownership	→	A means to provide an entity with the legal right to use or transfer a resource to others.

1.3 WHY PEOPLE TRY TO ATTACK?

- Gain unauthorized access to information.

- To use others's license for the purpose of; modifying information, gain unauthorized access, use information for their purposes...
- Modify information according to themselves.
- Learn who accesses which information and when the accesses are made.
- Damage the function of software or even the computer.
- Cause damage in the system.
- Prevent communications among other users.
- To prove that they are professional and nobody can stop them.
- Steal something, usually money by getting credit card number of the victim.

Intruders often want gain control of your computer so they can use it to launch attacks on other computer systems. Having control of your computer gives them ability to hide their true location as they launch attacks, often against high-profile computer systems such as government or financial systems. Intruders may be able to watch all your actions on the computer, or cause damage to your computer by reformatting your hard drive or changing your data.

Security could be anything from hardware, software to human process. But what is interesting to know is that researchers have been identifying four main areas of security which every security aspect fall into. These four are;

- Confidentiality – Preventing unauthorized persons getting access to information/data.
- Integrity – Preventing unauthorized modification of information/data.
- Availability – Preventing unauthorized persons making information/data unavailable.
- Authentication – Preventing falsification of identity.

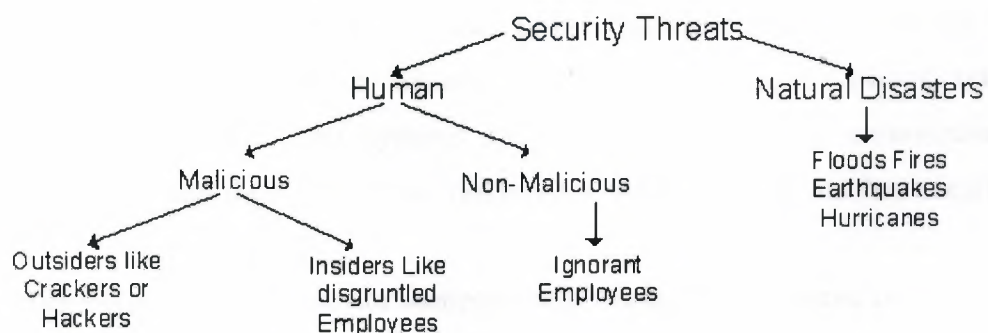


Figure 1.1. Security Threats

Nobody can stop nature from taking its course. Earthquakes, hurricane and fire can cause severe damage to computer systems. Information can be lost, downtime or loss of productivity can occur, and damage to hardware can disrupt other essential services. Few safeguards can be implemented against natural disasters. The best approach is to have disaster recovery plans in place. Other threats such as wars, terrorist attacks could be included here. Although they are human-caused threats, they are classified as disastrous.

Malicious threats consist of inside attacks by employees or by non-employees just looking to harm and disrupt an organization. The most dangerous attackers are usually insiders, because they know many of the codes and security measures that are already in place. Insiders likely to have specific goals and objectives, and have legitimate access to the system. Employees are the people most familiar with the organization's computers and applications, and they are most likely to know what actions might cause the most damage. Insiders can plant viruses, trojan horses or worms and they can browse through the file system. The insider attack can affect all components of computer security. By browsing through a system, confidential information could be revealed. Trojan horses are a threat to both the integrity and confidentiality of information in the system. Insider attacks can affect availability by overloading the system's processing or storage capacity, or by causing the system to crash.

People often refer to these individuals as "crackers" or "hackers". The definition of "hacker" has changed over the years. A hacker was once thought of as any individual who enjoyed getting the most out of the system he or she was using. A hacker would use a system extensively and study it until he or she became proficient in all its nuances. This individual was respected as a source of information for local computer users, someone referred to as "guru" or "wizard". Now however the term "hacker" refers to people who either break into systems for which they have no authorization or intentionally overstep their bounds on systems for which they do not have legitimate access.

The correct term to use for someone who breaks into systems is a "cracker". Common methods for gaining access to a system include password cracking, exploiting known security weaknesses, network spoofing, and social engineering. I will discuss all about those terms in detail in the previous parts of my project.

CHAPTER ONE: INFORMATION SECURITY

PART TWO: CRYPTOGRAPHY&ENCRYPTION

2.1 INTRODUCTION:

Data that can be read or understood without any special measures is called plaintext or cleartext. The method of disguising plaintext in such a way as to hide its substance is called *encryption*. Encrypting plaintext results in unreadable text called *ciphertext*. We use encryption to make sure that information is hidden from anyone for whom it is not intended. The process of reverting ciphertext to its original plaintext is called *decryption*. The following figure shows this process.

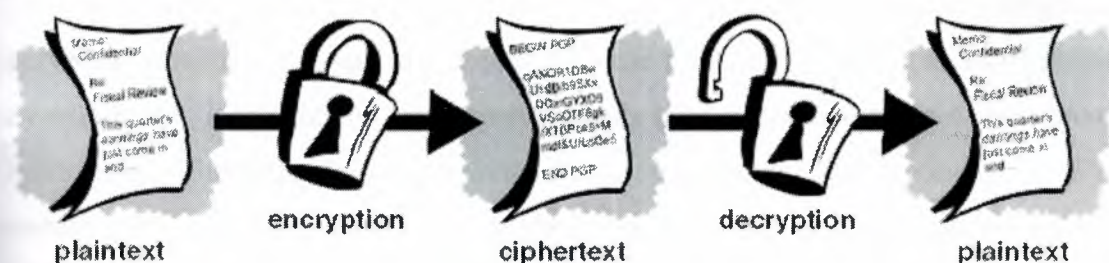


Figure 2.1. Encryption/Decryption Process.

Encryption is the process of taking information that exists in some readable form and converting it into a non-readable form. There are several types of commercially available encryption packages in both Hardware Software forms. Hardware encryption engines have the advantage that they are much faster than the software equivalent. The advantage of using encryption is that; Even if other access control mechanisms (passwords, file permissions etc.) are compromised by an intruder, the data is still unusable. Encryption ranges from simple encryption of files to special Network Hardware which encrypts everything without user intervention. Actually there are two main encryption techniques; Conventional, which is an old method and Public-Key, which is a new method used as an encryption technique.

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables us to store sensitive information or transmit it across insecure networks (like the Internet) so that it can not be read by anyone except the intended

recipient. While cryptography is the science of securing data, *cryptanalysis* is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination and luck. Cryptanalysts are also called attackers. A related discipline is *steganography*, which is the science of hiding messages rather than making them unreadable. Steganography is not cryptography; It is a form of coding. It relies on secrecy of the mechanisms used to hide the message. If for example, you encode a secret message by putting each letter as the first letter of the first word of every sentence, it's secret until someone knows to look for it, and then it provides no security at all.

2.2 CONVENTIONAL ENCRYPTION:

Conventional Encryption also referred to as symmetric encryption or single-key encryption was the only type of encryption in use prior to the development of public-key encryption. It remains by far the most widely used of the two types of encryption. Figure below illustrates the conventional encryption process. The original intelligible message, referred to as *plaintext*, is converted into apparently random nonsense, referred to as *ciphertext*. The encryption process consists of an algorithm and a key. The *key* is a value used to establish authority to access particular information by the operating system by assigning identification numbers to the memory. The *key* is a number used to encrypt plaintext into ciphertext. The algorithm will produce a different output depending on the specific key being used at the time. Changing the key, changes the output of the algorithm.

Once the ciphertext is produced, it can be transmitted. Upon reception, the ciphertext can be transformed back to the original plaintext by using a decryption algorithm and the same key that was used for encryption. Figure below demonstrates how conventional encryption works:

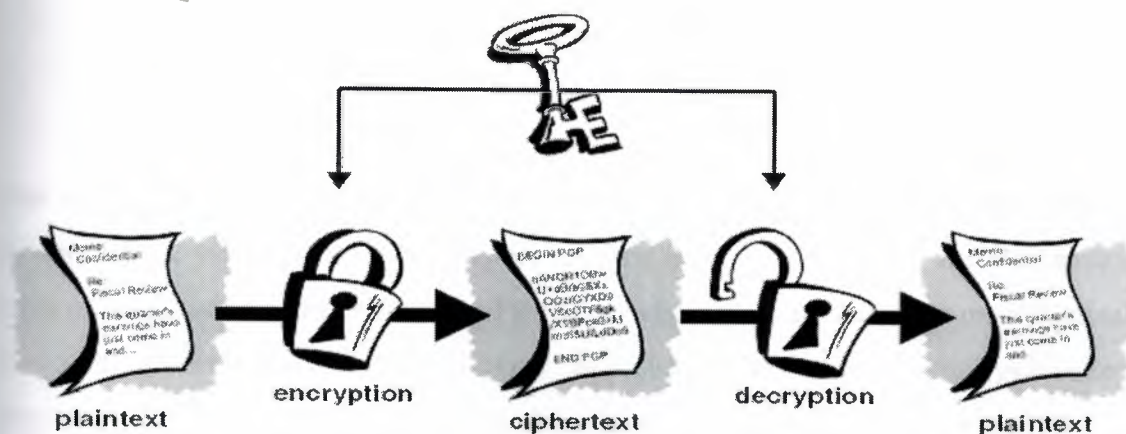


Figure 2.2. Conventional Encryption

The security of conventional encryption depends on several factors. First, the encryption algorithm must be powerful enough that it is impractical to decrypt a message on the basis of the ciphertext alone. Beyond that, the security conventional encryption depends on the secrecy of the key, not the secrecy of the algorithm. That is, it is assumed that it is impractical to decrypt a message on the basis of the ciphertext plus knowledge of the encryption/decryption algorithm. In other words, we do not need to keep algorithm secret, we need to keep only the key secret.

This feature of conventional encryption is what makes it feasible for widespread use. The fact that the algorithm need not be kept secret means that manufacturers can and have developed low-cost chip implementations of data encryption algorithms. These chips are widely available and incorporated into a number of products. With the use of conventional encryption, the principal security problem is maintaining secrecy of the key.

Conventional encryption has benefits. It is very fast. It is especially useful for encrypting data that is not going anywhere. However conventional encryption alone as a means for transmitting secure data can be quite expensive simply due to the difficulty of secure key distribution. For a sender and recipient to communicate securely using conventional encryption, they must agree upon a key and keep it secret between themselves. If they are in different physical locations, they must trust a courier or other secure communications medium to prevent the disclosure of the secret key during transmission.

2.3 PUBLIC-KEY ENCRYPTION:

The development of public-key cryptography is the greatest and perhaps the only true revolution in the entire history of cryptography. From its earliest beginnings to modern times, virtually all cryptographic systems have been based on the elementary tools of substitution and permutation. Public-key algorithms are based on mathematical functions rather than on substitution and permutation. Moreover public-key cryptography is asymmetric involving the use of two separate keys, in contrast to symmetric conventional encryption, which uses only one key. Public-key encryption is more secure than conventional encryption so it is widely used.

The public-key algorithms rely on one key for encryption and a different but related key for decryption. These algorithms have the following important characteristics:

- ▶ It is computationally impossible to determine the decryption key and encryption key.
- ▶ Either of the two related keys can be used for encryption, with the other used for decryption.

Figure below illustrates the public-key encryption process. The essential steps are following:

- 1) Each end system in a Network generates a pair of keys to be used for encryption and decryption of messages that it will receive.
- 2) Each system publishes its encryption key by placing it in a public register or file. This is the public key. The companion key is kept private.
- 3) If A wishes to send a message to B, it encrypts the message using B's public key.
- 4) When B receives the message, B decrypts it using B's private key. No other recipient can decrypt the message because only B knows B's private key.

Public-key cryptography uses pair of keys; A public-key, which encrypts data and a corresponding private key, for decryption. Because it uses two keys, it is sometimes called asymmetric cryptography. You publish your public-key to the world

while keeping your private-key secret. Anyone with a copy of your public key can then encrypt that only you can read, even people you have never met. It is computationally impossible to deduce the private-key from the public-key. Anyone who has a public-key can encrypt information but can not decrypt it. Only the person who has the corresponding private-key can decrypt the information.

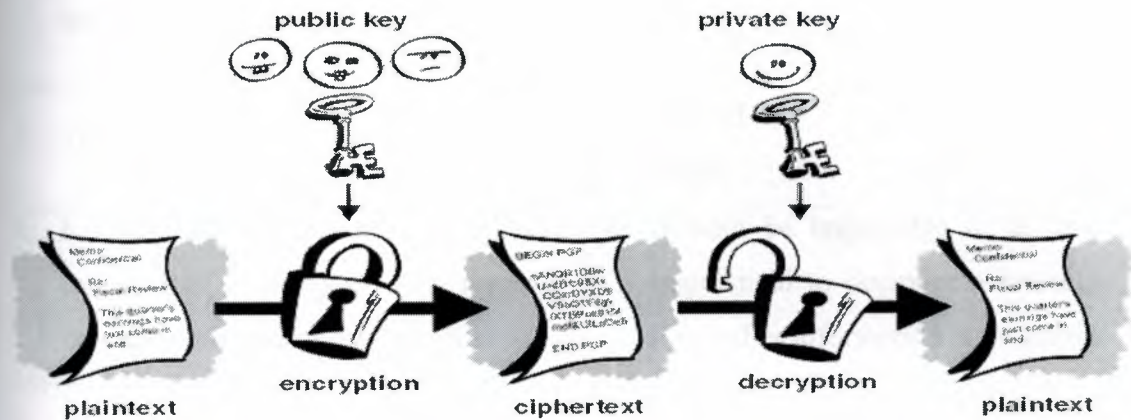


Figure 2.3. Public-Key Encryption

With this approach all participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed. As long as a system controls its private key, its incoming communication is secure. At any time, a system can change its private-key and publish the companion public-key to replace its old public-key. The two keys used for public-key encryption are referred to as the “*public key*” and the “*private key*”. The key used in conventional encryption refer to as a “*secret key*”.

The primary benefit of public-key cryptography is that it allows people who have no preexisting security arrangement to exchange message securely. The need for sender and receiver to share secret keys via some secure channel is eliminated; All communications involve only public-keys, and no private-key is ever transmitted or shared. Some examples of public-key cryptosystems are; Elgema (named for its inventor; Taher Elgema), RSA (named for its inventor; Ron Rivest, Adi Shamir and Leonard Adleman) and DSA, digital signature algorithm. Public-key encryption is the technological revolution that provides strong cryptography.

Table 2.1. Characteristics of Conventional and Public-Key Encryption.

<i>Conventional Encryption:</i>	<i>Public-Key Encryption:</i>
1. The same algorithm with the same key is used for encryption and decryption.	1. One algorithm is used for encryption and one for decryption.
2. The sender and receiver must share the algorithm and the key.	2. The sender and receiver must each have one of the matched pair of keys (not the same one).
3. The key must be kept secret.	3. One of the two keys must kept secret.
4. It must be impossible or at least impractical to decipher a message if no other information is available.	4. It must be impossible or at least impractical to decipher a message if no other information is available.
5. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	5. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

2.4 DIGITAL SIGNATURES

A major benefit of public-key cryptography is that it provides a method for employing digital signatures. *Digital signatures* let the recipient of information, verify the authenticity of the information's origin, and also verify that the information was not altered while in transit. Thus public-key digital signatures provide authentication and data integrity. These features are every bit as fundamental to cryptography as privacy, if not more.

A digital signature serves the same purpose as a seal on a document, or a handwritten signature. However because of the way it is created, it is superior to a seal or signature in an important way. A digital signature not only attests to the identity of the signer, but it also shows that the contents of the information signed has not been modified. A physical seal or handwritten signature cannot do that. However like a physical seal that can be created by anyone with the private key of that signing keypair.

The basic manner in which digital signatures are created is shown in the following figure. The signature algorithm uses your private-key to create the signature and the public-key to verify it. If the information can be decrypted with your public-key, then it must have originated with you.

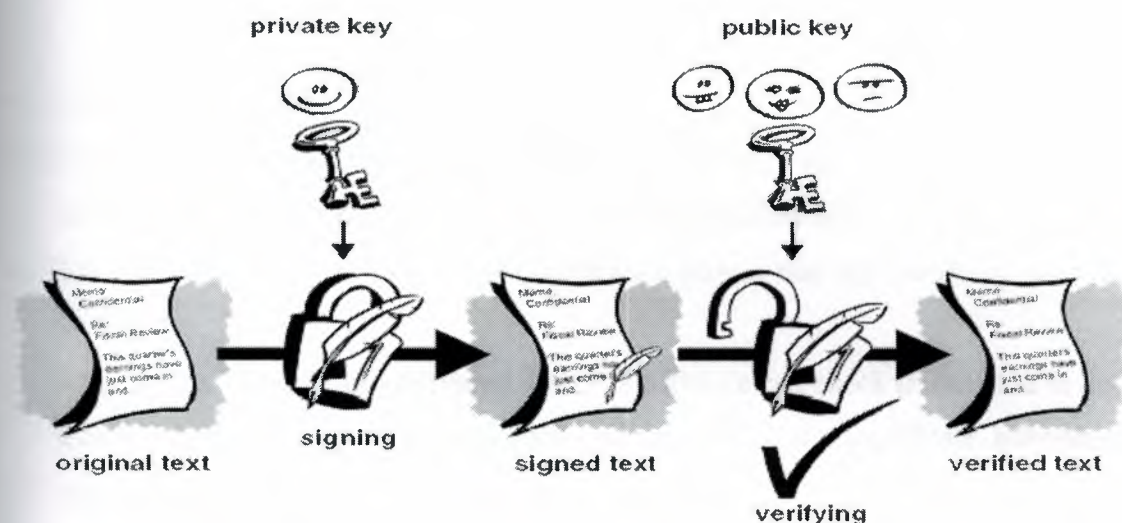


Figure 2.4. Digital Signatures.

Digital ID signature or certificate is an installed file resident on a computer that validates who you are. Digital signatures are used by programs on the Internet and local to the machines to confirm your identity to any third party concerned. Digital signatures have been confused with electronic signatures. Electronic signatures are scanned copies of a physical written signature.

2.5 REASONS FOR USING DIGITAL SIGNATURES

- It ensures by means of verification and validation that the user is whom he/she claims to be. This is done by combine the users credential to the digital certificate and in turn this method uses one point of authentication.
- Digital certificates ensure data integrity giving the user piece of mind that the message or transaction has not been accidentally or maliciously altered. This is done cryptographically.
- Digital certificates ensure confidentiality and ensure that messages can only be read by authorized intended recipients.

- Digital certificates also verify date and time so that senders or recipients can not dispute if the message was actually send or received.

2.6 THE PROCESS OF CHECKING THE VALIDITY OF DIGITAL SIGNATURE

- User A sends a signed document to user B.
- To verify the signature on the document, user B's application first uses the certificate authority's public-key to check the signature on user A's certificate.
- The successful de-encryption of the certificate proves that the certificate authority created it.
- After the certificate is de-encrypted, user B's software can check if user A is in good standing with the certificate authority and that all of the certificate information concerning user A's identity has not been altered.
- User B's software then takes user A's public key from the certificate and uses it to check user A's signature. If user A's public key de-encrypts the signature successfully, then user B is assured that the signature was created using user A's private key, for the certificate authority has certified the matching public-key.
- If the signature is found to be valid, then we know that an intruder didn't try to change the signed content.

Digital Certificates: Public-Key encryption wouldn't be practical to use for applications such as web servers for online transactions. For this purpose digital certificates were developed. The digital certificate is a small file provided to each computer by an independent system called a certification body, this tells each computer that the other one is who it says it is and that it can be trusted, the certification body then sends the public keys of each computer to the other, and they are free to communicate. The digital certificate method is mostly used in SSL (disgussed later).

CHAPTER ONE: INFORMATION SECURITY

PART THREE: FIREWALLS

3.1 WHAT DOES IT DO?

Basically a *Firewall* is a barrier to keep destructive forces away from your property or in other words it is a system that which enforces an access control policy between two networks. In fact that's why its called a firewall. Its job is similar to a physical firewall that keeps a fire from spreading from one area to the next. The firewall is inserted between the premises network and the Internet to establish a controlled link and to erect an outer security-wall or perimeter. The aim of this perimeter is to protect the premises network from Internet based attacks and to provide a single choke point where security and audit can be imposed. It is simply a program or hardware device that filters the information coming through the Internet connection into our private network or computer systems.

→ The following capabilities are within the scope of a firewall:

- A firewall defines a single check point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network, and provides protection from various kinds of attacks.
- A firewall provides a location for monitoring security related events. Audits and alarms can be implemented on the firewall system.
- Controls access to a service according to which user is attempting to access it.
- Controls how particular services are used. For example, the firewall may filter e-mail to eliminate spam, or it may enable external access to only a portion of the information on a local web server.

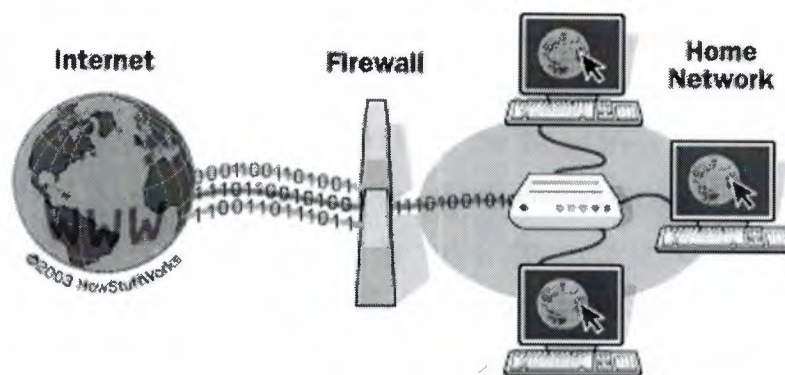


Figure 3.1. Firewalls block unwanted Internet traffic.

→ Firewalls can be classified into 3 types:

1. Packet filter firewalls.
2. Application proxy firewalls.
3. Packet inspection firewalls.
4. Stateful multilayer inspection firewall.

3.2 PACKET FILTER FIREWALLS

They are the earliest and the most criticized firewalls, which nowadays are not easily found. They are usually hardware based, i.e. router based (a router is a device which connects two networks together). Whenever a packet filter firewall receives a packet for permission to pass through, it compares the header information, i.e. the source and destination IP address, and port number with a table of predefined access control rules, if the header information matches, then the packet is allowed to pass, else the packet is dropped or terminated. They are not popular due to the fact that they allow direct contact between the untrusted system and the trusted private system. To understand such firewalls let's take the example of the secretary that sits in the office. This kind of secretary allows only those people who have an appointment with boss, but if you convince her that her boss wants to meet with you then she would allow you to pass. Such firewalls can be fooled by using techniques like IP spoofing in which we can change the source IP such that the firewall thinks that the packet has come from a trusted system which is among the list of systems which has access through the firewall.

A packet filter firewall applies a set of rules to each IP packet and then forwards or discards the packet. The router is typically configured to filter packets in both directions. Filtering rules are based on fields in the IP address and IP protocol. The packet filter firewall is typically set up as a list of rules based on matches to fields in the IP header. If there is a match to one of the rules, that rule is invoked to determine whether to allow or discard the packet. Depending on the packet and the criteria, the firewall can drop the packet, forward it or send a message to the originator. Rules can include source and destination IP address, port number and protocol used. The advantage of packet filter firewall is their low cost and low impact on network performance. Most routers support packet filtering. Table below gives some examples of packet-filtering rule sets.

Table 3.1. Packet-Filtering Rule Sets.

Action	Our Host	Port	Their Host	Port	Comment
Block	<u>www.internet.com</u>	25	<u>www.astalavista.com</u>	55	We don't trust these people.
Allow	<u>www.internet.com</u>	25	<u>www.cnn.com</u>	75	They can connect to our server.

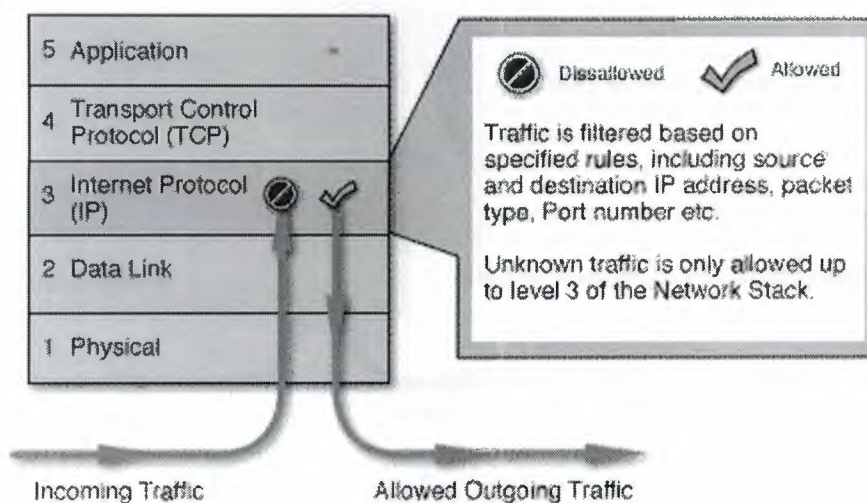


Figure 3.2. Packet Filter Firewall

3.3 APPLICATION PROXY FIREWALLS (application level)

This kind of proxy firewall examines what application or service packet is meant for, and if that particular service is available only, then the packet is allowed to pass through, and if the service is unavailable then the packet is discarded or dropped by the firewall. Once this is done the firewall extracts the data and delivers it to the appropriate service. There is not direct connection between the untrusted systems with the trusted systems, as the original data sent by the untrusted system is dropped by the firewall and it personally delivers the data. Lets again take the example of a secretary. Such a secretary would take a gift or something else for you, only if you are available in the office and she would not allow the visitor to deliver the thing but would personally deliver it to you. Although they are somewhat slower, they are much more secure as they do not allow a direct contact between an untrusted network and a trusted network.

An application level firewall that is configured to be a web proxy will not allow any ftp, telnet or other traffic through. Because they examine packets at application layer, they filter application specific commands such as http: get and post. This cannot be accomplished with other type of firewalls. Application level firewalls can also be used to log user activity and logins. They offer a high level of security, but have a significant impact on network performance. This is because of context switches that slow down network access dramatically.

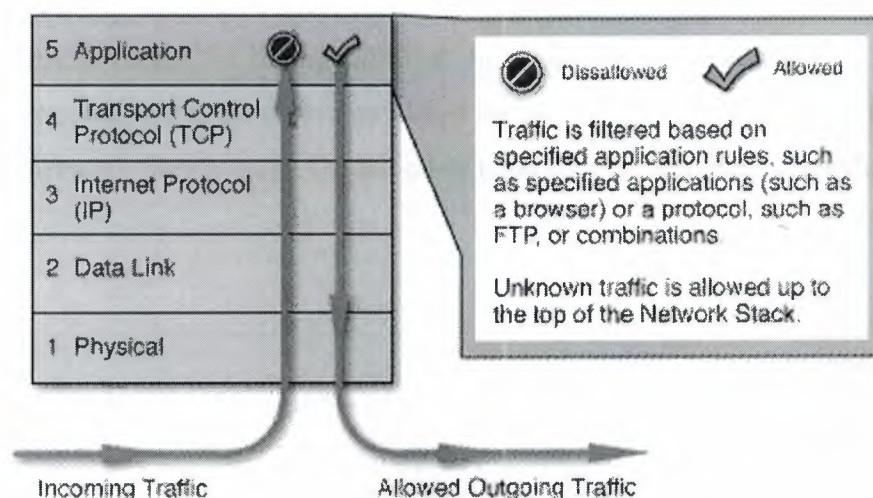


Figure 3.3. Application Proxy Firewall

3.4 PACKET INSPECTION FIREWALLS (circuit-level gateway)

It can be also known as an extension of the packet-filter firewall. It does not only verifies the source, destination IP's or ports, but it also takes into consideration or verifies the content of the data before passing it through. There are two ways in which this kind of firewall verifies the data to be passed: State and Session.

In case of *state* inspection, an incoming packet is allowed to pass through only, if there is a matching outward bound request for this packet. This means that the incoming packet is allowed to pass through only if the trusted server had requested for it or had sent an invitation for it. In case of *session* filtering, the data of the incoming is not verified, but instead the network activity is traced, and once a trusted system ends the session, no further packets from that system pertaining to that session are allowed to pass through. This protects against IP spoofing to a certain extend. Such firewalls can also be configured beforehand to act according to pre-defined rules when it is attacked. It can also be configured to disconnect from the Internet in case of an attack. Actually the decision to accept or reject a packet (data) is usually based on the source, destination or port number.

Circuit level firewalls work at the TCP layer of TCP/IP protocol. They monitor TCP port to determine whether a requested session is legitimate. Information passed to a remote computer through circuit level gateway appears to have originated from the gateway. This is useful for hiding information about protected networks. Circuit level gateways are relatively inexpensive and have the advantage of hiding information about the private network they protect. On the other hand they do not filter individual packets.

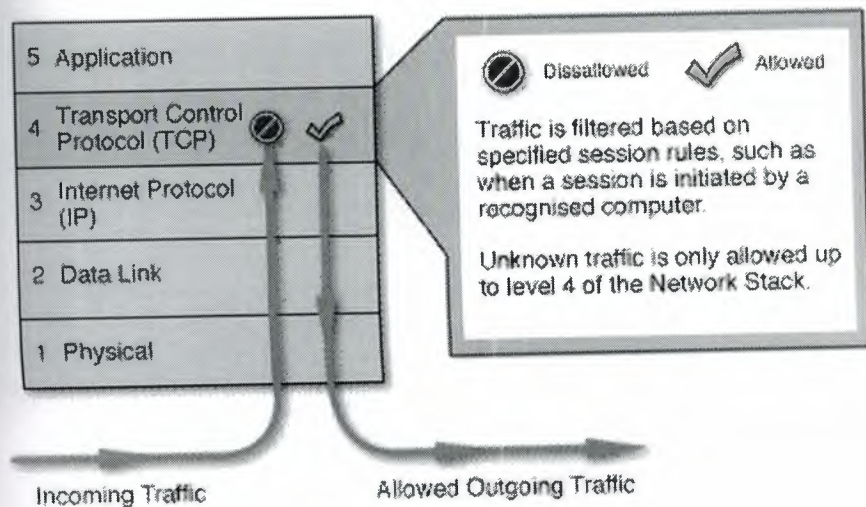


Figure 3.4. Packet Inspection Firewall

3.5 STATEFUL MULTILAYER INSPECTION FIREWALL

Stateful multilayer inspection firewalls combine the aspects of the other three types of firewalls. They filter packets at the network layer, determine whether session packets are legitimate and evaluate contents of packets at the application layer. They allow direct connection between client and host, avoiding the problem caused by the lack of transparency of application level gateways. They rely on algorithms to recognize and process application layer data instead of running application specific proxies. Those kind of firewalls offer a high level of security, good performance and transparency to end users. They are expensive however, and due to their complexity, are potentially less secure than simpler types of firewalls if not administered by highly competent personnel.

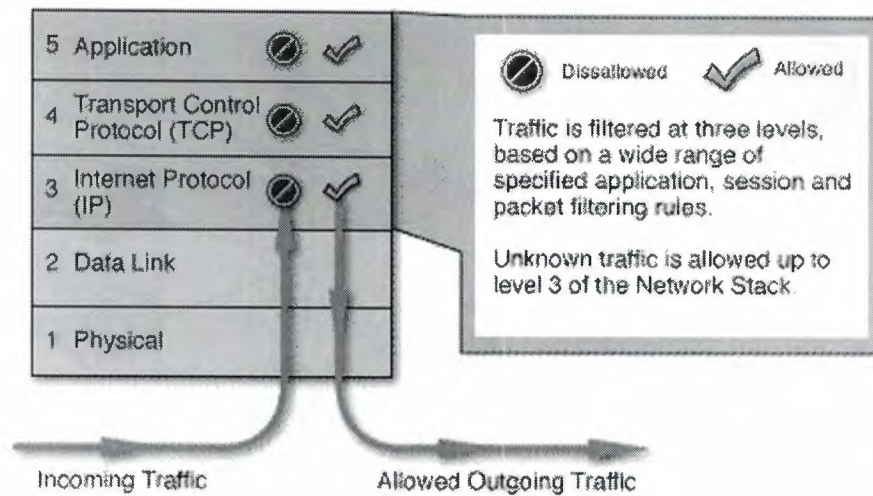


Figure 3.5. Stateful Multilayer Inspection Firewall.

3.6 HARDWARE VS. SOFTWARE FIREWALLS

All firewalls run a firewall software and they all run it on some sort of hardware, but the terms hardware and software firewall are used to distinguish between products marketed as an integrated appliance that comes with the software preinstalled, usually on a operating, and firewall programs that can be installed on general purpose network operating systems such as Windows or Unix. Hardware firewalls can be further divided into those that are basically dedicated PC's with hard-disks and those that are solid state devices. Those kind of firewalls are generally faster performers and don't have the hard-disk as a potential point of failure. Software firewalls include Symantec Firewall and McAfee Firewall. Hardware firewalls include Cisco PIX, Nokia and Watchguard. Hardware firewalls are often marketed as "turn key", because you dont have to install the software or worry about hardware configuration conflicts. Those that run on OS. claim greater security because the OS. is already "hardened". A disadvantage of hardware firewalls is that you are locked into the vendor's specifications. For instance, a firewall appliance will have a certain number of network interfaces, and you are stuck with that number. With a software firewall you can easily upgrade the standard PC on which the software firewall runs, easily adding standard RAM or even multiple processors for better performance.

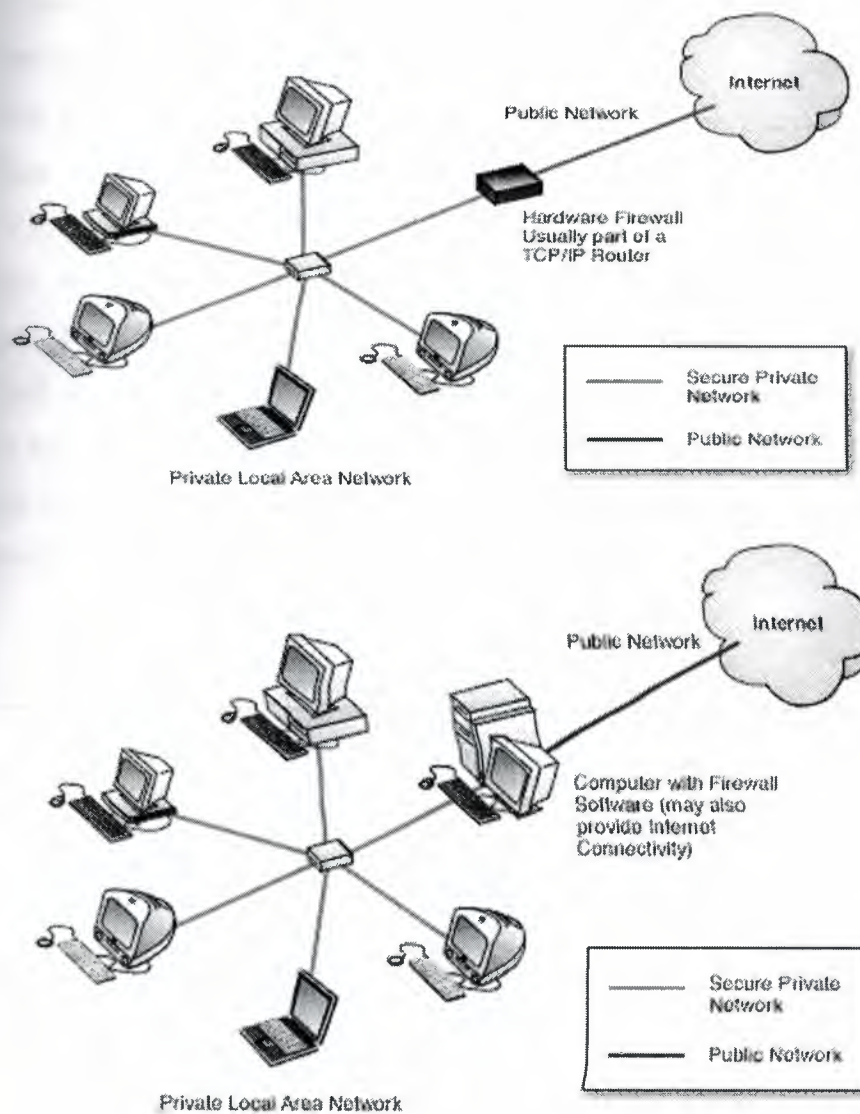


Figure 3.6. Hardware vs. Software Firewalls.

3.7 DO FIREWALLS PROVIDE ENOUGH SECURITY?

There is no such thing that a firewall is enough to fulfill or satisfy all our security concerns. Yes It does protect the trusted systems from the untrusted ones, but they are definitely not enough for all our security needs. We need to protect our systems to secure the company data. A firewall is still not able to protect the company data from Viruses or Trojans, they also does not provide physical protection to the networks. Firewalls have their limitations, including the following:

- Firewall can not protect against attacks that bypass the firewall.
- The firewall does not protect against internal threats, such as employees.
- The firewall cannot protect against the transfer of virus infected programs or files.

Since firewalls must examine every packet, they often decrease network performance. They also block the most obvious ports such as file sharing or remote control which are essential for communication. Another thing that firewalls can't really protect you against is possible hackers inside your network. Firewalls can't protect very well against things like viruses, although there are 'virus detecting' firewalls, they can't protect networks against new viruses, so they remain useless.

CHAPTER ONE: INFORMATION SECURITY

PART FOUR: PGP

4.1 WHAT IS PGP?

PGP which stands for 'Pretty Good Privacy' is an encryption technology which combines features of both conventional and public key cryptography.

Conventional Cryptography: This type of encryption uses the same key to encrypt and decrypt data. An example of conventional cryptography is 'Data encryption standard' which is used for commercial applications. Conventional cryptography has both pluses and minuses. It is very fast and suitable for data which won't be used by anyone except by the person who encrypted it. Unfortunately the secure key distribution is very difficult task to accomplish.

Public Key Cryptography: It solves the secure key distribution problem. Public key cryptography is an asymmetric system and uses two keys: A public key, used for encryption and a private key, used for decryption.

4.2 HOW DOES PGP WORK?

PGP combines some of the best features of both conventional and public-key cryptography. When a user encrypts plaintext with PGP, PGP first compresses the plaintext. Data compression saves modem transmission time, disk space and more importantly strengthens cryptographic security. Most cryptanalysis techniques exploit patterns found in the plaintext to crack the cipher. Compression reduces these patterns in the plaintext, thereby greatly enhancing resistance to cryptanalysis. PGP then creates a session-key, which is a one time only secret-key. This key is a random number generated from the random movements of your mouse and the keystrokes you type. The session-key works with a very secure, fast conventional encryption algorithm to encrypt the plaintext; the result is ciphertext. Once the data is encrypted session key is

⇒ Encryption: First PGP compresses plaintext (*plaintext is unencrypted data). It is useful for several reasons; You need less space on hard-disk, smaller message means saving time and money. PGP then generates a single use encryption key, known as a session key. It is random number, generated from random data such as contents of your PC's RAM, positions of window on the desktop. PGP uses a very fast and conventional session key to encrypt the data to produce ciphertext (*ciphertext is the result of the encryption). After encryption of the data the session key is then encrypted to the recipients public key and both the public key, encrypted session key and the ciphertext are transmitted.

⇒ Decryption: PGP uses the recipients private key to recover the session key. The session key is used to decrypt the conventionally encrypted ciphertext. The compressed data is decompressed. The combination of conventional and public keys provide cryptography with very fast and secure encryption system. This is achieved by the speed of conventional algorithms and safety of public key (*a key is a piece of data which is used to produce ciphertext).

→ PGP has grown explosively and is now widely used. A number of reasons can be cited for this growth:

- It is available free worldwide in versions that run on a variety of platforms, including Windows, Unix, Macintosh and many more.
- It is based on algorithms that have survived extensive public review and are considered extremely secure.
- It has a wide range of applicability, from corporations that wish to select and enforce a standardized scheme for encrypting files and messages to individuals who wish to communicate securely with others worldwide over the Internet and other Networks.
- It was not developed by, nor is it controlled by, any governmental or standards organization. For those with an instinctive distrust of 'the establishment' this makes PGP attractive.

4.3 HASH FUNCTIONS

The system described above has some problems. It is slow and it produces an enormous volume of data, at least double the size of the original information. An improvement is the addition of a *hash function* in the process. A hash function takes variable-length input, in this case, a message of any length, even thousands or millions of bits and produces a fixed length output, say 160 bits. The hash function ensures that, if the information is changed in any way, even just by one bit, an entirely different output value is produced. PGP uses cryptographically strong hash function on the plaintext that the user signing. This generates a fixed-length data item known as a *message digest*. Then PGP uses digest and the private-key to create the “signature”. PGP transmits the signature and the plaintext together. Upon receipt of the message, the recipient uses PGP to recompute the digest, thus verifying the signature. PGP can encrypt the plaintext or not. Signing plaintext is useful if some of the recipients are not interested in or capable of verifying the signature. As long as a secure hash function is used, there is no way to take someone’s signature from one document and attach it to another, or to alter a signed message in any way. The slightest change to a signed document will cause the digital signature verification process to fail. Digital signatures play a major role in authenticating and validating the keys of other PGP users.

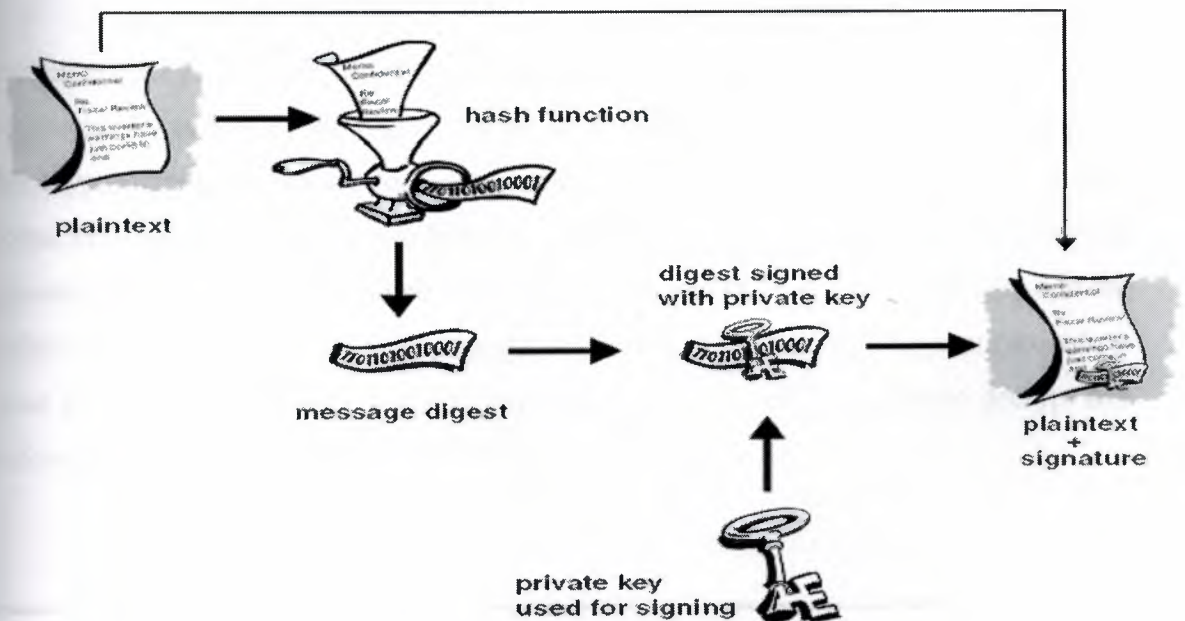


Figure 4.3. Hash Functions.

Hash functions take a message as an input and produce an output referred to as a hash-code, hash value or simply hash. More precisely, a hash function 'h' maps strings of arbitrary finite length to strings of fixed length, say n bits. The basic idea of cryptographic hash functions is that a hash value serves as a compact representative image of an input string, and can be used as if it were uniquely identifiable with that string. Hash functions are used for data integrity in conjunction with digital signatures, where for several reasons a message is typically hashed first, and then the hash value, as a representative of the message, is signed in place of the original message.

A typical usage of hash functions for data integrity is as follows; The hash value corresponding to a particular message, let's say X is computed at time $T1$. The integrity of this hash value (but not the message itself) is protected in some manner. At a subsequent time $T2$, the following test is carried out to determine whether the message has been altered, whether the message $X1$ is the same as the original message. The hash value of $X1$ is computed and compared to the protected hash value; If they are equal, one accepts that the inputs are also equal, and thus that the message has not been altered. The problem of preserving the integrity of a potentially large message is thus reduced to that of a small fixed-size hash value.

4.4 KEYS

A *key* is a value that works with a cryptographic algorithm to produce a specific ciphertext. Keys are basically really, really big numbers. Key size is measured in bits. In cryptography and PGP, the bigger the key, the more secure the ciphertext. Larger keys are cryptographically secure for a longer period of time. Keys are stored in encrypted form. PGP stores the keys in two files on our Hard-Disk. One for public-key and one for private-key. These files are called keyrings. As you use PGP, you will typically add the public-keys of your recipients to your public keyring. Your private keys are stored on your private keyring. If you lose your private keyring you will be unable to decrypt any information encrypted to keys on that ring.


CHAPTER ONE: INFORMATION SECURITY

PART FIVE: SSL

5.1 WHAT IS SSL?

Secure Socket Layer (SSL) is a protocol designed to enable secure communications on an insecure Network such as the Internet. SSL provides encryption and integrity of communications along with strong authentication using digital certificates. Most of all web-based online purchases and monetary transactions are now secured by SSL. When you submit your credit card to purchase something from amazon.com the order form information is sent through this secure tunnel so that only guys at amazon.com can view it.

The SSL protocol was originally developed by Netscape, to ensure security of data transported and routed through HTTP layer. SSL designed to make use of TCP as a communication layer to provide a reliable end-to-end secure and authenticated connection between two points over a Network. SSL can be used in protection of data in transit in situations related to any Network service, it is used mostly in HTTP server and client applications. Today almost each available HTTP server can support an SSL session, whilst IE or Netscape navigator browsers are provided with SSL enabled client software.

When you come across a web page that is secured, your browser will likely display a 'closed lock'  to inform you that SSL has been enabled. The web site address should also start with 'https://' rather than usual 'http://'. SSL allows secure connection between your browser web server. It is developed by Netscape communications and was based on encryption algorithms developed by RSA security

5.2 SSL OBJECTIVES AND ARCHITECTURE

→ The main objectives for SSL are:

- Authenticating the server and client to each other. The SSL protocol supports the use of standard key cryptographic techniques (public-key encryption) to authenticate

the communicating parties to each other. Through the most frequent application consists in authenticating the service client on the basis of a certificate, SSL may also use the same methods to authenticate the client.

- Ensuring data integrity: During a session, data cannot be either intentionally or unintentionally tampered with.
- Securing data privacy: data in transport between the client and the server must be protected from interception and be readable only by the intended recipient. This prerequisite is necessary for both the data associated with the protocol itself and the application data that is sent during the session itself. SSL in fact not a single protocol but rather a set of protocols that can additionally be further divided in two layers:

1. The protocol to ensure data security and integrity. This layer is composed of the *SSL Record protocol*.
2. The protocol that are designed to establish an SSL connection. Three protocols are used in this layer; The *SSL Handshake protocol*, the *SSL ChangeCipher Spec protocol* and the *SSL Alert protocol*. The SSL protocol stack is illustrated below;

SSL handshake protocol	SSL cipher change protocol	SSL alert protocol	Application Protocol (eg. HTTP)
SSL Record Protocol			
TCP			
IP			

Figure 5.1. SSL protocols.

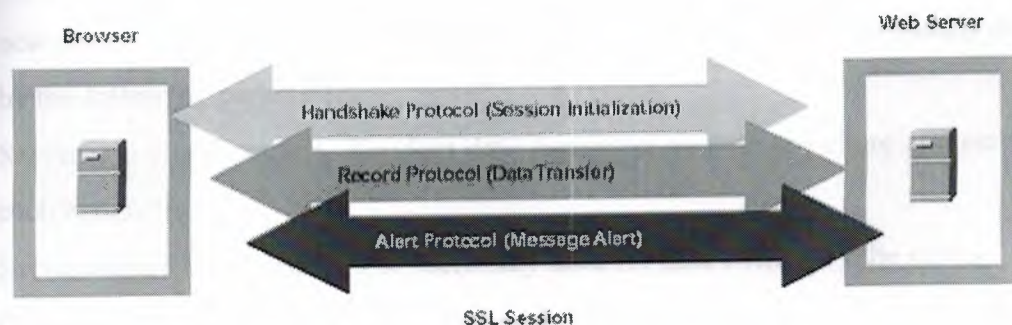


Figure 5.2. SSL protocol stack.

SSL uses these protocols to address the tasks as described above. The SSL record protocol is responsible for data encryption and integrity, it is also used to encapsulate data sent by other SSL protocols, and therefore, it is also involved in the tasks associated with the SSL check data. The other three protocols cover the areas of session management, cryptographic parameter management and transfer of SSL messages between the client and the server.


5.3 SSL SESSION AND CONNECTION

- **Connection:** This is a logical client/server link, associated with the provision of a suitable type of service.
- **Session:** This is an association between a client and a server that defines a set of parameters such as algorithms used, session number etc. An SSL session is created by the Handshake protocol that allows parameters to be shared among the connections between the server and the client, and sessions are used to avoid negotiation of new parameters for each connection. The concepts of a SSL session and connection involve several parameters that are used for SSL enabled communication between the client and the server. During the negotiations of the handshake protocol, the encryption methods are established and a series of parameters of the session state are subsequently used within the session. A session state is defined by the following parameters:
 - **Peer certificate:** X.509 certificate of the peer.
 - **Compression method:** A method used to compress data prior to encryption.
 - **Algorithm specification termed CipherSpec:** Specifies the bulk data encryption algorithm and the Hash algorithm used during the session.
 - **Master secret:** 48-byte data being a secret shared between the client and the server.
 - **"Is resumable":** This is a flag indicating whether the session can be used to initiate new connections. According to the specification, the SSL connection state is defined by the following parameters:
 - **Server and client random:** Random data generated by both the client and server for each connection.
 - **Server write *MAC secret:** The secret key used for data written by the server.
 - **Client write *MAC secret:** The secret used for data written by the client.

- **Server write key:** The bulk cipher key for data encrypted by the server and decrypted by the client.
- **Client write key:** The bulk cipher key for data encrypted by the server and decrypted by the server.
- **Sequence number:** Sequence numbers maintained separately by the server for messages transmitted and received during the data session.

5.4 HOW DOES IT WORK?

An *SSL* certificate lets users know that the information they send through a web site such as credit card numbers, online forms and other data is protected from interception or alteration over the web. Step by step description is given below:

1. A user contacts another site, lets say an online shopping site and accesses a secured URL: A page secured by server's ID. (<https://>)
2. Online shopping site's server responds and automatically sends the user it's digital certificate, which authenticates online shopping site. Digital certificate establish whether a public-key truly belongs to the owner. A certificate is a form of identification, like social security card or ID card of us.
3. User web browser generates a unique "session key" (like a code) to encrypt all communications with the site.
4. The user's browser encrypts the session key with the online shopping site's public-key, so only it can read the session key. Depending on the browser, the user may see a key icon  becoming whole or a padlock closing, indicating that the session is secure.
5. A secure session is now established. All communications will be encrypted and can only be decrypted by the two parties in the session. It all takes only seconds and requires no action by the user.

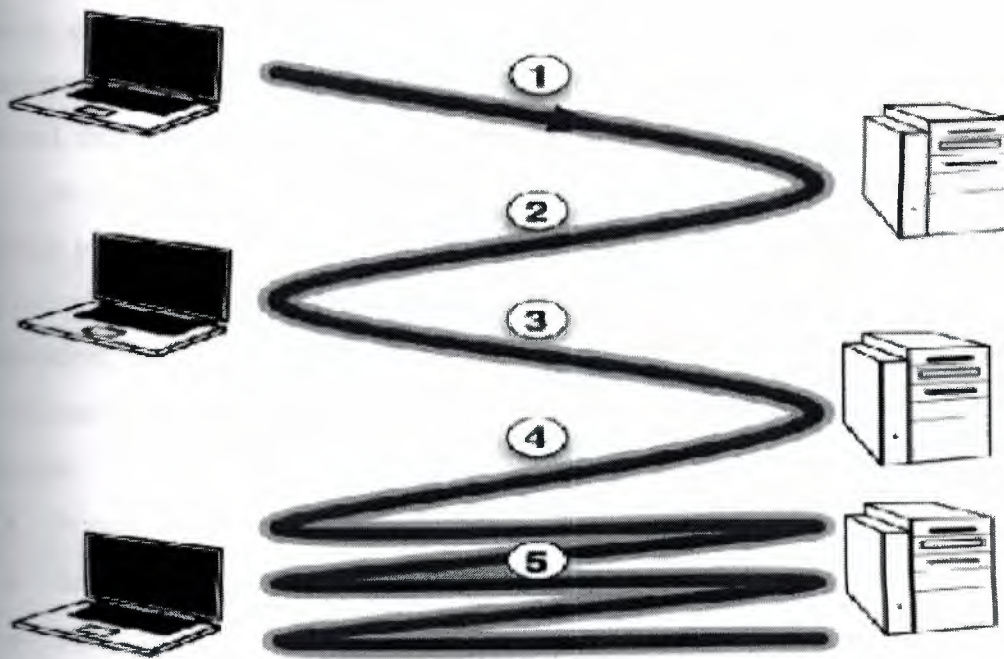


Figure 5.3. Step by step operations of SSL

What SSL does in short would be; Encrypt data at the senders end and decrypt data at the receivers end. This encrypted data cannot be picked up or hijacked in between.

As soon as you enter a secure site, SSL comes into play. The main SSL protocol is made up of two smaller sub-protocols;

- The Secure Socket Layer Record Protocol.
- The Secure Socket Layer Handshake Protocol.

5.5 THE SSL RECORD PROTOCOL

The SSL record protocol involves using SSL in secure manner and with message integrity ensured. The purpose of the *SSL Record protocol* is to take an application message to be transmitted, fragment the data which needs to be sent, encapsulate it with appropriate headers and create an object just called a record, which is encrypted and can be forwarded for sending under the TCP protocol. The first step in the preparation of transmission of the application data consists in its fragmentation i.e. breaking up the data stream to be transmitted into 16Kb or smaller data fragments followed by the

process of their conversion in a record. These data fragments may be further compressed, although the SSL 3.0 protocol specification includes no compression protocol, thus at present, no data compression is used.

At this moment, creation of the record is started for each data portion by adding a header to it, possible information to complete, the required data size and the MAC. The record header that is added to each data portion contains two elementary pieces of information, namely the length of the record and the length of the data block added to the original data. In the next step, the record data constructed consists of the following elements:

- Primary data.
- Some padding to complete the datagram as required.
- MAC value.

MAC is responsible for the verification of integrity of the message included in the transmitted record. A *secret-key* in creation of MAC is either a client write MAC secret or a server write MAC secret respectively, it depends on which party prepares the packet. After receiving the packet, the receiving party computes its own value of the MAC and compares it with that received. If the two values match, this means that data has not been modified during the transmission over the Network. The length of the MAC obtained in this way depends on the method uses for its computing. Next, the data plus the MAC are encrypted using a preset symmetric encryption algorithm. Both data and MAC are encrypted. This prepared data is attached with the following header fields:

- Content type: Identifies what payload is delivered by the packet to determine which higher protocols are to be used for processing of data included in the packet.
- Major version: Establishes the main portion of the protocol version to be used. For SSL 3.0 the value is 3.
- Minor version: Establishes the additional portion of the used version of the protocol. For SSL 3.0 the value is 0.

With the addition of the fields, the process of the record preparation is completed. Afterwards, the record is sent to the targeted point. The entire process of preparation of the packet to be sent is illustrated below.

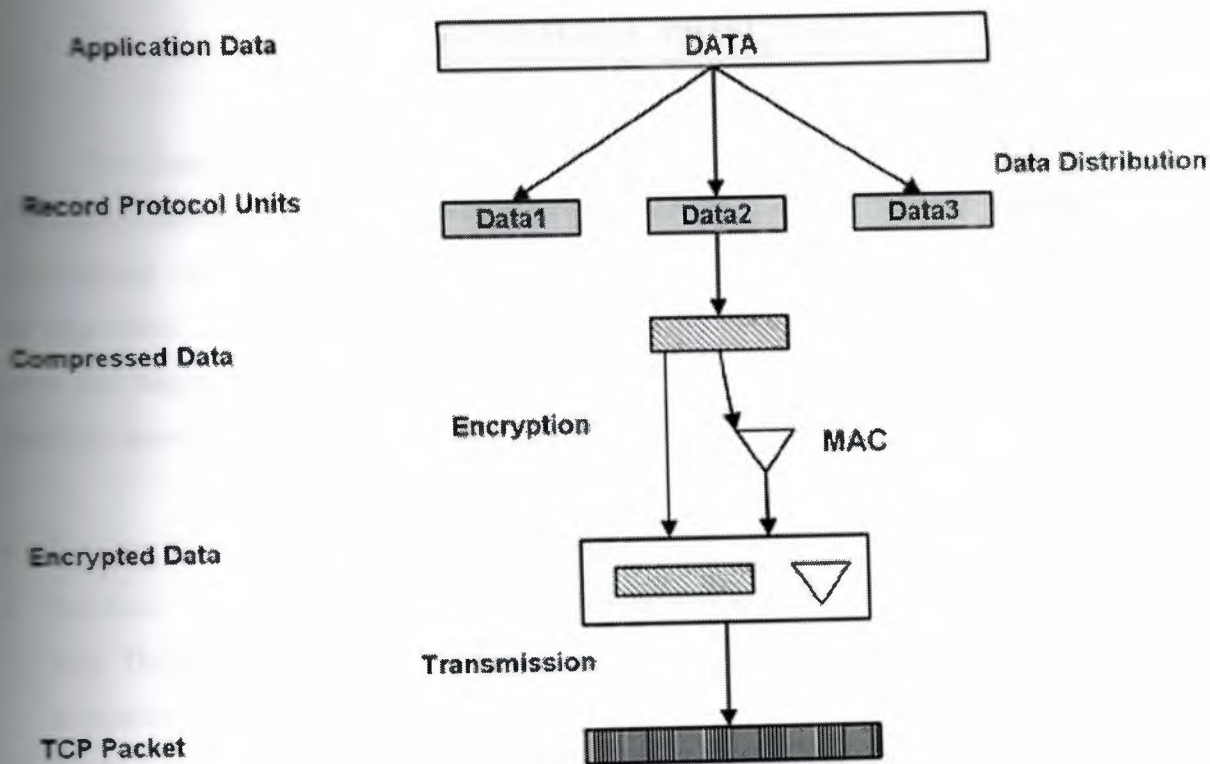


Figure 5.4. Creating a packet under SSL record protocol

The SSL record protocol is used to transfer any data within a session. Both messages and other SSL protocols (for example handshake protocol), as well as for any application data. The *Alert protocol* is used by parties to convey session messages associated with data exchange and functioning of the protocol. Each message in the alert protocol consists of two bytes. The first byte always takes a value, “warning”(1) or “fatal”(2), that determines the severity of the message sent. Sending a message having a “fatal” status by either party will result in an immediate termination of the SSL session. The next byte of the message contains one of the defined error codes, which may occur during an SSL communication session. The *ChangeCipher Spec protocol* is the simplest SSL protocol. It consists of a single message that carries the value of 1. The purpose of this message is to cause the pending session state to be established as a fixed state, which results, for example, in defining the used set of protocols. This type of message must be sent by the client to the server and vice versa. After exchange of messages, the session state is considered agreed. This message and any other SSL messages are transferred using the SSL record protocol.

5.6 SSL HANDSHAKE PROTOCOL

The Handshake protocol constitutes the most complex part of the SSL protocol. It is used to initiate a session between the server and the client. Within the message of this protocol, various components such as algorithms and keys used for data encryption are negotiated. Due to this protocol, it is possible to authenticate the parties to each other and negotiate appropriate parameters of the session between them. The client sends the server a client "hello" message containing data such as:

- Version: The highest version supported by the client.
- Random: Data consisting of a 32-bit timestamp and 28 bytes of randomly generated data. This data is used to protect the key exchange session between the parties of the connection.
- Session ID: A number that defines the session identifier. A nonzero value of this field indicates that the client wishes to update the parameters of an existing connection or establish a new connection on this session. A zero value on this field indicates that the client wishes to establish a new connection.
- CipherSuite: A list of encryption algorithms and key exchange method supported by the client. The server, in response to the client "hello" message sends a server "hello" message, containing the same set of fields as the client message, placing the following data:
 - Version: The lowest version number of the SSL protocol supported by the server.
 - Random Data: The same fashion as used by the client, but the data generated is completely independent.
 - Session ID: If the client field was nonzero, the same value is sent back. Otherwise the server's session ID field contains the value for a new session.
 - CipherSuite: The server uses this field to send a single set of protocols selected by the server from those proposed by the client. The first element of this field is a chosen method of exchange of cryptographic keys between the client and the server. The next element is the specification of encryption algorithms and Hash functions, which will be used within the session being initiated, along with all specific parameters.

The set of encryption algorithms and key exchange method sent in the *CipherSuite* field establishes three components:

1. The method of key exchange between the server and client.
2. The encryption algorithm for data encryption purposes.
3. A function used for obtaining the MAC value.

The server begins the next phase of negotiations by sending its certificate to the client for authentication. The message sent to the client contains one or a chain of X509 certificates. These are necessary for authentication of both the server and the certification path towards a trusted certification official of the certifying body for the server. This step is not obligatory and may be omitted, if the negotiated method of key exchange does not require sending the certificate. Depending on the negotiated method of key exchange, the server may send an additional server-key exchange message, which is however not required in the case when the fixed RSA key exchange technique has been negotiated. Moreover the server can request a certificate from the client. The final stage of phase 2 is the "server done" message, which has no parameters and is sent by the server merely to indicate the end of the server messages. After sending this message, the server waits for a client response. Upon receipt of the message, the client should verify the server's certificate, the certificate validation data and path, as well as any other parameters sent by the server in the "server hello" message. The client's verification consists of:

- Validation date check of the certificate and comparison with the current date, to verify whether the certificate is still valid.
- Checking whether the certifying body is included in the list of trusted *Certifying Authorities (CA)* in possession of the client. If the CA, which has issued the server's certificate is not included in the CA's list, the client attempts to verify the CA signature. If no information about the CA can be obtained, the client terminates the identification procedure by either returning the error signal or signalling the problem for the user to solve it.
- Identifying the authenticity of the public-key of the CA which has issued the certificate. If the certifying authority is included in the client's list of trusted CA's, the client checks the CA's public-key stated in the server's certificate with the

public key available from the list. This procedure verifies the authenticity of the certifying body.

- Checking whether the domain name used in the certificate matches the server name shown in the server's certificate.

Upon successful completion of all steps the server is considered authenticated. If all parameters are matched and the server's certificate correctly verified, the client sends the server one or multiple messages. Next is the client-key exchange message, which must be sent to deliver the keys. The content of this message depends on the negotiated method of key exchange. Moreover, at the server's request, the client's certificate is sent along with the message enabling verification of the certificate. This procedure ends phase 3 of negotiations.

Phase 4 is to confirm the message so far received and to verify whether the pending data is correct. The client sends a "change cipher spec" message (in accordance with the pending SSL ChangeCipher Spec), and then sets up the pending set of algorithm parameters and keys into the current set of the same. Then the client sends the finished message, which is first protected with just negotiated algorithms, keys and secrets. This to confirm that the negotiated parameters and data are correct. The server in response to the client sends the same message sequence. If the finished message is correctly read by either party, this confirms that the transmitted data, negotiated algorithms and the session-key are correct. This indicates that the session has been terminated and that it is possible to send the application data between the server and the client, via SSL. At this point the TCP session between the client and the server is closed, however a session state is maintained, allowing it to resume communications within the session using the retained parameters.

It is worth noticing that both phases 2 and 3 are used by both parties to verify the authenticity of the server's certificate and possibly the client's certificate during the Handshake step. If the server cannot be successfully authenticated by the client on the basis of the delivered certificate, the handshake terminates and the client will generate an error message. The same will occur at the server if the client's certificate authenticity cannot be confirmed. At first glance this process seems to be somewhat complicated, however this takes place at each connection with the server of an SSL-enabled service, for example while requesting the address of a site beginning with HTTPS://.

SSL is used in many services but mostly, SSL protects the HTTP communication channel over the Internet and therefore the SSL protocol is seen quite often as associated only with WWW pages. As it has been already mentioned, the SSL protocol can be used to protect the transmission for any TCP/IP service. Apart from the WWW accessing, the second most likely application of this protocol is associated with email sending and receiving.

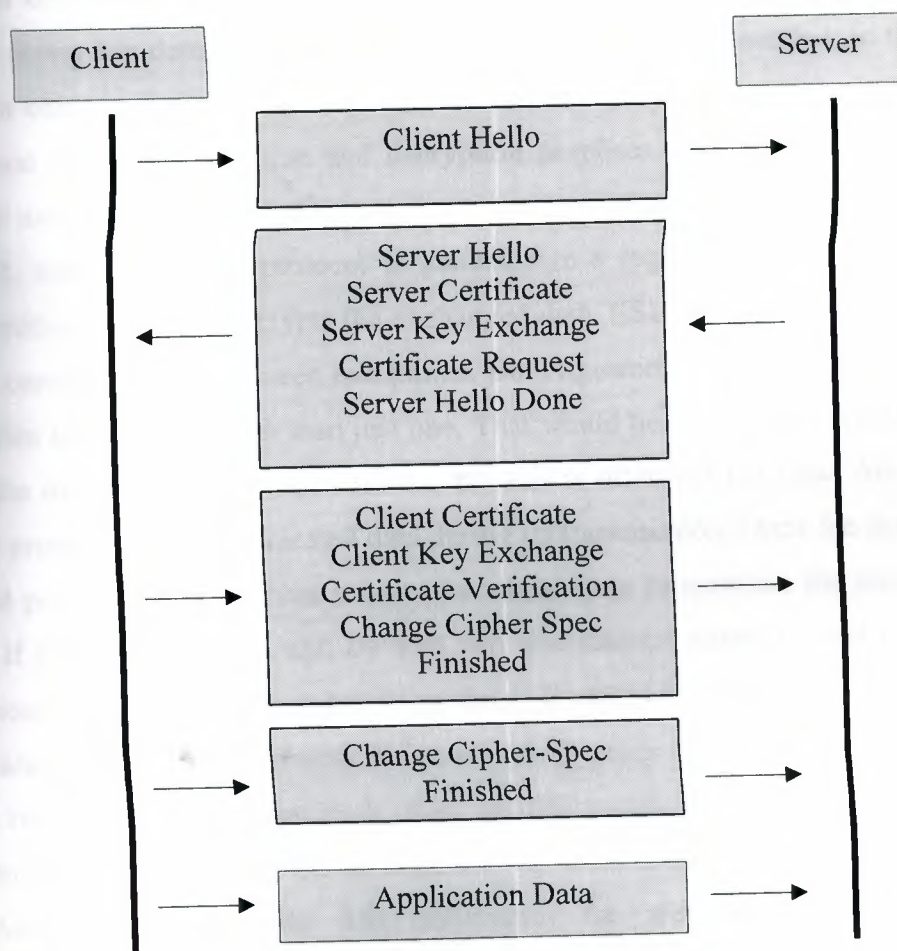


Figure 5.5. The SSL Handshake Protocol

Let us start again from what happens in short and make a summary, once you are already on a secure site the SSL Handshake protocol jumps into action. It sends browser SSL number, Encryption settings and other crypto information to the remote host. Once the remote server receives this it sends its SSL number back to the client. Then the client verifies the servers certificate authority. This done to ensure that the public key received by the client is that of the correct authentic server. Once the server identity has

been authenticated then the client creates a 'premaster secret password or key' which is unique for each new SSL session. This 'premaster password' is then encrypted using the client's public key and this encrypted 'premaster password' is sent to the server. When the server receives 'premaster password' it verifies the client identity. Once client identity has been authenticated, the server uses its private key to decrypt the 'premaster secret password' to obtain the 'master secret'. This master secret is used to determine the session key. Once all this is done the SSL record protocol comes into the picture. When the server has determined the 'symmetrical session key', it sends it to the client and further communication is done using this session key. As the key is symmetrical it can be used for both encryption and decryption purposes. The SSL record protocol handles all data transfers.

SSL uses Handshake protocol to authenticate a requester and responder with help of certificates. It also encrypts the exchanged data. SSL can only authenticate and encrypt a communication between two points. But requesting and responding to a web service often takes more routes than just one. That would be an adequate solution if the route of the message is known in advance, but this is often not the case. Additionally SSL only protects the communicated data during its transmission. Once the data arrives to the end point, the end host has to use other techniques to maintain the security. For example if two end points (A and D) with two intermediate points (B and C) want to communicate with each other, A has to go through B and C in order to get to D. In this case it cannot use SSL to authenticate itself and D, since SSL can only authenticate between two points. It could use encryption, but this requires intermediates B and C to be able to encrypt and decrypt the message in order to be able to process them.

Another problem with SSL techniques for web services is that SSL authentication and encryption consume a large amount of CPU time and consequently the transaction process is slowed down. Imagine how slow it would be for a server, to process several requests at the same time.

***MAC** means *message authentication code* that is used for transmission of data during the SSL session. This technique assumes that two communicating parties, say A and B, share a common secret key 'K'. When A has a message to send to B, it calculates the MAC as a function of the message and the key. The message plus MAC are transmitted to the intended recipient. The recipient performs the same calculation on the received message. Using the same secret key to generate a new MAC. The received MAC is

compared to the calculated MAC. If we assume that only the receiver and the sender know the identity of the secret key, and if the received MAC matches the calculated MAC, then the receiver is assured that the message has not been altered. If an attacker alters the message but does not alter the MAC, then the receiver's calculation of the MAC will differ from the received MAC. Because the attacker is assumed not to know the secret key, the attacker cannot alter the MAC to correspond to the alterations in the message. The receiver is assured that the message is from the alleged sender. Because no one else knows the secret key, no one else could prepare a message with a proper MAC. A MAC function is similar to encryption. One difference is that the MAC algorithm need not be reversible, as it must for decryption. It turns out that because of the mathematical properties of the authentication function, it is less vulnerable to being broken than encryption.

CHAPTER ONE: INFORMATION SECURITY

PART SIX: KERBEROS

6.1 WHAT IS KERBEROS?

Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret key cryptography. The Internet is an insecure place, many of the protocols used in the Internet do not provide any security. Thus, applications which send an unencrypted password over the network are extremely vulnerable. Some sites attempt to use firewalls to solve their network security problems. Unfortunately firewalls assume that the bad guys are on the outside, which is often a very bad assumption. Kerberos was created by Massachusetts Institute of Technology (MIT) as a solution to these network security problems. The kerberos protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection. After a client and a server have used kerberos to prove their identity, they can also encrypt all of their communications to assure privacy and data integrity as they go about their business. The name kerberos comes from Greek mythology, it is the three-headed dog that guarded the entrance to Hades.

Kerberos keeps a database of its clients and their private keys. The private key is a large number known only to kerberos and the client it belongs to. Because kerberos knows these private keys, it can create messages which convince one client that another is really who it claims to be. Kerberos also generates temporary private keys called session keys, which are given to two clients and no one else. A session key can be used to encrypt messages between two parties. Kerberos provides three distinct levels of protection. The application programmer determines which is appropriate, according to the requirements of the application. For example some applications require only that authenticity be established at the initiation of a network connection, and can assume that further messages from a given network address originate from the authenticated party. Other applications require authentication of each message, but do not care whether the content of the message is disclosed or not. For these kerberos provides safe messages. Yet a higher level of security is provided by private messages, where each message is

not only authenticated, but also encrypted. Private messages are used by the kerberos server itself for sending passwords over the network.

Encryption in kerberos is based on *DES*, the *Data Encryption Standard*. The encryption library implements those routines. Several methods of encryption are provided, with tradeoffs between speed and security. The kerberos database needs are straight forward; a record is held for each principal, containing the name, private key, and the expiration date of the principal, along with some administrative information. The authentication server (or kerberos server), on the other hand, performs read-only operations on the kerberos database, namely, the authentication of principals and generation of session keys.

6.2 HOW IT WORKS?

In kerberos authentication method when a user requests a service, his/her identity must be established. To do this, a ticket is presented to the server, along with proof that the ticket was originally issued to the user, not stolen. There are three phases to authentication through kerberos. In the first phase, the user obtains credentials to be used to request access to other services. In the second phase, the user requests authentication for a specific service. In the final phase, the user presents those credentials to the end server. There are two types of credentials used in the kerberos authentication model; *tickets and authenticators*. Both are based on private key encryption, but they are encrypted using different keys. A ticket is used to securely pass the identity of the person to whom the ticket was issued between the authentication server and the end server. A ticket also passes information that can be used to make sure that the person using the ticket is the same person to which it was issued. The authenticator contains the additional information which, when compared against that in the ticket proves that the client presenting the ticket is the same one to which the ticket was issued. A ticket is good for a single server and a single client. It contains the name of the server, the name of the client, the Internet address of the client, a lifetime, and a random session key. This information is encrypted using the key of the server for which the ticket will be used. Once the ticket has been issued, it may be used multiple times by the named client to gain access to the named server, until the ticket expires. Unlike the ticket, the authenticator can only be used once. A new one must be generated each time

a client wants to use a service. This does not present a problem because the client is able to build the authenticator itself. An authenticator contains the name of the client, the workstation's IP address, and the current workstation time. The authenticator is encrypted in the session key that is part of the ticket.

When a user walks up to a workstation, only one piece of information can prove his/her identity; the user's password. The initial exchange with the authentication server is designed to minimize the chance that the password will be compromised, while at the same time not allowing a user to properly authenticate himself without knowledge of that password. The user is prompted for her/his username. Once it has been entered, a request is sent to the authentication server containing the user's name and the name of a special service known as the *ticket-granting service*. The authentication server checks that it knows about the client. If so, it generates a random session key which will later be used between the client and the ticket-granting server. It then creates a ticket for the ticket-granting server which contains the client's name, the name of the ticket granting server, the current time, a lifetime for the ticket, the client's IP address, and the random session key just created. This is all encrypted in a key known only to the ticket-granting server and the authentication server. The authentication server then sends the ticket, along with a copy of the random session key and some additional information, back to the client. This response is encrypted in the client's private key, known only to kerberos and the client, which is derived from the user's password. Once the response has been received by the client, the user is asked for his/her password. The password is converted to a DES key and used to decrypt the response from the authentication server. The ticket and the session key, along with some of the other information, are stored for future use, and the user's password and DES key are erased from memory. Once the exchange has been completed, the workstation possesses information that it can use to prove the identity of its user for the lifetime of the ticket-granting ticket. As long as the software on the workstation had not been previously tampered with, no information exists that will allow someone else to impersonate the user beyond the life of the ticket.

In order to gain access to the server, the application builds an authenticator containing the client's name and IP address, and the current time. The authenticator is then encrypted in the session key that was received with the ticket for the server. The client then sends the authenticator along with the ticket to the server in a manner defined by the individual application. Once the authenticator and the ticket have been received by the server, the server decrypts the ticket, uses the session key included in

the ticket to decrypt the authenticator, compares the information in the ticket with that in the authenticator. The IP address from which the request was received, and the present time. If everything matches, it allows the request to proceed. It is assumed that clocks are synchronized to within several minutes. If the time in the request is too far in the future or the past, the server treats the request as an attempt to replay a previous request. The server is also allowed to keep track of all past requests with timestamps that are still valid. In order to further foil replay attacks, a request received with the same ticket and timestamp as one already received can be discarded. Finally if the client specifies that it wants the server to prove its identity too, the server adds one to the timestamp the client sent in the authenticator, encrypts the result in the session key, and sends the result back to the client. At the end of this exchange, the server is certain that, according to Kerberos, the client is who it says it is. If mutual authentication occurs, the client is also convinced that the server is authentic. Moreover, the client and server share a key which no one else knows, and can safely assume that a reasonably recent message encrypted in that key originated with the other party. Let's summarize that in a simple way with an example:

- I. First a user sends a message to the server that he/she needs to use a service of the server.
- II. When the server receives this message, it makes two copies of a brand new key. This is called the session key. It will be used in the direct exchange between user and service.
- III. It puts one of the session keys in *box 1* (box is just an encrypted message), along with a piece of paper with the name of the server written on it. It locks this box with the user's key.
- IV. It puts the other session key in a *box 2* (ticket), along with a piece of paper with the name of the user written on it. It locks this box with the service's key.
- V. It returns both the boxes to the user.
- VI. The user unlocks *box 1* with his key, extracting the session key and the paper with server name on it.
- VII. The user can't open *box 2* (since it's locked with the service key). Instead he puts a piece of paper with the current time written on it in *box 3* (authenticator), and locks it with the session key. He then hands both boxes to the service.

VIII. The service opens the box 2 with its own key, extracting the session key and the paper with the name of the user on it. It then opens box 3 with the session key to extract the piece of paper with the current time on it. These items demonstrate the identity of the user.

Recall that a ticket is only good for a single server. As such, it is necessary to obtain a separate ticket for each service the client wants to use. Tickets for individual servers can be obtained from the ticket-granting service. Since the ticket-granting service is itself a service, it makes use of the 'service access protocol' described previously. When a program requires a ticket that has not already been requested, it sends a request to the ticket-granting server. The request contains the name of the server for which a ticket is requested, along with the ticket-granting ticket and an authenticator built as described in the previous section.

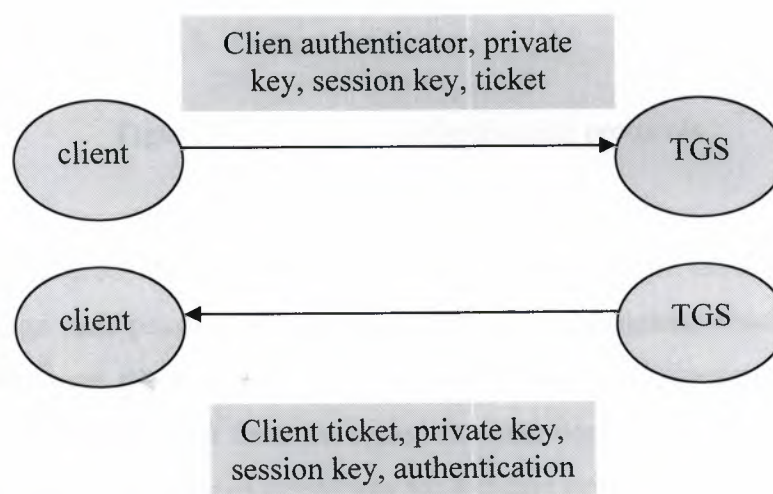


Figure 6.1. Getting a server ticket.

* TGS: Ticket-Granting Server.

The ticket-granting server then checks the authenticator and ticket-granting ticket as described above. If valid, the ticket-granting server generates a new random session key to be used between the client and the new server. It then builds a ticket for the new server containing the client's name, the server name, the current time, the client's IP address and the new session key it just generated. The lifetime of the new ticket is the minimum of the remaining life for the ticket-granting ticket and the default for the service. The ticket-granting server then sends the ticket, along with the session

key and other information, back to the client. This time, however, the reply is encrypted in the session key that was part of the ticket-granting ticket. This way, there is no need for the user to enter his/her password again.

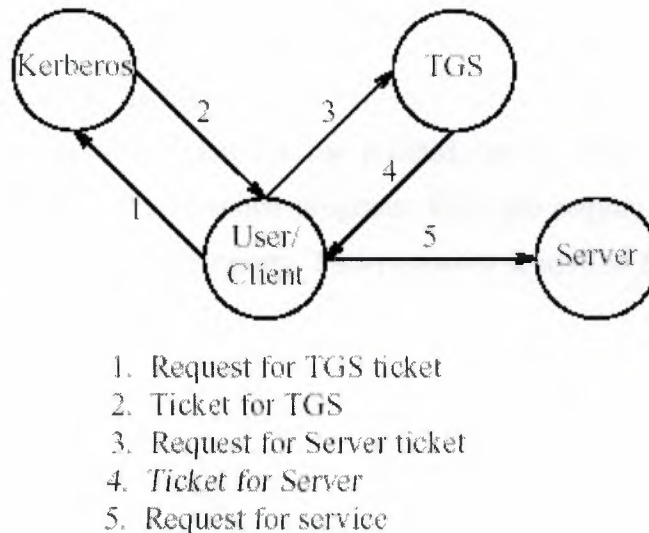


Figure 6.2. Kerberos authentication protocols.

The *Kerberos Database Management Service (KDBM)* handles requests from users to change their passwords. The client side of this program, which sends requests to the KDBM over the network, is the *Kerberos Password (kpasswd)* program. The KDBM also accepts request from kerberos administrators, who may add principals to the database, as well as change passwords for existing principals. The client side of the administration program, which also sends request to the KDBM over the network, is the *Kerberos Administration (kadmin)* program. The KDBM server accepts requests to add principals to the database or change the passwords for existing principals. This service is unique in that the ticket-granting service will not issue tickets for it. Instead, the authentication service itself must be used (the same service that is used to get a ticket-granting ticket). The purpose of this is to require the user to enter a password. If this were not so, then if a user left his/her workstation unattended, a passerby could walk-up and change his/her password for them, something which should be prevented. Likewise, if an administrator left his/her workstation unguarded, a passerby could change any password in the system. When a KDBM server receives a request, it authorizes it by comparing the authenticated principal name of the target of the request. If they are the

same, the request is permitted. If they are not the same, the KDBM server consults an access control list (stored in a file on the master kerberos system). If the requester's principal name is found in this file, the request is permitted, otherwise it is denied. All requests to the KDBM program, whether permitted or denied, are logged. Administrators of kerberos use the kadmin program to add principals to the database, or change the passwords of existing principals. An administrator is required to enter the password for their admin instance name when they invoke the kadmin program. This password is used to fetch a ticket for the KDBM server. Users may change their kerberos passwords using the kpasswd program. They are required to enter their old password when they invoke the program. This password is used to fetch a ticket for the KDBM server.

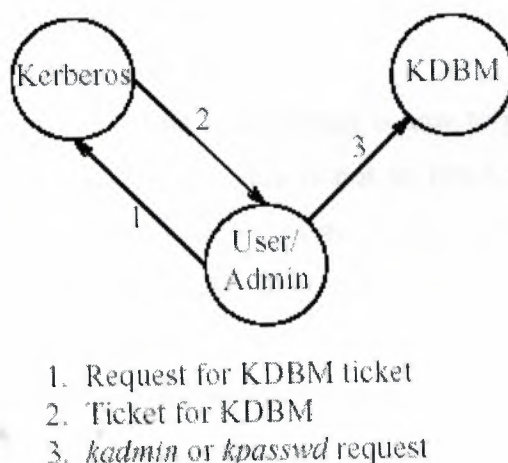


Figure 6.3. Kerberos administration protocol.

6.3 PROBLEMS WITH KERBEROS

There are a number of issues and problems associated with the kerberos authentication mechanism. Among the issues are; how to decide the correct lifetime for a ticket, how to allow proxies, and how to guarantee workstation integrity. The ticket lifetime problem is a matter of choosing the proper tradeoff between security and convenience. If the lifetime of the ticket is long, then if a ticket and its associated session key are stolen or misplaced, they can be used for a longer period of time. Such information can be stolen if a user forgets to log out of a public workstation.

Alternatively, if a user has been authenticated on a system that allows multiple users, another user with access to root might be able to find the information needed to use stolen tickets. The problem with giving a ticket a short lifetime, however, is that when it expires, the user will have to obtain a new one which requires the user to enter the password again.

An open problem is the proxy problem. How can an authenticated user allow a server to acquire other network services on his/her behalf? An example where this would be important is the use of a service that will gain access to protected files directly from a fileserver. Another example of this problem is what we call *authentication forwarding*. If a user is logged into a workstation and logs into a remote host, it would be nice if the user had access to the same services available locally, while running a program on the remote host. What makes this difficult is that the user might not trust the remote host, thus authentication forwarding is not desirable in all cases. We do not presently have a solution to this problem.

Another problem, and one that is important is how to guarantee the integrity of the software running on a workstation. This is not so much of a problem on private workstations since the user that will be using it has control over it. On public workstations, however, someone might have come along and modified the login program to save the user's password. the only solution presently available in our environment is to make it difficult for people to modify software running on the public workstations. A better solution would require that the user's key never leave a system that the user knows can be trusted. One way this could be done would be if the user possessed a smartcard capable of doing the encryptions required in the authentication protocol.

CHAPTER ONE: INFORMATION SECURITY

PART SEVEN: IP SECURITY



7.1 GOALS OF IP SECURITY

IP security (IP: Internet Protocol) is designed to provide high quality, cryptographically-based security for Internet. The set of security services offered includes access control, connectionless integrity, data origin authentication and confidentiality (encryption). These services are provided at the IP layer, offering protection for IP and/or upper layer protocols. These objectives are met through the use of two traffic security protocols, the *Authentication Header (AH)* and the *Encapsulating Security Payload (ESP)*, and through the use of cryptographic key management procedures and protocols. The set of IP security protocols employed in any context, and the ways in which they are employed, will be determined by the security and system requirements of users, applications or organizations. When these mechanisms are correctly implemented and deployed, they ought not to adversely affect users, hosts and other Internet components that do not employ these security mechanisms for protection of their traffic. A standard set of default algorithms are specified to facilitate interoperability in the global Internet. The use of these algorithms, in conjunction with IP security traffic protection and key management protocols, is intended to permit system and application developers to deploy high quality, Internet layer, cryptographic security technology. However the security offered by use of these protocols ultimately depends on the quality of their implementation. Moreover, the security of a computer system or network is a function of many factors, including personnel, physical, procedural and computer security practises. Thus IP security is only one part of an overall system security architecture. Finally, the security afforded by the use of IP security is critically dependent on many aspects of the operating environment in which the IP security is implementation executes. For example, defects in OS security, poor quality of management and practises can all degrade the security provided by IP security.

IP security provides the capability to secure communications across a LAN, WAN and Internet. Examples of its use include the following:

- Secure branch office connectivity over the Internet. A company can build a secure virtual private network over the Internet or over a public WAN.
- Secure remote access over the Internet. An end user whose system is equipped with IP security protocols can make a local call to an Internet service provider (ISP) and gain secure access to a company network.
- Establishing extranet and intranet connectivity with partners. IP security can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.
- Enhancing electronic commerce security. The use IP security enhances electronic commerce security.

The principal feature of IP security that enables it to support these varied applications is that it can encrypt or authenticate all traffic at the IP level. Thus all distributed applications, web access, email, file transfer and so on can be secured. The IP security networking device will typically encrypt and compress all traffic going into the WAN, and decrypt and decompress traffic coming from the WAN. When IP security is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter.

7.2 HOW IP SECURITY WORKS?

IP security provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithms to use for the services, and put in place any cryptographic keys required to provide the requested services. IP security can be used to protect one or more 'paths' between a pair of hosts, between a pair of security gateways, or between a security gateway (a router or a firewall implementing IP security is a security gateway) and a host.

IP security uses two protocols to provide traffic security; Authentication Header (AH) and Encapsulating Security Payload (ESP). The IP Authentication Header provides connectionless integrity, data origin authentication, and an optional anti-replay service. The Encapsulating Security Payload protocol may provide confidentiality (encryption) and limited traffic flow confidentiality. It also may provide connectionless integrity, data origin authentication, and an anti-replay service. Both AH and ESP are

vehicles for access control, based on the distribution of cryptographic keys and the management of traffic flows relative to these security protocols. These protocols may be applied alone or in combination with each other to provide a desired set of security. Each protocol supports two modes of use: *Transport mode* and *Tunnel mode*. In transport mode the protocols provide protection primarily for upper layer protocols, in tunnel mode, the protocols are applied to tunneled IP packets.

IP security allows the user (or system administrator) to control the offered security services. For example, one can create a single encrypted tunnel to carry all the traffic between two security gateways or a separate encrypted tunnel can be created for each TCP connection between each pair of hosts communicating across these gateways. IP security management must incorporate facilities specifying:

- Which security services to use and in what combinations.
- The algorithms used to effect cryptographic based security.

Because these security services use shared secret values (cryptographic keys), IP security relies on a separate set of mechanisms for putting these keys in place (the keys are used for authentication/integrity and encryption services) for automatic key management, but other automated key distribution techniques may be used (such as Kerberos, which is discussed later on).

There are several ways in which IP security may be implemented in a host or in conjunction with a router or firewall (to create a security gateway). A *Security Association (SA)* is a simplex connection that affords security services to the traffic carried by it. Security services are afforded to an SA by the use of AH or ESP but not both. If both AH and ESP protection is applied to a traffic stream, then two or more SA's are created to afford protection to traffic stream. To secure typical bi-directional communication between two hosts or between two security gateways are required. A security association is uniquely identified by a triple consisting of a *Security Parameter Index (SPI)*; The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed), an IP destination address, and a security protocol (AH or ESP) identifier. A security association is normally defined by the parameters, sequence number counter, sequence counter overflow, anti-replay window, AH information, ESP information, lifetime of this

security association, IP security protocol mode and maximum size of a packet that can be transmitted.

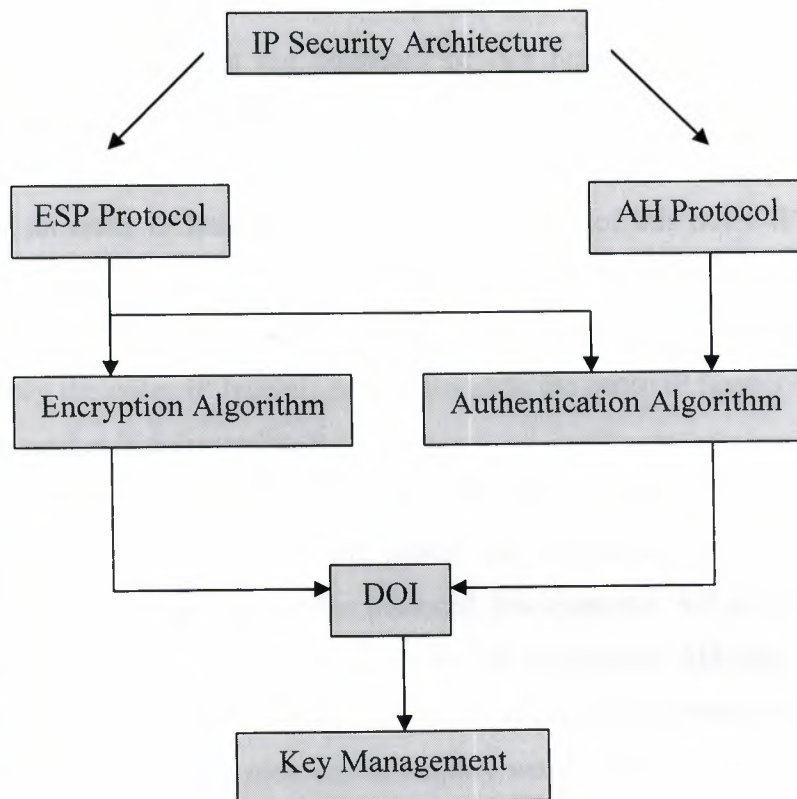


Figure 7.1. IP security architecture

Domain of Interpretation (DOI) contains values needed for the other documents to relate to each other. These include identifiers for approved encryption and authentication algorithms, as well as operational parameters such as key lifetime.

As noted before, two type os SA's are defined; Transport mode and Tunnel mode. Transport mode provides protection primarily for upper level protocols such as TCP or UDP. Typically transport mode is used for end-to-end communication between two hosts (e.g. a client and a server). Tunnel mode provides protection to the entire IP packet. To achieve this, after the AH or ESP fields are added to the IP packet. The entire packet plus security fields are treated as the payload of new "outer" IP packet with new outer IP header. The entire original, or inner packet travels through a "tunnel" from one point of an IP network to another, no routers along the way are able to examine the inner IP header. Because the original packet is encapsulated, the new larger packet may have totally different source and destination address, adding to the security.

Tunnel mode is used when one or both ends of an SA is a security gateway, such as a firewall or router that implements IP security. Here is an example of how tunnel mode IP security operates. Host A on a network generates an IP packet with the destination address of host B on another network. This packet is routed from the originating host to a firewall or secure router at the boundary of A's network. The firewall filters all outgoing packets to determine the need for IP security processing. If this packet from A to B requires IP security, the firewall performs IP security processing and encapsulates the packet in an outer IP header. The source IP address of this outer IP packet is this firewall, and the destination address may be a firewall that forms the boundary to B's local network. This packet is now routed to B's firewall, with intermediate routers examining only the outer IP header. At B's firewall, the outer IP header is stripped off, and the inner packet is delivered to B.

The set of security services offered by an Security Association (SA) depends on the security protocol selected, the SA mode, the endpoints of the SA, and on the election of optional services within the protocol. For example, AH provides data origin authentication and connectionless integrity for IP datagrams. AH also offers an anti-replay service at the discretion of the receiver, to help counter Denial of Service attacks (DoS). AH is an appropriate protocol to employ when confidentiality is not required. AH also provides authentication for selected portions of the IP header, which may be necessary in some contexts. For example, if the integrity of an IPv4 option or IPv6 extension header must be protected en route between sender and receiver, AH can provide this service. ESP optionally provides confidentiality for traffic and authentication.

The IP datagrams (data packets) transmitted over an individual SA are afforded protection by exactly one security protocol, either AH or ESP, but not both. Sometimes a security policy may call for a combination of services for a particular traffic flow that is not achievable with a single SA. In such instances it will be necessary to employ multiple SA's to implement the required security policy. The term "security association bundle" or "SA bundle" is applied to a sequence of SA's through which traffic must be processed to satisfy a security policy. The order of the sequence is defined by the policy. Security associations may be combined into bundles in two ways: transport adjacency and iterated tunneling. Transport adjacency refers to applying more than one security protocol to the same IP datagram, without invoking tunneling. Iterated tunneling refers to the application of multiple layers of security protocols effected

through IP tunneling. This approach allows for multiple levels of nesting, since each tunnel can originate or terminate at a different IP security site along the path.

Ultimately, a security association is a management construct used to enforce a security policy in the IP security environment. Thus an essential element of SA processing is an underlying *Security Policy Database (SPD)* that specifies what services are to be offered to IP datagrams and in what fashion. The form of database and its interface are outside the scope of this specification. The SPD must be consulted during the processing of all traffic, including non-IP security traffic. An SPD must discriminate among traffic that is afforded IP security protection and traffic that is allowed to bypass IP security. This applies to the IP security protection to be applied by a sender and to the IP security protection that must be present at the receiver. For any outbound or inbound datagram, three processing choices are possible: discard, bypass IP Security or apply IP security. The first choice refers to traffic that is not allowed to exit the host. The second choice refers to traffic that is allowed to pass without additional IP security protection. The third choice refers to traffic that is afforded IP security protection, and for such traffic the SPD must specify the security services to be provided, protocols to be employed, algorithms to be used, etc. For every IP security implementation there must be an administrative interface that allows a user or system administrator to manage the SPD. Specifically, every inbound or outbound packet is subject to processing by IP security and the SPD must specify what action will be taken in each case. Thus the administrative interface must allow the user to specify the security processing to be applied to any packet entering or exiting the system, on a packet by packet basis. In host systems, applications may be allowed to select what security processing is to be applied to the traffic they generate and consume. The SPD contains an ordered list of policy entries. Each policy entry is keyed by one or more selectors that define the set of IP traffic encompassed by this policy entry. Each entry includes an indication of whether traffic matching this policy will be bypassed, discarded or subject to IP security processing. If IP security processing is to be applied, the entry includes an SA specification, listing the IP security protocols, modes, and algorithms to be employed, including any nesting requirements. Because a security policy may require that more than one SA be applied to a specified set of traffic, in a specific order, the policy entry in the SPD must preserve these ordering requirements, when present. Thus, it must be possible for an IP security implementation to determine that an outbound or inbound packet must be processed through a sequence of SA's. When a packet is matched

against a SP entry and there is an existing SA or SA bundle that can be used to carry the traffic, the processing of the packet is controlled by the SA or SA bundle entry on the list. The SPD is used to control the flow of all traffic through an IP security system, including security and key management traffic from/to entities behind a security gateway.

IP security supports for both manual and automated SA and cryptographic key management. The IP security protocols, AH and ESP, are largely independent of the associated SA management techniques, although the techniques involved to affect some of the security services offered by the protocols. The simplest form of management is manual management, in which a person manually configures each system with keying material and security association management data relevant to secure communication with other systems. For example, a company could create a virtual private network (VPN) using IP security in security gateways at several sites. If the number of sites are small, and since all the sites come under the purview of a single administrative domain, this is likely to be a feasible context for manual management techniques. In this case, the security gateway might selectively protect traffic to and from other sites within the organization using a manually configured key, while not protecting traffic for other destinations. It also might be appropriate when only selected communications need to be secured. Widespread deployment and use of IP security requires an Internet standard, scalable, automated, SA management protocol. When an automated SA/key management protocol is employed, the output from this protocol may be used to generate multiple keys. The key management system may provide a separate string of bits for each key or it may generate one string of bits from which all of them are extracted.

The use of IP security imposes computational performance costs on the hosts or security gateways that implement these protocols. The costs are associated with the memory needed for IP security code and data structures, and the computation of integrity check values, encryption/decryption, and added per-packet handling. Use of SA key management protocols, especially ones that employ public key cryptography, also adds computational performance costs to use of IP security. The use of IP security also imposes bandwidth utilization costs on transmission, switching and routing components of the Internet infrastructure, components not implementing IP security. This is due to the increase in the packet size resulting from the addition of AH and/or

ESP headers, AH and ESP tunneling, and the increased packet traffic associated with key management protocols.

7.3 IP VERSION 6

The current version of the IP protocol is version 4, and the problems of authentication within *IPv4* is a challenging one. While firewalls, routers may help to stop hackers, they are add-ons designed to shore up the deficiencies in *IPv4*. They are not guarantees of security, and they involve significant efforts for each network that wants to implement them. According to those a new version of IP, called *IPv6* is designed. *IPv6* attempts to address the newfound knowledge of the importance of routing, security, etc. It promises to provide authentication and encryption on the Internet and could solve a lot of the existing problems with TCP/IP. *IPv6* includes two extension headers that serve as security options, the Authentication Header (AH) that we discussed before and encryption header. The AH allows the recipient to ascertain the identity of the sender and the encryption header ensures that only the recipient is able to look at the contents of the message. Both these options in *IPv6* require that the sender and the receiver agree on parameters such as the key, the authentication or the encryption algorithm, and the lifetime of the key. This is termed as a Security Association in *IPv6*. the recipient of a packet has to verify or decrypt only in the context of the security association between the sender and the receiver. The SA or the security context is identified using the Security Parameter Index (SPI). The SPI is normally negotiated as part of the key-exchange procedure. The SPI is usually chosen by the receiver for every (sender, receiver) pair. The receiver uses the SPI to identify the specific security context. In case of multi-casting, where there are multiple receivers, the SPI will be common to all the members of the multicast group. The SPI will be used by all the members to correlate the security context. The establishment of SA is crucial to both authentication and encryption. During a security association, the keys, other information about the keys such as the life-time of the key, the authentication and the encryption algorithms are exchanged between a receiver and a sender. In addition to this, this association context can be later referred to using the SPI, the security parameter index. Efficient deployment of *IPv6* security will rely on an efficient key distribution mechanism.

IPv6 uses the daisy chaining of options to support authentication as well. One of the extension header that can be sent after the base header could be an Authentication Header (AH). The AH has a very simple form as shown figure below. The SPI denotes the security context of the sender and the receiver. SPI would be used by the receiver to retrieve the security context of the sender. The sender computes a signature of the payload data, some fields of the IPv6 header and the extension headers, and this signature is sent as the authentication data. The algorithm used for the signature is negotiated as part of the key exchange process. The receiver retrieves the appropriate security context, using the SPI indicated and verifies the signature.

Next Header	Length	Reserved
Security Parameter Index (SPI)		
Authentication Data		

Figure 7.2. IPv6 authentication header

All data following the encryption header is encrypted, and the data preceding the encryption header is in plaintext. The format of the encryption header and the ESP header is shown below. The SPI is used by the receiver to retrieve the security context of the sender, and then decrypt the data following it.

IPv6 Base Header	Extension Headers	ESP Header	Encryption Header
------------------	-------------------	------------	-------------------

Figure 7.3. Encryption using the ESP header.

Security Parameter Index (SPI)
Encrypted Data and Parameters

Figure 7.4. The ESP header.

The Internet Protocol (IP) has been the foundation of the Internet and virtually all multivendor private internetworks. This protocol is reaching the end of its useful life and a new protocol, known as IPv6 has been defined to ultimately replace IPv4. the

driving motivation for the adoption of a new version of IP was the limitation imposed by the 32-bit address field in IPv4. In addition, IPv4 is a very old protocol, and new requirements in the areas of security, routing flexibility, and traffic support have developed. To meet these needs, IPv6 has been defined, and includes functional and formatting enhancements over IPv4. In addition, a set of security specifications have been issued that can be used with both IPv4 and IPv6. As IPv6 is gradually deployed, the Internet and corporate networks will be rejuvenated, able to support the applications of the 21st century.

TCP/IP as it exists today, has a general lack of security. Examples are; IP spoofing, SYN flooding, etc. show that this lack of security has lead directly to the development of tools and techniques to exploit TCP/IP weaknesses. Fixing some of these flaws today is possible (with add-ons like Kerberos) but these fixes are not always in widespread, everyday use your host may implement them, but host you want to communicate may not. Thus, most communication on today's Internet is still unsecured. Migrating to a new protocol suite, such as IPv6, may be the only answer to fixing the flaws in TCP/IP for good.

CHAPTER ONE: INFORMATION SECURITY

PART EIGHT: SSH

8.1 THE SECURE SHELL PROTOCOL

Secure Shell (SSH) is a standard for secure remote logins. Secure shell secures connections over the Internet by encrypting all transmitted confidential data, including passwords, binary files, and administrative commands. There are two versions of secure shell, secure shell version 2 (SSH2) that provides several security improvements as compared to the original secure shell version 1 (SSH1). The protocol versions are not compatible and it is recommended not to use SSH1 anymore. Secure shell was developed to solve the two most acute problems in the Internet, secure remote terminal logins and secure file transfers. Secure shell can also tunnel arbitrary TCP sessions over a single encrypted secure shell connection. *Tunneling* is a powerful feature that makes it possible to secure the communication of other applications and protocols without modifying the applications themselves. By using tunnels, users can continue to use existing insecure applications, such as email sessions, in a secure manner. With tunneling, secure shell can offer an encompassing solution for securing most of the communication tasks. SSH2 is an open and well documented standard, and SSH2 implementations of different vendors have been extensively tested for interoperability. The SSH2 protocol supports the use of the strongest available encryption algorithms, including the new U.S. *Advanced Encryption Standard (AES)*, and the old U.S. *Data Encryption Standard (DES)*. SSH2 provides data integrity by using *Hash Message Authentication Codes (HMAC)*.

SSH2 is a client-server protocol and both the client and the server are authenticated. The SSH2 protocol prevents the eavesdropping (listening) of passwords allowing password based authentication to be used safely. However, for some applications the protection offered by passwords cannot be considered secure enough. The average password contains only 28-bits of entropy, and a determined attacker may be able to crack weak passwords. For this purpose, the SSH2 protocol includes the possibility to use public-key authentication. Public-key authentication is based on public-key cryptography and offers strong security. The strongest authentication and the

best scalability is achieved by using public-key certificates. Certificates are actually an extension of the normal public-key authentication. When certificates are used, the secure shell user does not have to load the public-key separately to each server, but the server trusts the Certification Authority (CA) that has issued the certificate. To prevent man-in-the-middle attacks where a malicious party is impersonating as the server, also the server should be strongly authenticated. In the secure shell protocol, the server is authenticated with a digital signature based on a DSA or RSA public-key algorithm. Each server must have a public-private key pair. In implementations without support for certificates, clients refer to a local database of trusted server public keys.

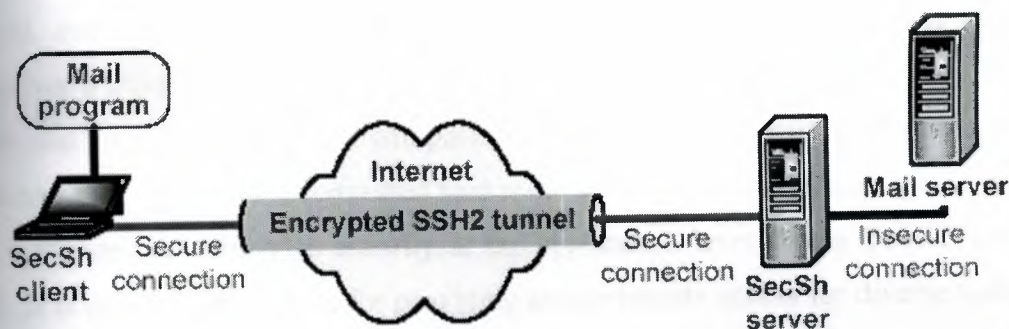


Figure 8.1. Example of Secure Shell usage; email security through SecSh tunnelling.

The main point of the SSH is to allow you to securely interact with remote machines. Before you actually do anything (run a session, transfer a file), there are three things that need to be done:

1. The client needs to establish that it is talking to the machine you asked it to, and not another machine that's spoofing it (or sitting in the middle accepting data and passing it on transparently).
2. The server on the remote machine may want to establish that you are connecting from the machine you appear to be, and not another machine that's spoofing it.
3. The client needs to convince the server that you are who you claim to be, and are authorised to do things on the server.

Host identification is done by using a technique known as RSA (public-key encryption algorithm) algorithm. Hosts that supports SSH arrange to have a host

identification key consisting of a public and private part. The private part of the key is secret and only known to the administrator of the machine, while the public part of the key is freely available. A message RSA encrypted with the public part of the key can only be decrypted with knowledge of the private part of the key. During this stage of the session setup, the server sends its public-key to the client, and the client encrypts a random session key with it and sends the result back to the server. The client clearly knows the session key as it generated it, and the server can use its private key to decrypt the message the client sent to it. The session key is used by whichever cipher is chosen for encryption. Since only the client and a server running on a machine that knows the secret half of the server's host key can know the session key, this both secures the session, and assures the client that it must be talking to the correct server machine. Whenever a client connects to the server, in response, the server sends a set of both public and host keys. The client compares the host public key with a master key. This procedure is to verify if the connected host is the same as the host in the first query.

Secure shell provides security at the application layer of the TCP/IP protocol stack. It is an application suite for providing secure remote access for diverse tasks in a flexible way, a versatile security solution that has become an essential tool in remote administration. Secure shell was developed to solve the two most acute problems in the Internet, secure remote terminal logins and secure file transfer. Telnet and FTP offer no protection for data and are easy targets for attacking. The primary goal of secure shell has been to replace these insecure protocols with a secure one. Secure shell utilizes client-server architecture; a server listens on a TCP port 22, which has been officially assigned for secure shell, and clients initiate connections to this port.

With SSH it is possible to create a secure communication channel between the server and the client. This channel can be used for different purposes. SSH supports variety of authentication methods, and new options may be added if required. Data in transit can be protected with multiple symmetric algorithms that use the key negotiated at the beginning of an SSH session. Both the client and the server can authenticate each other to enhance security against different kinds of attacks.

CHAPTER ONE: INFORMATION SECURITY

PART NINE: HACKERS&ATTACKS

9.1 MEANING OF BEING A HACKER

The word "*hacker*" means, a person who illegally gains access to and sometimes tamper with information in a computer system, unauthorized access to other computer system. A hacker is a person that loves to study all things in depth, especially the more apparently meaningless details, to discover hidden information, new features and weakness in them. A hacker has the tendency to use his skills also beyond of the computer context, and anywhere tends to use the hacking techniques and to discover what is normally hidden to the common man. For a hacker, the ability to reason, harness his full brain capacity and maintain his mind at maximum efficiency levels, is most important. A hacker is certainly a programming maniac, hackers often spend many day's and night's in front of a computer, programming and experimenting with new techniques. After spending so many hours in front of a computer, a hacker gains a remarkable ability to analyze large amounts of data very quickly. The ability to program quickly can be a characteristic of a hacker. As far as a hacker is concerned, it is faster to type on a keyboard, than it is to write things down, many hackers spend quite a lot of time analyzing previously written code, while they are programming. A hacker studies a system or a program to discover weaknesses, hidden features of it and then use them to go beyond its limits, with creativeness and imagination. The person with these skills can use his knowledge to try to access information to which he doesn't have the right to access. But perhaps more than anything else, curiosity and above average intelligence are the signatures of a true hacker, the hacker has an almost physical need of knowledge of any kind.

Hackers believe that essential lessons can be learned about the systems, about the world, from taking things apart, seeing how they work and using this knowledge to create new and even more interesting things. They resent any person, physical barrier or law that tries to keep them from doing this. A hacker is never satisfied with the default settings of a program or of the custom installations, he always has to open the configuration menu and set the options to get the maximum performance, and to make

the product work as close as possible to his way. He must be able to use, to modify and to check all the possible features of a program. Beside that, there are "dark-side" hackers, that don't have any respect of the hacker ethic and don't hesitate to perform actions meant to damage computer systems or other people. The dark-side hackers are accorded the same dignity and recognized as having the ability of a hacker, but their orientation makes them a dangerous element for the community. A more common definition, reserved for those that damage someone else's computer systems without drawing any benefit from it, it is that of malicious hackers. Hacking can be used like as a form of protest, breaking into and modifying the websites of very well known societies and government or military corporate entities, can be a way to make public certain injustices or violations of human rights. There are also paid-hackers in big companies, that try to find any open holes, weaknesses in the network of the company and report it to them.

As long as there have been computers, there have been hackers. The net is large and it generates it's own rules by itself, so there will be hackers in the future also.

9.2 COMMON ATTACKS:

9.3 DENIAL OF SERVICE ATTACKS

Denial of Service (DoS) is an attack designed to render a computer or network incapable of providing normal services. The most common DoS attacks will target the computer's bandwidth or connectivity. Bandwidth attacks flood the network with such a high volume of traffic, that all available network resources are consumed and legitimate user requests can not get through. Connectivity attacks flood a computer with such a high volume of connection requests, that all available operating system resources are consumed, and the computer can no longer process legitimate user requests. DoS attacks flood a remote network with an enormous amount of protocol packets. Routers and servers eventually become overloaded by attempting to route or handle each packet. Within minutes, network activities exponentially rises and the network stops responding to normal traffic and service requests from clients. Types of denial of service attacks include:

SYN Flood Attack: When a client attempts to contact a server service, the client and server exchange a series of messages. The client starts by sending a TCP connection request or SYN message to the server. The server responds to the SYN message with an acknowledgement ACK-SYN message. The client then acknowledges the server's ACK-SYN message with an ACK message. After these three actions take place, the connection between the client and server is open and they can exchange service specific data. The problem arises when the server has sent ACK-SYN message back to the client but has not yet received an ACK response from the client. This is now a half-open connection. The server keeps the pending connection in memory, waiting for a response from the client. The half-open connection in memory eventually will time out on the server, freeing up valuable resources again. Creating these half-open connections is accomplished with IP spoofing. The attacker's system sends a SYN message to the victim's server. These messages seem to be legitimate but in fact are references to a client system that is unable to respond to the server's SYN-ACK message. This means that the server will never be able to send an ACK message to the client computer. The server now has half-open connections in memory and eventually will fill up the server connections. The server is now unable to accept any new connections. The time limit on half-open connections will expire. However, the attacker's system keeps sending IP spoofed packets faster than the expire limit on the victim's server. In most cases the victim of such an attack will have difficulty accepting any new, legitimate incoming connections.

This type of attack does not really affect any of the current connections or outgoing connections. Normally it consumes an enormous amount of memory and processing power on the server, causing it to crash. The location of the attacking system is difficult to trace because the attacker's system address was masquerading as a legitimate IP address. This type of attack does not depend on the attacker being able to consume network bandwidth. In this case, the intruder is consuming valuable server resources. The implication is that an intruder can execute this attack from dial-up connection against a computer on a very fast network.

Distributed Denial of Service Attack: The intruder may also be able consume all the available bandwidth on a network by generating a large number of packets directed to the network. Typically, these packets are *Internet Control Message Protocol (ICMP)* echo packets, but in principle they may be anything. The ICMP is used to convey status

and error information including notification of network congestion and other network related problems. ICMP can be used to determine if a computer on the internet is responding. To do this, an ICMP echo request packet is sent to computer on the network. If the computer is operating, it will respond to the request by sending an ICMP echo reply packet. A common example of this is the ping command.

Three parties are involved in these attacks: the attacker, the intermediary, and the victim. The intermediary can also be a victim. The intermediary receives an ICMP echo request packet that is directed to the IP broadcast network address. When the attackers create these packets, they do not use their own IP source address. Instead they use the source address of their intended victim, this is known as IP spoofing. The result is that when the intermediary computers respond to the ICMP echo request packet, they send the replay packet to the victim's address. The victim's computer is now subjected to network congestion that could cause the network to stop responding. DDoS attacks involve breaking into hundreds or thousands of computers accross the internet. Then the attacker installs DDoS software on them, allowing the attackers to control all of these computers and launch coordinated attacks on victim sites. These attacks typically exhaust bandwidth, router processing capacity, or network stack resources, breaking network connectivity to the victims. The attacker starts by breaking into weakly secured computers, using well-known defects in standard network service programs, and common, weak configurations in operating systems. Then they perform some additional steps on each system, first they install software to break-in and to hide traces of their activities. for example, they replace the standard commands for displaying running processes with versions that fail to display the attacker's process. Then they install a special process used to remotely control the computer. This process accepts commands from over the Internet, letting intruder to launch an attack over the Internet against some designated victim site. Finally they make a note of the IP address of the computer they have taken over.

All these steps are highly automated. A cautious intruder will begin by breaking into just a few sites, then using them to break into some more, and repeating this cycle for several steps. By the time they are ready to mount the attacks, they have taken over thousands of computers and assembled them into a DDoS network. Once the attaker has installed the DDoS software, the attacker runs a single command that sends command packets to all the captured computers, instructing them to launch an attack against a specific victim. When the attacker decides to stop the attack, he or she sends another

single command. The controlled computer being used to mount the attacks send a stream of packets. For most of the attacks, these packets are directed at the victim computer. The first signs of an attack may be when thousands of compromised systems all over the world begin to flood the victim's network with traffic all at once. The first symptom is likely to be router crash, or something that looks a lot at one, traffic simply stops flowing between the victim and the Internet.

E-Mail Attack: Mail bombing is an email based attack. Email floods the attacked system until it fails. A system will fail in different ways, depending on the type of server and how it is configured. The email server accepts email messages until the disk where email is stored fills-up.

9.4 SOCIAL ENGINEERING

This is a common form of cracking. It can be used both by outsiders and by people within an organization. Social engineering is a hacker term for tricking people into revealing their password or some form of security information. A common example of social engineering would be where a hacker sends email to an employee, claiming to be an administrator who needs the employee's password to do some administrative work. The normal user who has not been taught about security might not know the difference between the actual administrator and the imposter administrator, especially in large organization. Other variations of this type of social engineering would be where someone claiming to be the administrator, phones a user and asks for the user's password and logon credentials. The user gives out the logon and password and the hacker now has full access. Another form of social engineering is guessing a user's password. When people can learn things about certain users personal or social lives, they can use this against them. For example, users might choose a daughter or son's name or birth date or friend's name as a password. Users also often use passwords that they can read on their desks in the work area. This gives the hacker a chance at guessing the password.

The basic goal of the social engineering is the same as hacking in general. Typical targets include big-name corporations, military and government agencies. Basically there are two categories of social engineering attack; computer/technology

based deception and human based deception. Technology based approach is to deceive the user into believing that he is interacting with 'real' computer system and get him to provide confidential information. The human approach is done through deception, by taking advantage of the victim's ignorance. It can take place in workplace, phone and even while online. A hacker can simply walk in door and pretend to be a maintenance worker or consultant who has access to the organization. Another technique is to watch the employee type in the password. Social engineering can also be done by phone, this is the most prevalent type of attack. A hacker will call-up and imitate someone in a position of authority or relevance. Help desks are particularly prone to this type of attack. Hackers are able to pretend that they are calling from inside the corporation by playing tricks on the company operator. Another way hackers may obtain information is by pretending to be network administrator, sending email through the network or sending a fake mail for any purpose. Email can also be used for more direct means of gaining access to a system. For instance, mail attachments sent from someone of authenticity can carry viruses, worms and trojan horses. A good example of this was an AOL hack. The hacker called AOL's tech support and spoke with the support person for an hour. During the conversation, the hacker mentioned that his car was for sale cheaply. The tech supporter was interested, so the hacker sent an email attachment with a picture of the car. Instead of a car photo, the mail executed a backdoor exploit that opened a connection out from AOL through the firewall.

→ In order to protect yourself from social engineers you should:

- Not to implicitly trust others without proper verification.
- Verify the identity or authority of any person making a request for information or action.
- Implement/use a difficult to guess password.
- Remember that friends are not always friends.
- Learn the methods used by social engineers to accomplish their objectives.

9.5 PING/TRACERT/NETSTAT

Ping: Now let's start with what exactly ping is. Ping is a part of the Internet Control Message Protocol (ICMP). ICMP is a protocol used to troubleshoot TCP/IP networks. Ping is a command which sends out a packet of data to the specified host. This specified host if online (turned on) sends out a reply or echoes off the same packet of data. If the packet of data that reaches back to your computer has the same packet of data that was sent then it means that host is online. So ping is basically a command which allows you to check if a host is alive or not. It can also be used to calculate the amount of time taken for a data to reach the host. It is so deadly that it can be used to ping a hostname which may even cause the host to crash. Now what happens is that when a host receives a ping signal, it allocates some of its resources to attend to or to echo back to packet of data. If you ping a host continuously, then a time will come when all resources of the host are used and the host either hangs or restarts. Hackers also use ping command to find their victim's IP address. Due to ping's deadly nature, most shell account ISP's hide the ping utility. To use the ping command:

- Enter to MS-DOS mode, at the DOS prompt type;
- Ping-hostname.where hostname stands for web address you would like to ping.
For example; ping yahoo.com.
- It will return to you with the server's IP address and amount of time taken for a data to reach the host.

Tracert: When you type hotmail.com in your browser, then your request passes through a large number of computers before reaching hotmail.com. to find out the list of servers your request passes through, you can use tracert command. Again like ping command simply type tracert at the DOS prompt:

```
C:\windows>tracert hotmail.com
```

Netstat: This is by far the most interesting hacking tool which gives some important information about your ISP (Internet Service Provider). Some important parameters used in Netstat is:

C:\windows>netstat -a → Displays all connections and listening ports. Ports open on your machine, your IP address, the IP of the host you are connected to and also the port of the host to which you are connected to.

C:\windows>netstat -e → Displays Ethernet statistics.

C:\windows>netstat -s → Displays per protocol statistics. By default statistics are shown for TCP, UDP and IP.

9.6 PORT SCANNING

What a port scanner does is, it checks a remote host for open ports, ports listening for a connection request or remote services etc. The importance of port scanning a system is to find out the services it has open. If an attacker knows what services a server has open, he can then research and try to find flaws for those services. Also attacker can do certain DoS attacks if he knows which ports are open. There are many port scanners, where can be downloaded from underground web sites.

Port scanning is one of the most popular technique hackers use to discover services they can break into. By port scanning the attacker finds which ports are available. Essentially a port scan consists of sending a message to each port, one at a time. The kind of response received indicates whether the port is used and can be therefore be probed further for weakness. The simplest port scan tries (e.g. sends a carefully constructed packet with a chosen destination port number) each of the ports 0...65535 on the victim to see which ones are open. Simple port scanning techniques listed below:

Strobe: A strobe does a narrower scan, only looking for those services the attacker knows how to exploit. The name comes from one of the original TCP scanning programs, though now virtually all scanning tools include this feature.

Stealth Scan: A stealth scan is a kind of scan that is designed to go undetected by auditing tools. Port scanners scan a host rapidly by firing off packets at different

ports. So, scanning very slowly (taking a day or more) becomes a stealth scan technique.

SYN Scanning: This technique is also called half-open scanning, because a TCP connection is not completed. A SYN packet is sent (to open a connection), and the target host responds with a SYN+ACK, this indicates that the port is listening, and an RST indicates a non-listener. The server process is never informed by the TCP layer because the connection did not complete.

FIN Scanning: The typical TCP scan attempts to open connections. Another technique sends packets at a port, expecting that open listening ports will send back different error messages than closed ports. The scanner sends a FIN packet, which should close a connection that is open. Closed ports reply to a FIN packet. Open ports, on the other hand, ignore the packet, this is required TCP behavior. If no services is listening at the target port, the operating system will generate an error message. If a service is listening, the operating system will silently drop the incoming packet. Therefore, silence indicates the presence of a service at the port. However since packets can be dropped accidentally on the wire or blocked by firewalls, this isn't a very effective scan.

Bounce Scans: The ability to hide their tracks is important to attackers. Therefore, attackers looking for systems they can bounce their attacks through. FTP bounce scanning takes the advantage of a vulnerability of the FTP protocol itself. It requires support for proxy FTP connections. This bouncing through an FTP server hides where the attacker comes from. A port scanner can exploit this to scan TCP ports from a proxy FTP server. Thus you could connect to an FTP server behind a firewall, and then scan ports that are more likely to be blocked. If the FTP server allows reading from and writing to a directory, you can send arbitrary data to ports that you do find open. If our target host is listening on the specified port, the transfer will be successful. The advantages to this approach are obvious (harder to trace, potential to bypass firewall). The main disadvantages are that it is slow, and that some FTP server implementations have finally disabled the proxy feature.

UDP Scanning: In order to find *UDP (User Datagram Protocol)* ports, the attacker generally sends empty UDP datagrams (packets), if the port is listening, the service should send back an error message or ignore the incoming datagram. If the port is closed, then most operating systems send back an "ICMP port unreachable" message. Thus, you can find out if a port is not open, and determine which ports are open. Neither UDP packets, nor the ICMP errors are guaranteed to arrive, so UDP scanners of this sort must also implement retransmission of packets that appear to be lost. Also this scanning technique is slow.

9.7 IP SPOOFING

IP spoofing is one of the most common forms of online camouflage. In IP spoofing, an attacker gains unauthorized access to a computer or a network by making it appear that a malicious message has come from a trusted machine (changing the source IP address) by "spoofing" the IP address of that machine. TCP/IP protocol has no way to check if the source IP address in the packet header actually belongs to the machine sending it. There are a few variations on the types of attacks that successfully employ IP spoofing. Although some are relatively dated, others are very pertinent to current security concerns.

Blind Spoofing: This is a more sophisticated attack, because the sequence and acknowledgement numbers are unreachable. In order to circumvent this, several packets are sent to the target machine in order to sample sequence numbers.

Non-Blind Spoofing: This type of attack takes place when the attacker is on the same subnet (communication subnetwork) as the victim. The biggest threat of spoofing in this instance would be session hijacking. This is accomplished by corrupting the datastream of an established connection, then re-establishing it based on correct sequence and acknowledgement numbers with the attack machine. Using this technique, an attacker could effectively bypass any authentication measures taken place to build the connection.

Man in the Middle Attack: In these attacks, a malicious party intercepts a legitimate communication between two friendly parties. The malicious host then controls the flow of communication and can eliminate or alter the information sent by one of the original participants without the knowledge of either the original sender or the recipient. In this way, an attacker can fool a victim into disclosing confidential information by "spoofing" the identity of the original sender, who is presumably trusted by the recipient.

9.8 BUFFER OVERFLOW

The principle of exploiting a buffer overflow is to overwrite parts of memory which aren't suppose to be overwritten by arbitrary input and making the process execute this code. A buffer overflow problem is based in the memory where the program stores its data.

'Buffers' are data storage areas, which generally hold a predefined amount of finite data. A buffer overflow occurs when a program attempts to store data into a buffer, where the data is larger than the size of the buffer.

For example imagine a 33 cl. glass. This is analogous to the buffer. This buffer (empty glass) can store 33 cl. of liquid (data). Now imagine that I wish to transfer a pint, which is about 47 cl., of beer from my full pint glass into the empty 33 cl. glass. As I begin to fill the glass (buffer) with beer (data), everything is fine until the end when beer begins to spill over the glass and onto the table. This is an example of an overflow. Clearly such an overflow is bad for beer and unfortunately, even worse if such vulnerabilities exist in computer programs.

When the data exceeds the size of the buffer, the extra data can overflow into adjacent memory locations, corrupting valid data and possibly changing the execution path and instructions. The ability to exploit a buffer overflow allows one to possibly inject arbitrary code into the execution path. This arbitrary code could allow remote control, system level access, giving unauthorized access to not only malicious hackers, but also to replicating malware. Buffer overflows are generally broken into multiple categories, based on both ease of exploitation and historical discovery of the technique. While there is no formal definition, buffer overflows are broken into three generations. First generation buffer overflows involve overwriting stack memory, second generation

overflows involve heaps, function pointers, and off-by-one exploits, and finally third generation overflows involve format string attacks and vulnerabilities in heap structure management.

First Generation Buffer Overflows: First generation buffer overflows involve overflowing a buffer that is located on the stack. For example the following program declares a buffer that is 256 bytes long. However the program attempts to fill it with 512 bytes of the letter 'A'.

```
int i;
void function(void)
{
    char buffer[256]; //create a buffer
    for(i=0;i<512;i++) //iterate 512 times
        buffer[i]='A'; //copy the letter 'A'
}
```

Overflow instead of filling the buffer full of A's, a classic exploit will fill the buffer with its own malicious code. The exploit utilized by name *CodeRed* was an example of first generation buffer overflow.

Second Generation Buffer Overflows: Errors in counting the size of the buffer can occur usually resulting in a single byte overflow known as an off-by-one. A "heap" is memory that has been dynamically allocated. This memory is logically separate from the memory allocated for the stack and code. Heaps are dynamically created and removed. Overflowing heaps can potentially overwrite data or modify pointers to data or functions. Another second generation overflow involves function pointers. A function pointer occurs mainly when callbacks occurs.

Third Generation Buffer Overflows: Third generation buffer overflows occur due to bad coding by software engineers/programmers. Thus by not following the exact specification, programmers can allow hackers to overwrite values in memory and execute arbitrary code. Therefore, most popular applications have been tested by security researchers for such vulnerabilities. Nevertheless, new applications are

developed constantly and unfortunately developers continue to use the format functions improperly, leaving them vulnerable.

9.9 BRUTE FORCE ATTACKS

A brute force attack used by a hacker to try all possible password combinations until the unknown password is found and authentication is approved. Did you know that if a password consists of 4 numbers, then there are $10 \times 10 \times 10 \times 10$ possible combinations, that is 10,000 combinations? In that case the brute force attack is very simple, testing all possible combinations. When the key is long and complicated using brute force is pointless, your computing power will not suffice, and you have to look for other weak links in the system if you want to force your way in. Many times a brute force attack finds ways to eliminate a lot of possible combinations and the actual attack on the remaining few combinations. A good PC can manage 1,000,000 combinations per second today. Brute force cracking programs will attempt to crack the password using every combination of numeric, alphabetic and special characters available, no matter how long it takes. In other words, given enough time, brute force cracking will eventually determine the password, whether done offline or online on a system where no account policies have been set. If the intruder simply tries every combination against the system while online, most good systems will eventually lock out the account. However if the attacker manages to get a copy of the passwords in encrypted format, the attacker can then copy the file to another location on his/her own system and then take as much time as needed to crack the password using brute force offline.

A brute force attack typically involves a dictionary attack against a password file although an individual password can also be a target. While most administrators establish a maximum password attempt before lock out, brute force attacks are successful since the password or password file is captured and attacked-off of the target system. While password files are often the target, individual passwords can be obtained using sniffers (discussed later). Brute force attacks are successful since passwords are frequently found in dictionaries or are modified in predictable ways by inserting numerics. Brute force password tools can be set up to check for these variations. It is not uncommon to be able to crack as many as 90% of the passwords used in an enterprise within minutes. Individual passwords are often compromised when some

knowledge of the password owner is known. People tend to use passwords that are associated with them making them relatively easy guess.

9.10 SNIFFING

To capture the information going over the network is called sniffing. Sniffing is observing the activity of one's victim on a network (usually the Internet). This can include stealing passwords, reading email, and etc. Sniffing can be done by many popular sniffing programs. Sniffing programs have been around for a long time in two forms. Commercial packet sniffers, are used to help maintain networks. Underground packet sniffers are used to break into computers. Typical uses of both two sniffer programs include:

- Conversion of data to human readable format so that people can read the traffic.
- Fault analysis to discover problems in the network, such as why computer A can't talk to computer B.
- Network intrusion detection in order to discover hackers.
- Network traffic logging, to create logs that hackers can't break into.
- And of course to get passwords/usernames to break into systems.

9.11 INTRUSION DETECTION SYSTEMS

An intrusion detection system is a system that attempts to discover, alert, and possibly respond to an instance of undesired access. Intrusion detection products are software and hardware products designed to monitor a device or network for malicious activity. An intrusion can be defined as the attempted use of a system without authorization, an attempt on the part of the legitimate user to abuse or increase her privileges on a system without authorization, or an attempt to deny access to a computer system or application. There are a wide variety of intrusion detection systems currently available or being researched. These include network based, host based, hybrid, and honeypots or honeynets.

A *network based* intrusion detection system usually consists of a passive sensor machine which has a network card configured in promiscuous mode, enabling it to

collect all of the network traffic that passes along its segment of the network. Network intrusion detection systems are dedicated software systems that sit on a network wire and analyze network packets. The data encapsulated in these packets is compared to a database of known attack signatures. If the data passing along the network does not match a known attack listed in the database, then the traffic continues without suspicion. However if the packet data matches a known attack, then some sort of response may be generated. Network based intrusion detection systems are generally easier to deploy than other intrusion detection systems, because they can monitor many hosts from a single location. Network based intrusion detection systems are limited to detecting the traffic that is on their network segments.

Host based systems run on the host they are monitoring. In general, they watch the application and system logs for attacks. In addition to monitoring log files, host based agents might monitor accesses and changes to critical system files and user privilege. Administrators search through log files at the end of the day to detect any suspicious activity that had occurred during a defined time period. As the event is logged to a log file, the local software agent installed on the resource checks to see whether that event matches any of those listed in its attack database. Host based intrusion detection can also include monitoring of critical system files. System files can be monitored by keeping a record of checksums for installed files and periodically verifying the recorded value with the value reported by the file on disk.

Hybrid intrusion detection systems combine the functionality of the network and host based systems using both traffic and log files for data. These systems usually consist of distributed sensors that monitor a host or a network segment, and report alerts back to a central monitoring console.

Honeypots and Honeynets take a different task than standard intrusion detection systems. A honeypot is a resource whose value is being probed, attacked, or compromised. Generally, a honeypot is a non-production system placed in a carefully monitored location accessible to the open Internet. Honeypots can display different levels of service depending on the intent of their managers, but should always be configured such that the compromise of the system or applications running on the honeypot cannot be used as the source of an intrusion into another system. This means that the honeypot should reside behind a firewall configured to contain traffic leaving the honeypot. Honeypots include full operating systems which actually perform no business function but which run modified kernel modules that are able to monitor and

record all activity on a system. In another variation, they might execute on a well known port a simple state machine which simulates expected initial connection conditions so activity can be recorded and an attack analyzed. The use of honeypots gives the ability to track attacker attempts at entry and respond before they come across a vulnerability we are susceptible to. Honeynets are virtual or physical networks of honeypots.

Using a honeypot has numerous advantages. First, it wastes the attacker's time. Depending on the depth of the attack, an intruder can spend large amounts of time attempting to explore and exploit the honeypot. Second, it gives the attacker a false impression of the existing security measures. He or she might spend time finding tools to exploit the honeypot that might not work on a real system. Third, the existence of a honeypot decreases the likelihood that a random probe or attack will hit a real machine. Finally, it provides extremely detailed information about the attacker's processes and methods which can be used to enhance existing security measures such as firewall rules. Honeypots are also capable of detecting attacks that other forms of intrusion protection systems are not. New vulnerabilities can be found and analyzed because all actions an attacker takes are recorded. Since all communication with a honeypot is a suspect, new attack tools can be detected based on their interaction. Honeypots can be classified into three primary categories; target servers, facades, and instrumented systems.

Target Servers: A target server is a computer deployed with the sole purpose of being attacked. It usually consists of an off-the-shelf system placed in a vulnerable location open to attacks and intrusions. It can be built from virtually any device. A typical implementation involves loading the operating system, configuring some applications, and leaving it on the network to see what happens. The administrator examines the system periodically to determine if it has been compromised and, if so, what was done to it.

Facades: A facade is a system which provides a false image of a target device or host. When a facade is probed or attacked, it gathers information about the attacker. The depth of the simulation varies depending on the implementation. Facades are not real systems, they emulate systems rather than providing a full set of capabilities.

Instrumented Systems: Instrumented systems provide an ideal compromise between the low management cost of a facade and the depth of detail provided by a

target server. An instrumented system with modifications provide more information, containment, or control.

9.12 CYBERTERRORISM

Cyberterrorism is the convergence of terrorism and cyberspace. It is generally understood to mean, attacks and threats against computers, networks and the information stored areas against government for political or social objectives. Further to qualify as cyberterrorism, an attack should result in violence against persons or property, or at least cause enough harm to generate fear. Attacks that lead to death, injury, explosions, plane crashes or severe economic loss would be examples. Serious attacks against critical infrastructures could be acts of cyberterrorism, depending on their impact. Attacks that disrupt non-essential services or that are mainly costly nuisance would not. While a computer cannot act as the sole perpetrator of a terrorist event, the computer can radically alter interpersonal interactions between people. Anonymization can make recruiting easier, and virtual identities can influence group dynamics. Additionally, the Internet complicates the issues of national, subnational and international groupings. The computer can aid the terrorist in many other ways as well. For example, email can be used for messaging to each other. The web provides a powerful information gathering tool and arena for identity theft. Individuals and groups are no longer confined to meeting with other individuals in their own countries, the possibilities are almost endless. As we can see, computers can play a huge role in any terrorist event, whether or not it takes place in the virtual world. The functional tasks of the terrorist group having a WWW presence may be distributed among several sites, it is relatively easy for a terrorist organization to solicit funds for operations via the WWW (Ecommerce), promote their cause and finally for those accessing information via the WWW also helps distance sympathizers to join or help them (a common example is September 11 attack).

Defending against terrorism where a computer or the Internet plays an important part in the terrorism is very similar to defending against terrorism that does not. The regular practises (law, defense, diplomacy, etc.) are still effective, except that the scope of certain elements is expanded. Governments can use their power to make terrorism too costly for those who seek to use it. They can do this by military strikes against terrorist

forces, collective punishment or other methods. The Internet was developed primarily as an unregulated, open architecture, computers can play an enormous role in terrorism, at the same time they can provide perhaps our biggest defense against terrorism if used to our advantage.

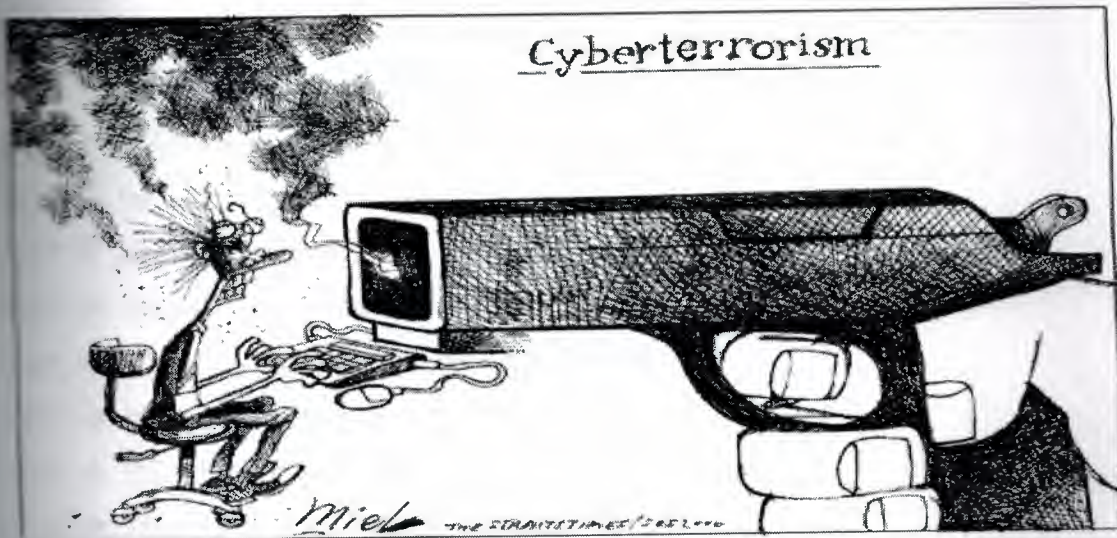


Figure 9.1. Cartoon about Cyberterrorism

9.13 HACKER INCIDENTS

- 1983 → A famous hacker (Mitnick) arrested for gaining illegal access to the Pentagon's website.
- 1983 → Fred Cohen creates the first virus at University of Southern California (Vax virus, called VD).
- 1986 → Chaos computer clubs cracked German government computer that had info about Chernobyl.
- 1986 → Hackers break into Stanford computers, which is very famous university in USA.
- 1987 → Hackers accessed to AT&T computers, stealing more than \$1 million.
- 1987 → Chaos computer club hacks NASA's network.
- 1987 → Jerusalem memory resident virus; 1st file-infecting virus.
- 1990 → Hackers break into British clearing banks.
- 1991 → Deutsch hackers break into US military computers .
- 1991 → 1,000 viruses exist in the world.

- 1994 → Citibank hacked by a hacker, \$10 million in illegal transfers.
- 1994 → Hackers cracked into Pentagon, altering and erasing records.
- 1995 → A very famous hacker Mitnick arrested (had 20,000 credit card numbers).
- 1995 → Hacker cracked "New York Times" Internet service, bringing it down.
- 1995 → Famous hacker Mitnick captured; had 20,000 credit card numbers.
- 1996 → NYPD (New York Police Department) voice mail system hacked.
- 1996 → Cambridge University hacked, confidential files broken into.
- 1996 → CIA web page hacked by 5 Swedish hackers.
- 1996 → 1st known Excel virus, Laroux.
- 1997 → Hackers attacked LAPD web site (Los Angeles Police Department)
- 1997 → NASA web site hacked.
- 1997 → Altavista homepage got hacked (altavista.com).
- 1997 → Hackers hit Coca-Cola web site.
- 1997 → RSA's 56-bit encryption key cracked.
- 1998 → Janet Jackson and Rolling Stones web site hacked.
- 1998 → BMW web site hacked.
- 1998 → UNICEF homepage hacked.
- 1998 → Citizens Bank homepage hacked.
- 1998 → Military sites (.mil and some .gov) hacked – 200 servers.
- 1998 → US. Army Air Defense Artillery School hacked.
- 1998 → University of Minnesota hacked.
- 1998 → Indian nuclear research center servers hacked.
- 1998 → Time Warner systems hacked.
- 1998 → 180,00 passwords stolen, 48,00 cracked by "Jack the Ripper" program.
- 1998 → Paramount web site hacked.
- 1998 → Australian government's web site hacked.
- 1998 → NY Times hacked.
- 1998 → 24,000 domains hacked; hosted by japan.co.jp
- 1999 → Melissa macro virus affected 100,000 email users.
- 1999 → Chernobyl CIH virus released.
- 1999 → Hack against White House web site (whitehouse.com).
- 1999 → Hack against FBI web site.
- 1999 → Hack against US senate web site.
- 2000 → Yahoo, ebay, amazon, buy.com, CNN hit with a Distributed DoS attack.

- 2000 → Internet.com hijacked.
 - 2000 → Nike site hacked.
 - 2000 → 55,000 credit card numbers stolen from creditcards.com
 - 2000 → Love Bug virus sent from Philippines.
 - 2000 → Microsoft hacked.
 - 2001 → Code Red, Nimda Worm.
 - 2001 → Whitehouse.gov hacked.
 - 2001 → Nimda worm released.
 - 2002 → New York times again hacked.
 - 2002 → Klez virus.
 - 2003 → Slammer worm; fastest worm in history.
 - 2003 → Bugbear worm; tries to steal passwords and credit card information.
 - 2003 → Sobig worm; \$50 million damages.
 - 2003 → Anti-spam sites hacked by spammers.
-

CHAPTER ONE: INFORMATION SECURITY

PART TEN: SPAM&SPYWARE

10.1 SPAM EMAILS

Spam; the junk mail of the Internet or in other word unwanted emails. Most of us get spam every day, some of us get a little, and some of us get a lot, but if you have an email account it is always there. Spam is incredibly annoying, especially in large quantities. If you have a public email address you can receive hundreds of spam messages for every legitimate message that arrives. Even with good filters, some of the spam makes it through. And filters can sometimes delete messages that you really do want to receive.

In a single day in May [2003], no 1 Internet service provider AOL time warner blocked 2 *billion* spam messages, 88 per subscriber, from hitting its customers email accounts. Microsoft which operates no 2 Internet service provider MSN plus email service Hotmail, says it blocks an average of 2.4 billion spams per day.

One of the problems with spam, and the reason why there is so much of it, is that it is so easy to create. You could easily become a spammer yourself. Lets say that you have a recipe from your mother for best posters ever created. A friend suggests that you sell the posters for \$5 each. You decide that your friend might be on to something, so you send an email to the 100 people in your personnal email address book with the subject line, "best posters with best price", your email contains a link to your web site where you can sell posters. As a result of your 100 emails, you get two orders and make \$10. "Wow!" you think, "It cost me nothing to send those 100 emails, and I made \$10. If I sent 1000 emails I could make \$100. If I sent million emails I could make \$100,000! I wonder where I could get a million email addresses...". As it turns out, there are hundred of companies that will sell you CD's filled with millions of valid email addresses. Thats the way how spammers work and send us spam mails every day. This is the problem with spam. It is incredibly easy for anyone to send it. It costs particularly nothing to send it.

10.2 HOW DO THEY GET MY ADDRESS?

Where does a company get millions of valid email addresses to put on a CD and sell it to spammers? There are number of primary sources. The first is newsgroups and chat rooms, especially on big sites like AOL. People often use their first screen names, or leave their actual email addresses in newsgroups. Spammers use pieces of software to extract the screen names and email addresses automatically. The second source for email addresses is the web itself. There are tens on millions of web sites, and spammers can create search engines that spider the web specifically looking for the telltale "@" sign that indicates an email address. The programs that do the spidering are often called *spambots*. The third source is sites created specifically to attract email addresses. For example, a spammer creates a site that says, "win \$1 million!!! Just type your email address here", in the past lost of large sites also sold the email addresses of their members. Or the sites created "opt-in" email lists by asking, "would you like to receive email newsletters from our partners?", if you answered yes, your address was than sold to a spammer. Probably the most common source of email addresses is a *dictionary search* of the email servers of large email hosting companies like AOL, MSN or Hotmail. A dictionary attack utilizes software that opens a connection to the target mail server and then rapidly submits millions of random email addresses. Many of these addresses have slight variations such as "jack_abc@hotmail.com" and "jack_bcd@hotmail.com". The software than records which addresses are "live/exists", and adds those addresses to the spammers list. These lists are typically resold to other spammers. Email addresses generally are not private. Once a spammer gets hold of your email address and starts sharing it with other spammers, you are likely to get a lot of spam.



Figure 10.1. Example of spam message

Table 10.1. 8 ways spammer can get your email address

8 ways spammers can get your email address.	
1)	From user registrations
2)	From user newsgroup postings
3)	From user chat sessions
4)	From spambots that crawl the web for any @ sign.
5)	From email lists the spammer buys.
6)	From mailing lists to which users subscribe.
7)	By randomly generating name combinations for your domain.
8)	By stealing all the email addresses on your company's server.

10.3 STOPPING SPAM

The best technology that is currently available to stop spam is *spam filtering softwares*. The simplest filters use keywords such as “sex”, “xxx”, “viagra”, etc., in the subject line to attempt to identify and delete spam. These simple filters are easy to sidestep by spelling “sex” as “s-e-x”, there are of course, thousands of ways to spell

"sex" if you are willing to add extra characters like that, and it is difficult for the simple filters to keep up. Also simple filters are most likely to block "real" email that you do want to receive. For example, if your friend sends you her favorite recipe for baked chicken breasts, the filter blocks the email because of the word "breasts". More advanced filters, known as "heuristic filters" try to take this simple approach quite a bit further to statistically identify spam based on word patterns or word frequency. But there are still ways to get around them (mainly by using short messages). Large ISP tried blocking multiple emails with the same subject line or message body. This had the unwanted side effect of blocking email newsletters, so ISP's made "white lists" to identify legitimate newsletter senders. Then spammers sidestepped the issue by inserting different random characters into each subject line and message body. That's why you get email messages with subject lines like: "Women Wanted *puklq*". The word "*puklq*" is random, and it is different on every email the spammer sends. There are several organizations that publish lists of IP addresses that are used by spammers. Any large spammer will have an array of server machines blasting out spam messages, and each server machine has its own IP address. Once spam is detected from an IP address, that IP address is put in a list. Companies that host email accounts can look at the sending IP address of every email and filter out those that appear in the list. Spammers get around this approach in two different ways. First, they change their IP addresses frequently. The unfortunate problem with this approach is that the old IP addresses that spammers discard get recycled, and the people who get these discarded IP addresses find them to be useless, they are tainted by their former association with spam, and cannot be used for sending legitimate email. Lately, spammers have started to get more aggressive. For example, it is thought that recent viruses like SoBig.F were sent out specifically to recruit "zombie machines" for spammers. The *zombie machines* are generally personal computers owned by unsuspecting private citizens who happened to contract the SoBig virus. The virus opens their machines for spammers, who can then route spam emails through their machines. Since the IP addresses of these machines are new, they do not appear in the IP address blacklists and millions of spam emails can route through them before they get blacklisted. Another front in the war against spam is legislation. For example, it has been suggested that the U.S. Federal Government set-up a national "do not spam" list identical to the national "do not call" list designed to block telemarketers. However, it is believed by most people that spammers are so clever that

they would set up spam servers in foreign countries and actually use the “do not spam” list as a source of fresh email addresses.

→ You can minimize the spam by doing the following:

- ❖ Report it and then delete it. Never reply through email or go to a web site, because this simply confirms your address and adds your address to more lists.
- ❖ Use an email filter offered by your provider or available in your email program. The greatest protection is to only receive messages from specific addresses that you trust. Otherwise, you can also usually filter emails based on keywords, domain names, or other measures.
- ❖ Use a public email address to hide your real/true email address. For instance, you may post to newsgroups with an email address like address@yourisp.com, rather than your real email address.
- ❖ Read the privacy statement for those companies that collect your email address. If this policy doesn’t exist, you should be concerned. In addition, check this policy to ensure your information is not provided to anyone else without your permission.

→ Lets look at a typical spam message and examine the methods used to identify and block it:

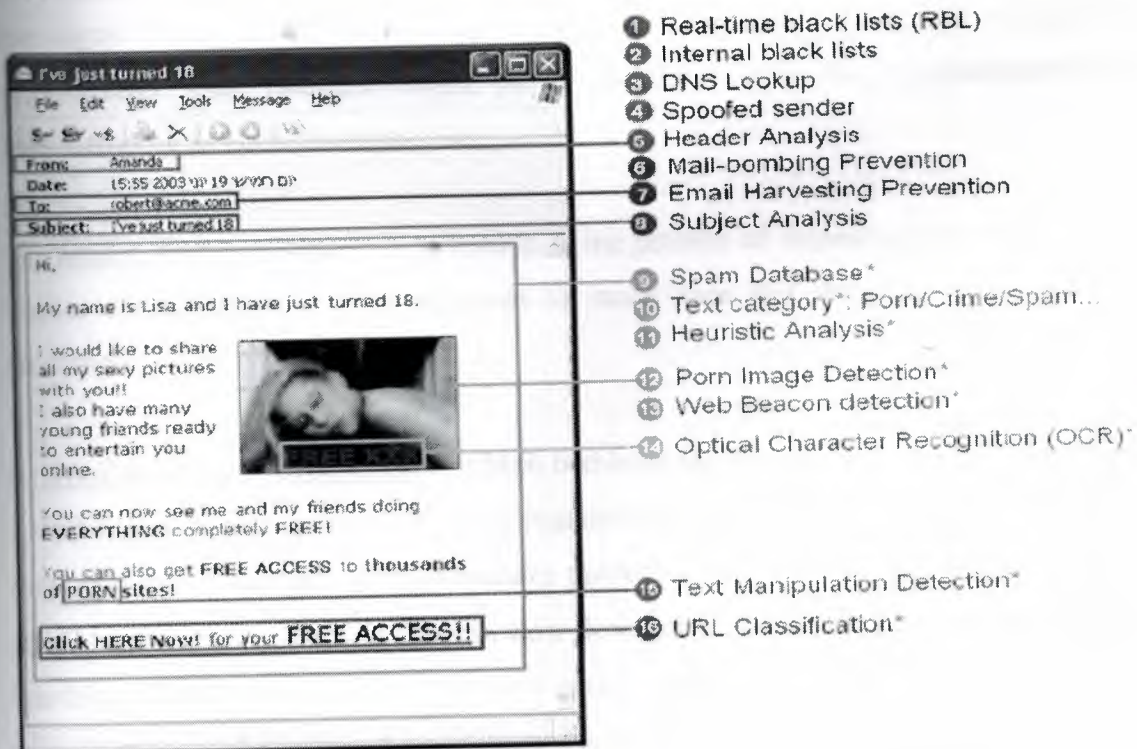


Figure 10.2. Methods used to identify and block Spam mail

1- Real-Time Black Lists: This technique, commonly referred to as RBL (real-time black-hole lists), checks the incoming IP address against various black lists to verify that the sending server is not listed as an open mail relay that spammers can use to relay their unsolicited emails.

2- Internal Black Lists and White Lists: The external RBL and other blacklists are sometimes not enough. In some situations there will be need for some specific and accurate lists. Blocking by sender email address is an old technique because today spammers use random email addresses with existing valid domain names each time they send a new bulk of spam. However, blocking email from certain domains known to be used by spammers can yield good results. White lists are the opposite of black lists and contain addresses that can be misidentified as spam sources such as some subscribed mailing lists and newsletters or emails arriving from certain domains that also host spammers or that are open relays.

3- DNS Lookup: This technique can identify if the sending mail server is a legitimate one and has a valid host name. This will eliminate the majority of spam sent by mail servers connected to the Internet using a dial-up connection.

4- Anti-Spoofing: Email address spoofing is a technique used to send email messages from outside sources masquerading as internal addresses within the organization. One example would be to send an email to john@company.com masquerading as alice@company.com.

5- Header Analysis: Header verification is the process of inspecting the email SMTP header for compliance with standards to make sure that they are not forged by spammers.

6- Mail-Bombing Prevention: Mail bombing can be executed as a DoS attack on a mail server, or as a spam attack on an organization, attempting to send masses of spam messages to the domain, using dictionary mailers. Anti-bombing regulates the flow of email to prevent an overload that can slow down email servers or cause DoS. The anti-bombing feature monitors the number of email sessions that are open at the same time as well as the total number of email messages in the spool.

7- Directory Harvesting Attacks Prevention: Directory Harvest Attack (DHA), is a technique that allows spammers to gather valid email addresses of targeted organizations. During a DHA, spammers attempt to deliver email to different addresses that are not rejected by the receiving mail server are determined as valid and are later used to compile and sell as part of spam mailing lists.

8- Subject Analysis: Many spam messages contain common text in the subject of the email. A sample list of such subjects that can be used to clearly identify spam would include text like:

- a) Get rich fast
- b) University diploma
- c) Save money
- d) Viagra online
- e) Credit repair
- f) Increase your...
- g) Make \$\$\$

9- Spam Database: The spam database technology extracts hash signatures from received email and compares against a database of known spam emails. This is a very powerful technology if implemented correctly because it has the potential of blocking spam in real time.

10- Text Category Analysis: Text category analysis is a very sophisticated technology that looks at the content of the email and tries to analyze if it contains text that can be interpreted as spam (offer of purchase something, offer to use services, invite to visit a web site, etc).

11- Heuristic Analysis: Heuristic analysis can identify email that looks like spam based on the existence of certain common characteristics, such as the usage of mixed foreign character sets, image links which are server queries, mix of different obscure or non-printable characters, different encoding methods, etc.

12- Porn Image Detection: A large portion of spam messages contain visual pornographic content. This content is not just time and resource consuming but can be

also offensive. Porn image detection analysis images, using pattern matching algorithms and block almost all pornographic visual content from emails.

13- Web-Beacons, Cookies&Scripts: Web beacons are a very powerful tool in the hands of sophisticated spammers that identify “live” email addresses and intensify sending of spam to these addresses. When an email with a web-beacon arrives at a user’s inbox, it is usually displayed in the preview panel on the screen. The spammer will immediately know that the email arrived at the inbox and that it was viewed by the user. This indication is enough to immediately start bombarding this user with more and more spam. Web beacons can also be embedded inside scripts, cookies and other more sophisticated HTML commands.

14- OCR Text Recognition: Many spam messages arrive as a graphic image and not as text. Many anti-spam systems cannot analyze the text that appears in the graphic image and thus unable to identify spam. Optical Character Recognition (OCR) technique knows to read the text even if it appears as a graphic image.

15- Text Manipulation Analysis: Many spam messages use tricks to make it harder for anti-spam tools to analyze its content. Text manipulation is a method or replacing certain characters in the spam text with visually similar characters, seperating characters so it is difficult to analyze as a whole word, using auditory similar symbols or letter to represent words and parts of words, and more. Some examples are:

- a) P0RN instead of PORN (zero instead of capital O).
- b) V.I.A.G.R.A instead of VIAGRA.
- c) 4U instead of “for you”.

16- URL Classification: Virtually any spam messages contain a URL link as this is the major for spammers. Beside visible links like “click here for more information” or an image link, many emails also contain dynamically downloaded content like images text and advertisements, only visible when the email is being viewed. Checking all these URL links against a database of known classified URLs, yields surprisingly accurate results. Moreover, this technology delivers an almost zero rate of false positives.

17- Anti-Relay: Anti-relay systems protect mail servers from being hijacked and used by spammers to broadcast unsolicited emails. The anti-relay option blocks all email to recipients that do not belong to the organization.

The final front in the war on spam is the elimination of email in the traditional sense. Many businesses are being forced to take this approach. Even the White house has been forced to follow this path. That may be what happens to all email in the long run. The amount of spam, and the inability to control that spam, may become so unmanageable that the traditional email system we know today collapses and gets replaced either with forms or with a set of advanced, secure servers that put spammers out of business.

10.4 ABOUT SPYWARE

Spyware programs are applications that send information about its user via the Internet to the creator of the spyware, or the publisher. Spyware usually consists of core functionality and functionality for information gathering. The core functionality appeals to users and entices them to install and use the spyware. Information that is sent to the publisher is normally used for improved direct marketing purposes. The type of sent information differs depending on the spyware program. Users often overlook the information gathering functionality of spyware, leaving them unaware that the spyware publisher is gathering data from their computers. Most spyware programs are free programs that are available on the Internet, and in some cases are useful tools. Some examples are:

- ▶ Download utilities
- ▶ Games
- ▶ Media players
- ▶ Accounting software

Technically, spyware can be considered as two separate pieces of software that are shipped in one package:

1. The core functionality that is visible and useful to the user.

2. Information-gathering functionality that gathers, maintains, monitors and sends user and/or computer information in the background.

The question arises as to why users would want to use spyware. Most users, if not all, are unaware of the information gathering functionality of spyware programs. Spyware is generally freeware, and the information gathering functionality is not mentioned before users install the software, making it attractive to users. The type of information that is sent varies per spyware program. Let's take a closer look at a spyware Internet Browser to see how the spyware program operates:

John just installed a new Internet browser to experience the "enhanced browsing and downloading experience" as the spyware publisher advertised. The registration process includes answering some questions about personal details such as name, age, gender, nationality, profession and level of education. After finishing the registration process, John decides to start browsing and downloading some software. The software downloads slightly faster than with his old browser, making him a happy user of the spyware product. During every browsing session, John is shown several advertisements. Some are interesting to him and others are not. Over time, John follows several advertisement hyperlinks. Each time the browser notifies the spyware publisher. The spyware publisher constructs a profile based on the gathered information so that John is only presented with advertisements that are likely to be in his field of interest. This example shows that users of spyware may not be aware of the information exchange that occurs in the background between the spyware program and the spyware publisher.

10.5 HOW SPYWARE OPERATES

Depending on the goal of the information gathering functionality of spyware, the nature of the gathered information varies among spyware programs. Some spyware programs only send the time of use and other statistical data. Other spyware programs that incorporate improved advertising correlate the gathered data. In order to keep the gathered information linked to a specific installation, all the information sent to the spyware publisher needs to be uniquely identified. The unique identifier must be stored on the user's computer. There are different methods for creating the unique identifier. The two most commonly used methods are generating a Globally Unique Identifier

(GUID) and storing a cookie on the hard disk during the installation of the spyware program. A GUID contains data that is unique to a computer's hardware. A cookie is a technical term for a file that contains data, which a specific program often uses. The data that a cookie contains depends on the program that creates it. In the case of spyware, a cookie could contain uniquely identifiable data such as the user name, computer specifications, and installation version. Every time a spyware program sends information to the spyware publisher, the unique identifier is sent as well, allowing the spyware publisher to update the customer database. Figure 1 shows how the spyware publisher and the users are connected, and the data that is stored on each side of the connection.

Like most commercial software packages, each spyware program includes a EULA that users agree to before they can use the software. The EULA includes all of the usual clauses, as well as information on the software gathering process for statistical purposes and improved software experience. Many spyware EULA's are worded in such a way that; they contain so much information that is difficult to extract meaningful data that deals with the information-gathering functionality, the meaningful data that deals with the information-gathering functionality is ambiguous. This is dangerous, because the majority of users accept the EULA without understanding the implications. Many users are upset when they discover that some of the software they use is spyware. They often consider it malicious software. The majority of spyware users remain unaware of the fact that they use spyware. The users in this group may be subject to customer profiling without their knowledge. Many requests have been made to spyware publishers to change the EULA's or the products so that users understand what information is sent, when it is sent, and the purpose for which it is used. This might prevent users from getting upset about spyware, and if implemented properly, it could give users some control over the types of gathered information as well. Another notification method is for software publishers to set-up web sites that describe the information-gathering functionality of their software, but the information that is presented on the sites is often ambiguous

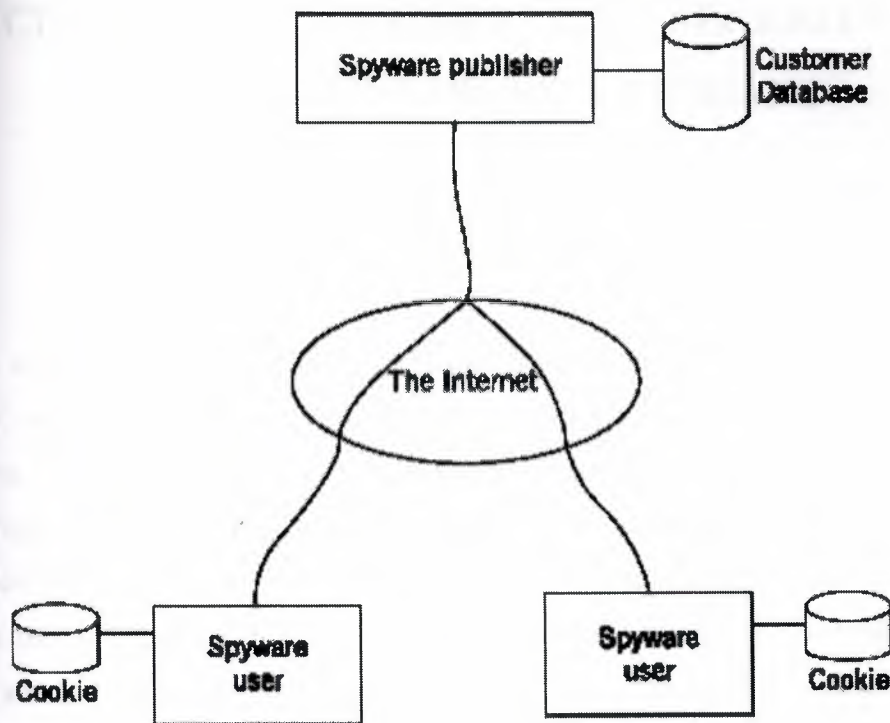


Figure 10.3. The way Spyware operates

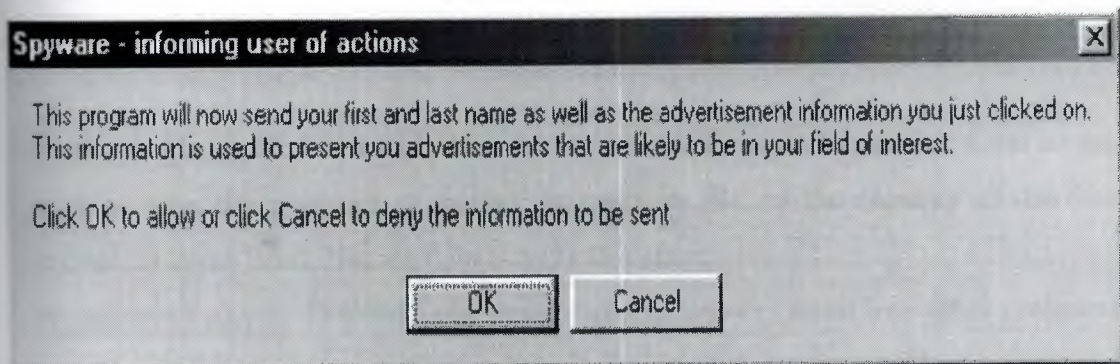


Figure 10.4. A dialog box notification example

Many users are unaware that they are using spyware because of the poor notification on the information-gathering functionality of their software. This subjects users to customer profiling without their knowledge. For this reason, it is important for users to read and understand the EULA and other notification methods before installing software. It is also of equal importance that software publishers provide users with clear and unambiguous notifications of the actions their software performs.

CHAPTER ONE: INFORMATION SECURITY

PART ELEVEN: VIRUSES&TROJANS

11.1 ALL ABOUT VIRUSES

A computer virus is a small program that designed to replicate and spread, generally without the user's knowledge. Computer viruses spread by attaching themselves to other programs (e.g. word processor, MS outlook) or to the boot sector of a disk. When an infected file is activated, executed or when the computer is started from an infected disk, the virus itself is also executed. A virus can do anything that other programs do. The only difference is that it attaches itself to another program and executes secretly when the host program is run. Once a virus program is executing, it can perform any function, such as erasing files and programs. Some type of viruses include; Boot viruses, Windows viruses, VBScript viruses and Macro viruses (viruses inside MS office). During its lifetime, a typical virus goes through the following four stages:

- Dormant phase: The virus is idle. It will eventually be activated by some event, such as date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage.
- Propagation phase: The virus places an identical copy of itself into other programs or into certain system areas on the disk. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.
- Triggering phase: The virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.
- Execution phase: The function is performed. The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files.

Most viruses carry out their work in a manner that is specific to a particular operating system and, in some cases, specific to a particular hardware platform. Thus, they are designed to take advantage of the details and weaknesses of particular systems.

Table 11.1. Cost impact of Worms and Viruses

Year	Virus/Worm	Estimated Damage
1999	Melissa virus	\$80 million
2000	Love bug virus	\$10 billion
2001	Code Red I and II worms	\$2.6 billion
2001	Nimda virus	\$590 million to \$2 billion
2002	Klez worm	\$9 billion
2003	Slammer worm	\$1 billion

Table 11.2. Number of industry incidents reported

Year	1988	1989	1990	1991	1992	1993	1994	1995
Incidents	6	132	252	406	773	1,334	2,340	2,412
Year	1996	1997	1998	1999	2000	2001	2002	2003
Incidents	2,573	2,134	3,734	9,859	21,756	52,658	82,094	114,855

11.2 SIMPLE VIRUSES

A simple virus that merely replicate itself is the easiest to detect. If a user launches an infected program, the virus gains control of the computer and attaches a copy of itself to another program file. After it spreads, the virus transfers control back to the host program, which functions normally. Yet no matter how many times a simple virus infects a new file or floppy-disk, the infection always makes an exact copy of itself. Anti-Virus software need only search or scan sequence of bytes, known as signature, found in the virus.

11.3 ENCRYPTED VIRUSES

In response, virus authors began encrypting viruses. The idea was to hide the fixed signature by scrambling the virus, making it unrecognizable to a virus scanner. An encrypted virus consists of a virus decryption routine and an encrypted virus body. If a user launches an infected program, the virus decryption routine first gains control of the computer, then decrypts the virus body. Next, the decryption routine transfers control of the computer to the decrypted virus. An encrypted virus infects programs and files as any simple virus does. Each time it infects a new program, the virus makes a copy of both the decrypted virus body and its related decryption routine, encrypts the copy, and attaches both to a target. To encrypt the copy of the virus body, an encrypted virus uses an encryption key that the virus is programmed to change from infection to infection. As this key changes, the scrambling of the virus body changes, making the virus appear different from infection to infection. This makes it extremely difficult for Anti-Virus software to search for a virus signature extracted from a consistent virus body. However the decryption routines remain constant from generation to generation, a weakness that Anti-Virus software quickly evolved to exploit. Instead of scanning just for virus signatures, virus scanners were modified to also search for the sequence of bytes that identifies specific search routine.

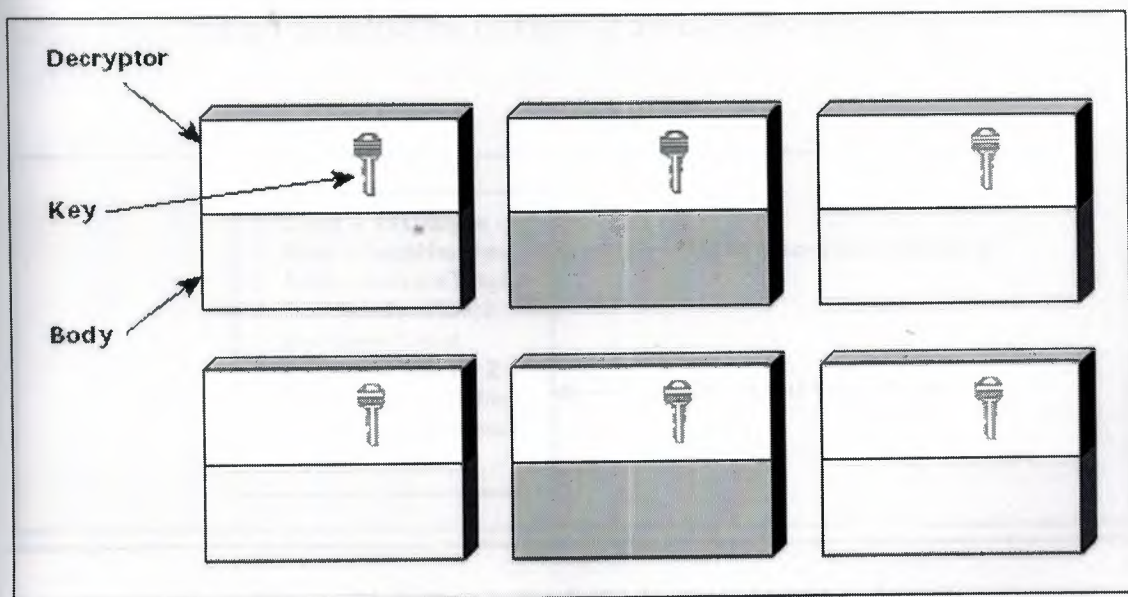


Figure 11.1. An encrypting virus always propagates using the same decryption routine. However, the key value within the decryption routine changes from infection to infection. Consequently, the encrypted body of the virus also varies, depending on the key value.

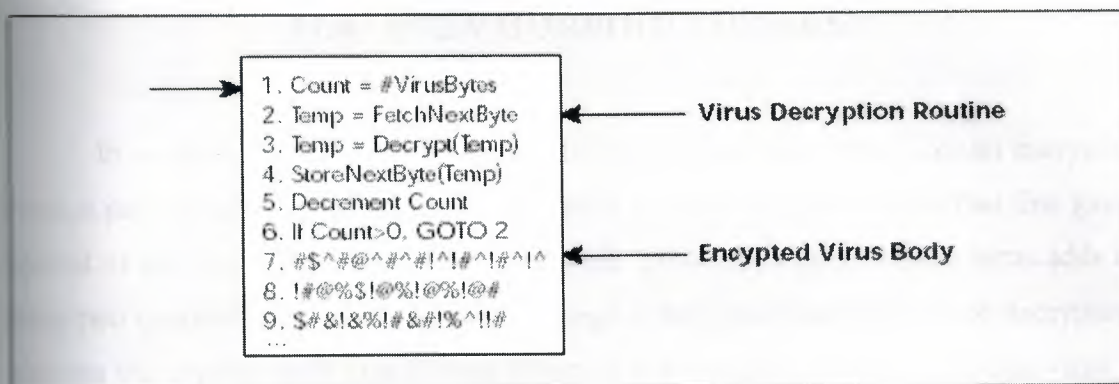


Figure 11.2. This is what an encrypted virus looks like before execution

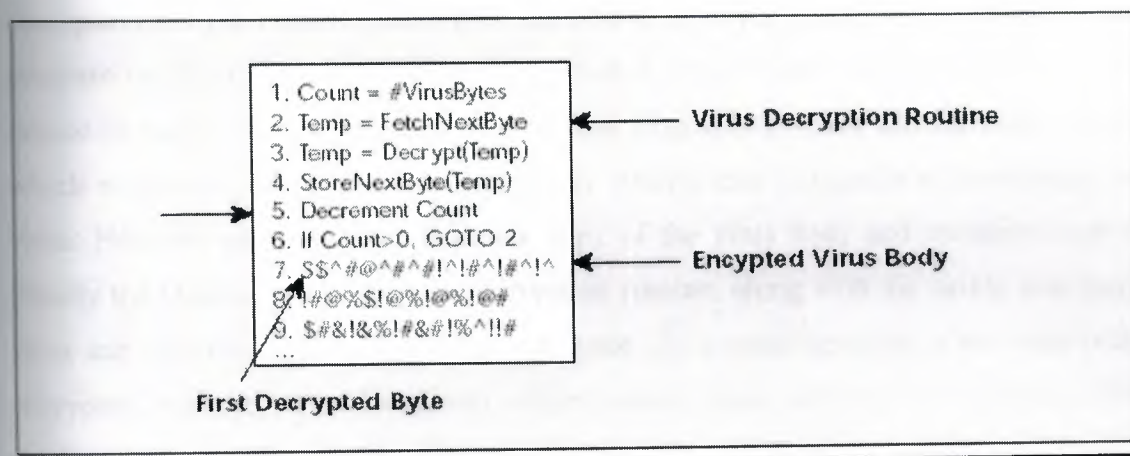


Figure 11.3. At this point, the virus has executed its first five instructions and has decrypted the first byte of the encrypted virus body.

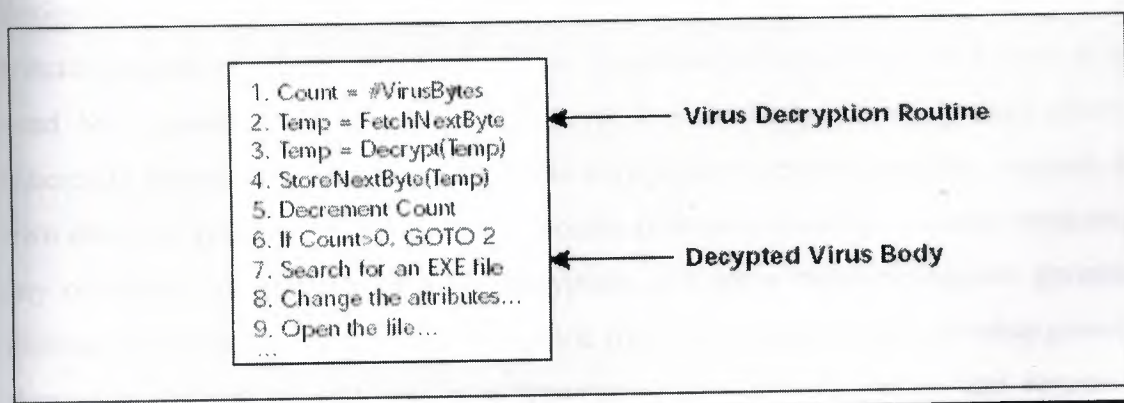


Figure 11.4. This is the fully decrypted virus code

11.4 POLYMORPHIC VIRUSES

In retaliation, virus authors developed the polymorphic virus. Like an encrypted virus, a polymorphic virus includes a virus body and a decryption routine that first gains control of the computer, then decrypts the virus. However a polymorphic virus adds to *these two components a third; A mutation engine that generates randomized decryption routines* that change each time a virus infects a new program. In a polymorphic virus, a *mutation engine* and the virus body are both encrypted. When a user runs a program infected with a polymorphic virus, the decryption routine first gains control of the computer, then decrypts both the virus body and the mutation engine. Next the decryption routine transfers control of the computer to the virus, which locates a new program to infect. At this point, the virus makes copy of both itself and the mutation engine in random access memory (RAM). The virus then invokes the mutation engine, which randomly generates a new decryption routine that is capable of decrypting the virus. Next the virus encrypts this new copy of the virus body and mutation engine. Finally the virus appends this new decryption routine, along with the newly encrypted virus and mutation engine, onto a new program. As a result not only is the virus body encrypted, but the virus decryption routine varies from infection to infection. This confounds a virus scanner searching for the sequence of bytes that identifies a specific decryption routine. With no fixed signature to scan for, and no fixed decryption routine, no two infections look alike.

Anti-Virus researchers first fought back by creating special detection routines designed to catch each polymorphic virus, one by one. By hand line by line, they wrote special programs designed to detect various sequences of computer code known to be used by a given mutation engine to decrypt a virus body. This approach proved inherently impractical, time-consuming and costly. Each new polymorphic requires its own detection program. Also a mutation engine produces seemingly random programs, any of which can properly perform decryption, and some mutation engines generate billions of variations. These shortcomings led Anti-Virus researchers to develop *generic decryption* techniques that trick a polymorphic virus into decrypting and revealing itself. Generic decryption assumes:

- The body of polymorphic virus is encrypted to avoid detection.

- A polymorphic virus must decrypt before it can execute normally.
- Once an infected program begins to execute, a polymorphic virus immediately take control of the computer to decrypt the virus body, then yield control of the computer to the decrypted virus.

A scanner that uses generic decryption relies on this behavior to detect polymorphics. Each time it scans a new program file, it loads this file into a self-contained virtual computer created from RAM. Inside this virtual computer, program files execute as if running on a real computer. The scanner monitors and controls the program file as it executes inside the virtual computer. A polymorphic virus running inside the virtual computer can do no damage because it is isolated from the real computer. When a scanner loads a file infected by a polymorphic virus into this virtual computer, the virus decryption routine executes and decrypts the encrypted virus body. This exposes the virus body to the scanner, which can then search for signatures in the virus body that precisely identify the virus strain. If the scanner loads a file that is not infected, there is no virus to expose and monitor. In response to non-virus behavior, the scanner quickly stops running the file inside the virtual computer, removes the file from the virtual computer, and proceeds to scan the next file. The process is like injecting a mouse with a serum that may or may not contain a virus, and then observing the mouse for adverse effects. If the mouse becomes ill, researchers observe the visible symptoms, match them to known symptoms, and identify the virus. If the mouse remains healthy, researchers select another vial of serum and repeat the process.

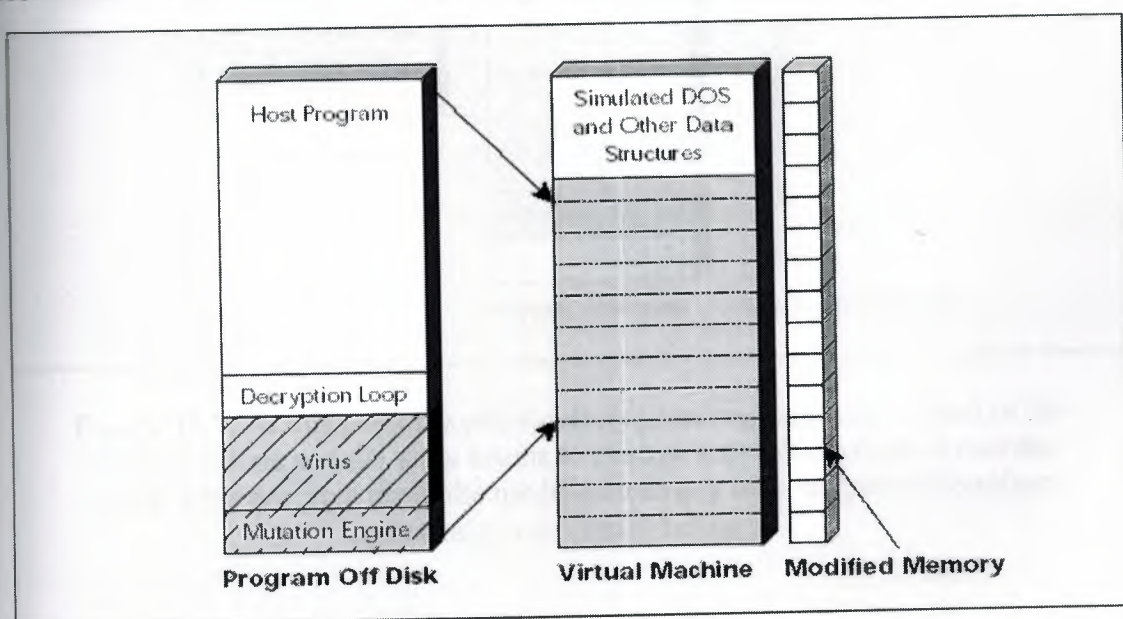


Figure 11.5. The generic decryption engine is about to scan a new infected program

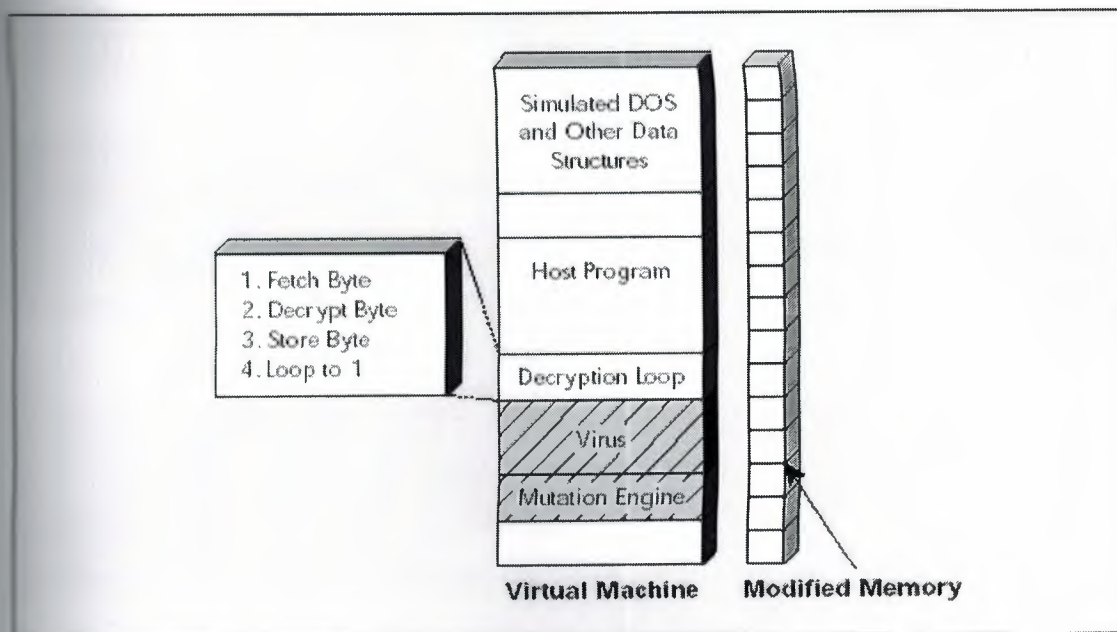


Figure 11.6. The generic decryptor loads the next program to scan into the virtual machine. Notice that each section of memory in the virtual machine has a corresponding modified memory cell depicted on the right side of the virtual machine. The generic decryption engine uses this to represent areas of memory that are modified during the decryption process.

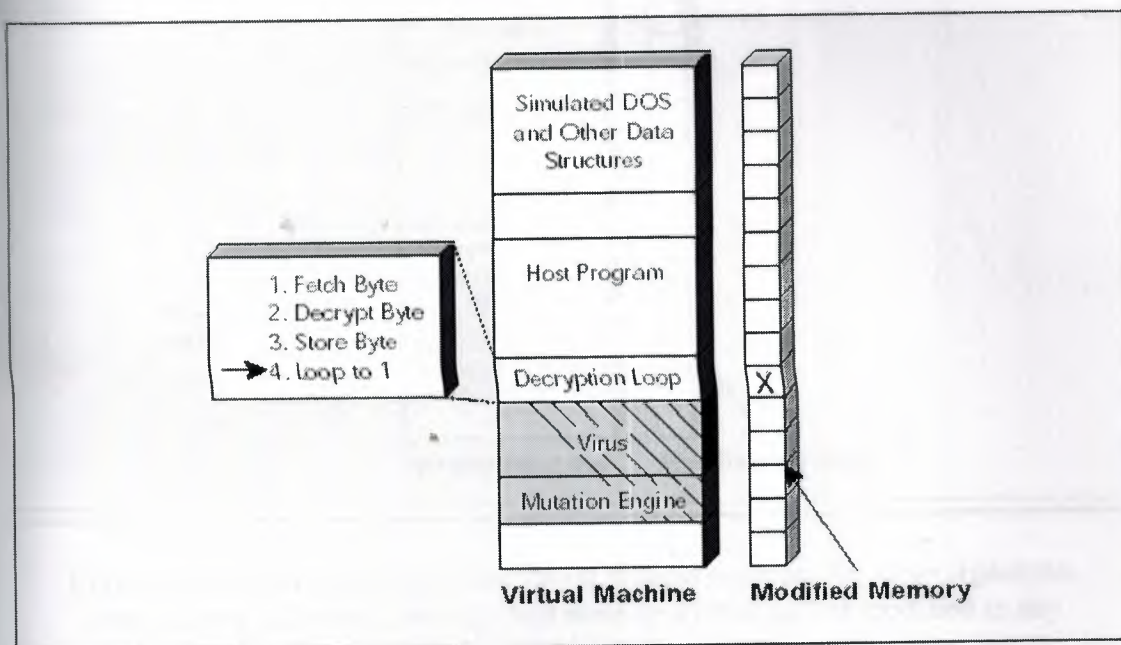


Figure 11.7. At this point the generic decryption engine passes control of the virtual machine and the virus begins to execute a simple decryption routine. As the virus decrypts itself, the modified memory table is updated to reflect the changes to virtual memory.

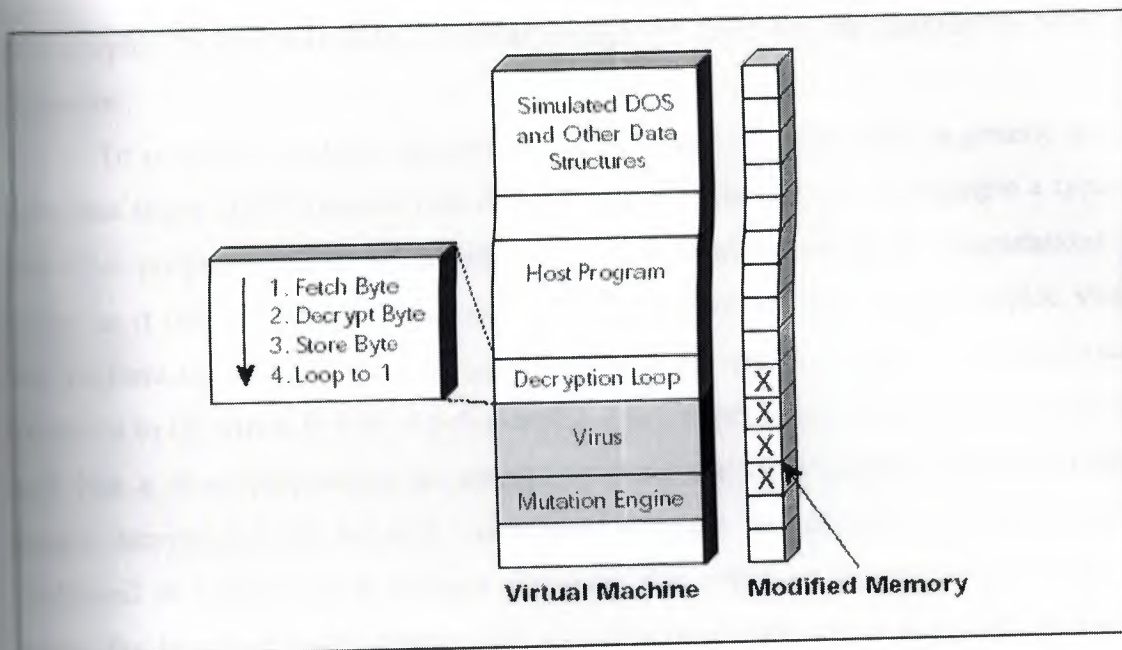


Figure 11.8. Once the virus has decrypted enough of itself, the generic decryption engine advances to the next stage.

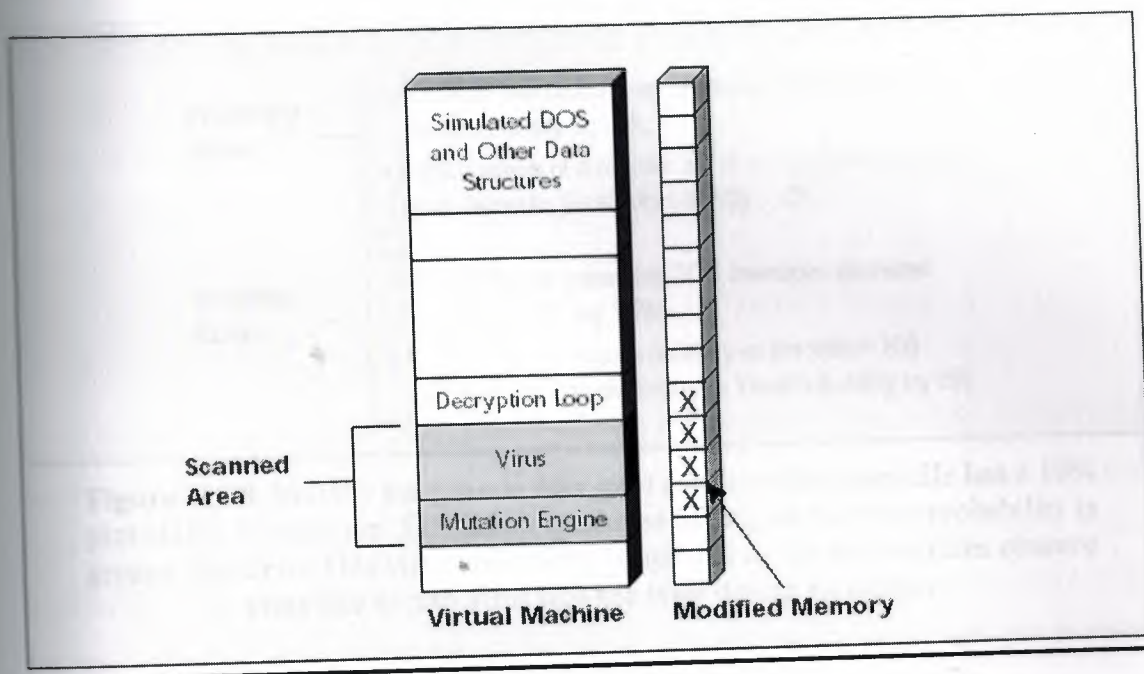


Figure 11.9. Now the generic decryption scanner searches for virus signatures in those areas of virtual memory that were decrypted and/or modified in any way by the virus. This is the most likely location for virus signatures.

The key problem with generic decryption is speed. Generic decryption is of no practical use if it spends five hours waiting for a polymorphic virus to decrypt inside the virtual computer. Similarly if generic decryption simply stops short, it may miss a

polymorphic before it is able to reveal enough of itself for the scanner to detect a signature.

To solve this problem, generic decryption employs “*heuristic*”, a generic set of rules that helps to differentiate non-virus from virus behavior. As an example a typical non-virus program will in all likelihood use the results from math computations it makes as it runs inside the virtual computer. On the other hand, a polymorphic virus may perform similar computations, yet throw away the results, because those results are irrelevant to the virus. In fact, a polymorphic may perform such computations solely to look like a clean program in an attempt to elude the virus scanner. Heuristic-based generic decryption looks for such inconsistent behavior. An inconsistency increases the likelihood of infection and prompts a scanner that relies on heuristic-based rules to extend the length of time a suspect file executes inside the virtual computer, giving a potentially infected file enough time to decrypt itself and expose a lurking virus.

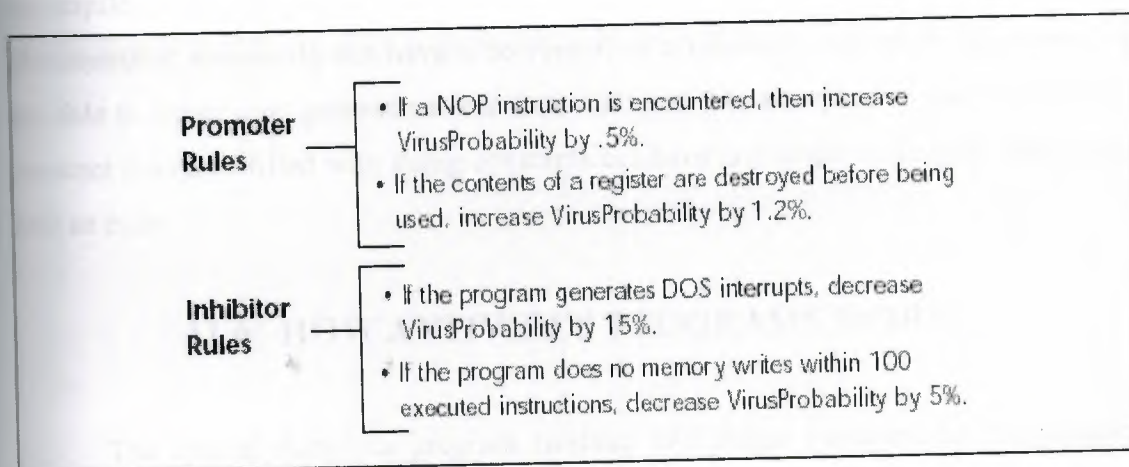


Figure 11.10. Initially the generic decryptor assumes that every file has a 10% probability of infection. Emulation continues as long as the virus probability is greater than zero. This virus probability is updated as the various rules observe virus like or non-virus like behavior during emulation.

Unfortunately, heuristic demand continual research and updating. Heuristic rules tuned to detect 500 viruses, for example, may miss 10 of those viruses when altered to detect 5 new viruses. Also as virus writers continue trying to make viruses look like clean programs, heuristic can easily balloon to the point where almost any program might share attributes that trigger the scanner to lengthen the time it takes to examine a file. In addition generic decryption must rely on a team of Anti-Virus researchers able to analyze millions of potential virus variations, extract a signature, then modify a set of

heuristic while also guarding against the implications of changing any heuristic rules. This requires extensive, exhaustive regression testing. Without this commitment, heuristic quickly becomes obsolete, inaccurate and inefficient.

11.5 METAMORPHIC VIRUSES

Virus writers still need to waste weeks or months to create a new polymorphic virus that often does not have a chance to appear in the wild because of its bugs. On the other hand, a researcher might be able to deal with the detection of such a virus in a few minutes or few days. Obviously virus writers try to implement various new code evolution techniques in order to make the researcher's job more difficult. Metamorphic virus carries its source and drops it whenever it can find a compiler installed on the machine. The virus inserts junk code into and removes it from its source, and then recompiles itself. This way a new generation of the virus will look completely different. Metamorphic viruses do not have a decryptor, or a constant virus body, however they are able to create new generations that look different. Metamorphic viruses do not use a constant data area filled with string constants but have one single code body that carries data as code.

11.6 HOW ANTIVIRUS PROGRAMS WORK?

The typical Antivirus program contains two major components; the scanning application and the scanning engine. The scanning application provides a user interface, alert functions, and logging mechanisms. The application determines which files to scan and how to react when a virus is found. However, it knows absolutely nothing about computer viruses. Every time it scans a file or a floppy disk, it calls upon the scanning engine to detect computer viruses in the designated location. If a scanning engine locates a virus, it reports back to the scanning application. The scanning application then informs the user of the infection and prompts the user to repair the file. If the user chooses to do so, the scanning application again calls upon the scanning engine to repair the infected file or disk. The scanning engine contains dozens of complex searching algorithms along with CPU emulators and elaborate program logic. In contrast to the scanning application, the engine knows nothing about user interfaces, which files to

scan, or what to tell the user when it finds a virus. It only knows how to detect and repair viruses. It simply examines the file or disk the scanning application directs to it, and determines whether there are any viruses present. Typically, scanning engines work by scanning each file or disk for thousands of virus fingerprints. These fingerprints are stored in the virus definition data files that users around the world download each week when they obtain their virus software updates.

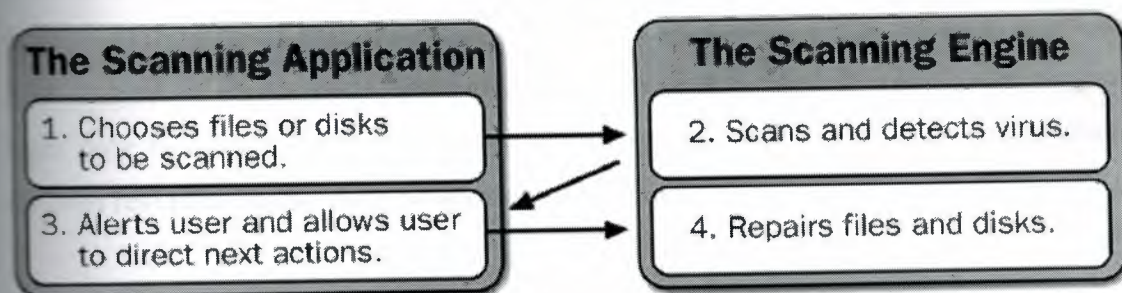


Figure 11.11. How the scanning engine and scanning software work together.

The inflexibility of traditional Antivirus architecture, with its combined scanning application and scanning engine, becomes apparent when an IT organization is faced with a complex new virus. The obstacles to successful enterprise-wide Antivirus protection are numerous, and virus eradication is time-consuming and expensive. Some of the consequences in applying typical Antivirus programs to special case viruses are:

- ❖ A new version of the Antivirus product must be released to eradicate a special case virus. When a virus is discovered that cannot be handled by a single fingerprint update, the entire Antivirus program must be updated and reinstalled. Even if no changes are required to the scanning application itself, changes to the scanning engine require new deployment of the entire product.
- ❖ The Antivirus software on each supported platform must be updated to include the new scanning engine logic. And each of these new product inlines must be deployed by the IT organization.
- ❖ Producing new code for each of the Antivirus platform is time intensive and costly, and forces the typical Antivirus vendor to develop and deploy their full spectrum of product support over a period of months.

“Abnormal” viruses have become the norm. Unless detection and repair for complex viruses can be implemented with the same ease as the fingerprinting technology of the past, creating and maintaining a robust Antivirus solution becomes impossible. Therefore, powerful new detection and repair strategies must become part of the normal virus definition update. The result is new technology, which separates the scanning engine from the scanning application. The scanning engine can now be updated on its own, improved on its own, and redistributed as part of the standard virus definitions through all available update methods. Each update of the engine is generated from one set-of source code. That means engineers only need to modify the program logic once in order to properly update the scanning engines for all Antivirus products, for both real time and on demand scanning functions.

→ New virus eradication technology:

- ❖ An employee’s workstation crashes over and over. She calls the IT help desk.
- ❖ Several hours later, an IT representative arrives. He notices that the computer infected by a virus. He sends the infected files to the Antivirus research center.
- ❖ Researchers analyze the files and realize that they are dealing with a new virus that’s too different for the existing Antivirus program to detect or repair.
- ❖ Engineers update the Antivirus engine to deal with the new virus.
- ❖ Updated version of the Antivirus engine delivered to the user and also to the enterprise through online automatic update or as a file on their server’s.

With complex new viruses becoming the norm rather than the exception, it is more important than ever to employ an Antivirus product that has a modular engine. This architecture can save countless hours of testing, updating, manual virus elimination, and calls to the help desk.

11.7 WORMS

Worm is a virus like program that seeks out other connected hosts in a computer network and, by exploiting a vulnerability, transfers itself to them. A worm uses Internet connection to spread from system to system. Once active within a system, a

worm can behave as a computer virus or Trojan, or perform any number of destructive actions. To replicate itself a worm usually uses electronic mail to spread. Here are some of the killer worms:

Melissa Worm: Melissa was among the most dangerous worm to cause a serious impact with email systems accross the public Internet. Melissa worm spread to the first 50 addresses of the user's Outlook address book. Damage from the Melissa worm was estimated at \$1.1 billion.

ILove You: In May 2000, the I Love You worm appeared on the Internet, first in Asia, then spreading quickly accross the globe. Instead of sending a copy of the worm to the first 50 or 100 addresses in the user's Outlook address book, I love You used every single address in the user's address book, spreading faster and further than any previous email worm. Damage estimates from the I Love You worm went as high as \$8.75 billion.

Code Red: In July 2001 Code Red appeared, taking advantage of a flaw in Microsoft's IIS web server to spread across the Internet. Once it found a vulnerable host, Code Red would copy "command.com" and rename it to "root.exe" in the web server's publically accessible scripts directory providing complete command line control to anyone who knew the web server had been compromised. Code Red would then use the compromised web server to explore the Internet and identify other vulnerable IIS server. After 25 days after installation, Code Red was designed to launch a Denial of Service (DoS) attack against the White House's IP address. Code Red damage estimates were \$2.6 billion.

Nimda: The Nimda worm appeared in September 2001. When received as an email attachment, the user did not have to open the attachment to execute the program. It would automatically execute when simply viewed in Outlook's preview panel. Nimda contained its own email program for sending email, so it did not have to depend on the host having an operating mail server to further propagate the Nimda worm. If the compromised user had an operational email program, Nimda would still use its own mail server so as not to leave evidence in the user's mail program that it was emailing copies of Nimda to addresses in the user's address book. Nimda was clear evidence that

worms were evolving and becoming smarter, but damage estimates of \$645 million were lower than for Code Red.

Klez: The Klez worm appeared in October 2001. Like many of its predecessors, Klez took advantage of a bug in Outlook that allowed it to be installed simply by viewing the email in the preview panel. While propagating only by email, Klez incorporated a rather unique approach, selecting one email address from the user's address book to use as the "from" address, then sending the worm to all the other addresses. In this manner, the email often appeared to have been sent from a co-worker or someone the address actually knew. While it is difficult to determine the total dollar amount of damages attributable to Klez and its variants, estimates range as high as \$9 million.

SQL Slammer: The SQL Slammer worm struck January 25, 2003, and entire sections of the Internet began to go down almost immediately. SQL Slammer took advantage of a known vulnerability in Microsoft SQL server software, a limit to the actual number of servers compromised. Using the non-familiar random address scanning technique to search for vulnerable hosts, SQL Slammer included elements that enabled it to propagate rapidly. Damage estimates for SQL Slammer were \$1.2 billion.

Future worms will take advantage of new fast scanning routines that will dramatically accelerate the initial propagation phase and even use pre-scanning data to virtually eliminate the first slow phase of scanning for vulnerable hosts.

11.8 WHAT IS A TROJAN?

An apparently innocent program designed to circumvent the security features of a system. A trojan horse includes command procedure containing hidden code that, when invoked performs some unwanted or harmful function. Trojan horses could be used to gain access to the files of another user and data destruction. A trojan horse could be either:

- Unauthorized instructions contained within a legitimate program. These instructions perform functions unknown to the user.

- A legitimate program that has been altered by the placement of unauthorized instructions within it. These instructions again perform functions unknown to the user.

The name "trojan horse" came from in the 12. century B.C. Greece declared war on the city of Troy. The dispute erupted when the prince of Troy abducted the Queen of Sparta and declared that he wanted to make her his wife, which made the Greeks and especially the Queen of Sparta quite furious. The Greeks gave chase and engaged Troy in a 10-year war, but unfortunately for them, all of their efforts went down to drain. Troy was too well fortified. In a last effort the Greek army pretended to be retreating, leaving behind a huge wooden horse. The people of Troy saw the horse and thinking it was some kind of a present from the Greeks, pulled the horse into their city, without knowing that the finest soldiers of Greece were sitting inside it, since the horse was hollow. Under the cover of night, the soldiers sneak out and opened the gates of the city, and later, together with the rest of the army, killed the entire army of Troy. This is why such a program is called a *Trojan horse*. It pretends to do something while it does something completely different, or does what it is supposed to be, and hides its malicious actions from the user's prying eyes.

11.9 HOW DO TROJANS WORK?

Most trojans come into two parts, a client and a server, but there are exceptions where the trojan does not need a client, as it is able to automatically do what it was intended to do (stealing passwords etc.), without any intervention from the attacker. However those who use both client and server in order to operate, need assistance from the attacker. Once the victim runs the server (unknowingly), the attacker will use a port to connect to the server, and start using your computer. TCP/IP is the usual protocol used. When the server (trojan) is executed on the victim's machine, it will hide itself somewhere within the computer and start listening on the specified by the attacker port. However there are trojans that automatically listen for incoming connections once run, which will wait a period of time to reduce the risk of being detected.

It's necessary for the attacker to know the victim's IP address to connect to his/her machine. Many trojans have features such as the ability to mail the victim's IP,

as well as the ability to message the attacker via ICQ or IRC. Most trojans use auto-starting methods in order to auto-run each time your computer is started. These methods include, but are not limited to, using the Windows registry, using some of the windows system files, as well as using third party configuration files.

Windows trojans vary in their functions and abilities, although here is a brief *summary of the most common ones:*

- *Change the victim's screen resolution.*
- Notify attacker when victim goes online.
- Process monitoring. The attacker has the ability to monitor all of your processes, start new ones, as well as the ability to kill current one.
- Registry editor. It gives the attacker, the ability to view/create/delete/change everything in the registry.
- Find files feature. Provides the attacker with the opportunity to find any file on the hard-drive, if he/she is looking for something particular.
- Disconnect victim. The attacker can hang-up the victim's connection to the net anytime.
- The attacker can make screenshots of your activities.
- Open the browser at an address specified by the hacker.
- Enable/disable keyboard.
- Restart Windows.
- Open/close the CD-ROM tray.
- Turn monitor on/off.
- Retrieve passwords on victim's computer.

11.10 MOST COMMON TROJANS

Remote Administration Trojans: These Trojans are the most popular trojans now. Let you have access to someone else hard-drive, and also perform many functions on his computer. Modern remote administration trojans are very simple to use. If you get someone's IP address, you have full control over his/her computer. You can also bind trojans into other programs which appear to be legitimate. Remote administration trojans have the common remote access trojan functions like; Keylogging, which means

logging the target's keyboard functions and sometimes even interfering with them, thus being able to use your keyboard to type instead of the target. Upload and download functions, make a screenshot of the targets monitor and so on. Trojans would usually want to automatically start whenever you boot-up your computer.

Remote administration trojans open a port on your computer and bind themselves to it. Make the server file listen to incoming connections and data going through these ports. Then, once someone runs his client program and enters the victim's IP, the trojan starts receiving commands from the attacker and runs them on the victim's computer.

Password Trojans: Password trojans steal passwords from our computers and then send them to the attacker or the author of the trojan. Whether it's Internet password, Hotmail password or ICQ password, there is a trojan for every password. These trojans usually send the information back to the attacker via e-mail.

Priviledges-Elevating Trojans: These trojans would usually be used to fool system administrators. They can either be binded into a common system utility or pretend to be something unharmful and even quite useful and appealing. Once the administrator runs it, the trojan will give the attacker more priviledges on the system. These trojans can also be sent to less-priviledge users and give the attacker access to their account.

Keyloggers: These trojans are very simple. They log all of your keystrokes (including passwords), and then either save them on a file or email them to the attacker once in a while. Keyloggers usually don't take much disk space and can masquerade as important utilities, thus making them hard to detect. Some keyloggers can also highlight passwords found in text boxes with titles such as "enter password" or just the word password somewhere within the title text.

Destructive trojans: These little fellows do nothing but damaging your computer. These trojans can destroy your entire hard-drive, encrypt or just scramble important files and basically make you feel very unpleasent. Some might seem like joke programs, they can either pretend to be formatting your hard-drive, sending all of your passwords to some evil cracker, self-destructing your computer etc.

Denial of service (DoS) attack trojans: These trojans are becoming very popular these days. The main idea is that if attacker start attacking to the victim simultaneously, this will generate a lot of traffic (more than victim's bandwidth) and its access to the Internet will be shut down. Another version of a DoS trojan is the mail-bomb trojan, whose main aim is to infect as many machines as possible and simultaneously attack specific e-mail address with random subjects and contents which cannot be filtered.

Proxy/Wingate trojans: An interesting feature implemented in many trojans is the ability to turn the victim's computer into proxy/wingate server available to the whole world or only to the attacker. It's used for anonymous Telnet, ICQ, IRC etc. and also to registered domains with stolen credit cards and many other illegal activities. This gives the attacker complete anonymity and the chance to do everything from victim's computer, and if he/she gets caught the trace leads back to you.

FTP trojans: These trojans are probably the simplest ones and are kind of outdated as the only thing they do is open port 21 (the port for FTP transfer) and let everyone connect to your machine or only the attacker.

Software detection killers: They are programs that will kill popular protection programs such as; ZoneAlarm, Norton Anti-virus and many other that protect your machine. When they are disabled, the attacker will have full access to your machine, enabling the attacker to perform some illegal activities.

11.11 IN WHAT WAYS COULD I BE INFECTED?

1. Via ICQ/IRC: Suppose you are talking with someone, probably a girl (but actually an attacker), you ask for a picture of her, she sends it to you but in reality she (attacker) changed the name and extension of the trojan to something else and when you run it you become infected with that trojan. So before you run something, even its with a JPG icon, check its extension and make sure it really is an image file.
2. Via Attachments: When users receive an email containing an attachment saying that they will get free porn, free internet access etc., they run it without completely

understanding the risk to their machines. Many people have gotten themselves infected by the famous "Microsoft Internet Explorer Update" sent directly to their mailboxes, but these updates are definitely trojans. Because Microsoft never sends updates via email. Even if you receive an email attachment from your friend be sure that the file is safe and clean before opening it, because attackers can use any name to send trojans to you.

3. Browser and Email Software Bugs: Users do not update their software versions as often they should be, and a lot of the attackers are taking advantage of this well known fact. Latest version of program reduces the risk to a minimum.
4. Fake Programs: Imagine a freeware program that's very suitable for your needs. These programs can be very dangerous and as a very useful and easy way for attackers to infect your machine with trojans. Some of them may open a port on your computer for attacker to access your information or simply it can perform the job of a trojan.

You must realize that there isn't a 100% sure way of protecting against windows trojan and virus infections. Although you can significantly reduce the risk by using Anti-Virus programs, trojan cleaner programs and keep them up-to-date for new kind of trojans/viruses. Also you should be careful with email attachments, scan your computer regularly to keep it clean and do not trust anyone while using ICQ/IRC kind chat programs. When executing files, first check their types. Is it a really a .doc or is its some executable with a .doc icon? Make sure you always have the latest version of the software you are using as new bugs appear very often and programs are regularly updated. Consider freeware programs as very risky software to download. Don't download any software you have never heard about or at least scan it after you download it.

CHAPTER ONE: INFORMATION SECURITY

PART TWELVE: BIOMETRIC SECURITY

12.1 WHAT IS A BIOMETRIC SECURITY?

As organizations search for more secure authentication methods for user access, e-commerce and other security applications, biometrics is gaining increasing attention. Biometric security systems are basically systems that identify an individual by using a physical, natural feature, such as fingerprint, or by the person's voice. These systems are meant to replace passwords in many cases, which can be stolen, forgotten, guessed or cracked. The information security field uses three different types of authentication:

- 1) Something you know – A password, PIN, or piece of personal information.
- 2) Something you have – A card key, smart card, or token (like a secure ID card).
- 3) Something you are – A biometric.

Of these, a biometric is the most secure and convenient authentication tool. It can be borrowed, stolen, or forgotten, and forging one is practically impossible. Biometrics measure individuals unique physical or behavioral characteristics to recognize or authenticate their identity. Common physical biometrics include; fingerprints, hand or palm geometry, and retina/iris/facial characteristics. Behavioral characters include signature, voice, keystroke pattern and gait. Of this class of biometrics, technologies for signature and voice are the most developed. Process involved in using a biometric system for security is as follows:

1. Capture the chosen biometric (such as fingerprint or retina).
2. Process the biometric and extract and enroll the biometric template.
3. Store the template in a local repository, a central repository, or a portable token such as a smartcard.
4. Live-scan the chosen biometric.
5. Process the biometric and extract the biometric template.
6. Match the scanned biometric against stored templates.

7. Provide a matching score to business applications.
8. Record a secure audit trail with respect to system use.

12.2 FINGERPRINTS

These basically consist of a device where you would lay your finger on and it reads your fingerprint to match with a template. The computer takes sections of the persons fingerprints. A fingerprint looks at the patterns found on a fingertip. There are a variety of approaches to fingerprint verification. Some emulate the traditional police method of matching, others use straight pattern-matching devices, and still others are a bit more unique, including things like ultrasonics. Some verification approaches can detect when a live finger is presented, some cannot. A greater variety of fingerprint devices is available than for any other biometrics. As the prices of these devices and processing costs fall, using fingerprints for user verification is gaining acceptance. Fingerprint verification may be a good choice for in-house systems, where you can give users adequate explanation and training, and where the system operates in a controlled environment. It is not surprising that the workstation access application area seems to be based almost exclusively on fingerprints, due to the relatively low cost, small size, and ease of integration of fingerprint authentication devices.

12.3 HAND GEOMETRY

Hand geometry involves analyzing and measuring the shape of the hand. These biometric systems scan the measurements of a users hand by using recording light from the fingertips to the webbing of the hand. It measures each finger within 1/10,000 of an inch, marking where the beginning and end of the finger is by the varying intensities of light. The information obtained is stored digitally in a system as a template or possibly even coded on a magnetic stripped ID card. This biometric offers a good balance of performance characteristic and is relatively easy to use. It might be suitable where there are more users or where users access the system infrequently and are perhaps less disciplined in their approach to the system. Accuracy can be very high if desired, and flexible performance tuning and configuration can accommodate a wide range of applications. Organizations are using hand geometry readers in various scenarios,

including time and attendance recording, where they have proved extremely popular. Ease of integration into other systems and processes, coupled with ease of use, makes hand geometry an obvious first step for many biometric projects.

12.4 RETINA AND IRIS SCANNERS

A retina based biometric involves analyzing the layer of blood vessels situated at the back of the eye. An established technology, this technique involves using a low-intensity light source through an optical coupler to scan the unique patterns of the retina. Retinal scanning can be quite accurate but does require the user to look into a receptacle and focus on a given point. This is not particularly convenient if you wear glasses or are concerned about having close contact with the reading device. For these reasons, retinal scanning is not warmly accepted by all users, even though the technology itself can work well.

An iris based biometric, on the other hand, involves analyzing features found in the colored ring of tissue that surrounds the pupil. Iris scanning, undoubtedly the less intrusive of the eye-related biometrics, uses a fairly conventional camera element and requires no close contact between the user and the reader. In addition, it has the potential for higher than average template-matching performance. Iris biometrics work with glasses in place and is one of the few devices that can work well in identification mode. Ease of use and system integration have not traditionally been strong points with iris scanning devices, but you can expect improvements in these areas as new products emerge.

12.5 FACE RECOGNITION

Face recognition analyzes facial characteristics. It requires a digital camera to develop a facial image of the user for authentication. This technique has attracted considerable interest, although many people don't completely understand its capabilities. Some vendors have made extravagant claims, which are very difficult, if not impossible to substantiate in practice for facial recognition devices. Because facial scanning needs an extra peripheral not customarily included with basic PC's, it is more of a nice market for network authentication. However, the casino industry has

capitalized on this technology to create a facial database of scam artists for quick detection by security personnel.

12.6 SIGNATURE SCANNING

Signature verification analyzes the way a user signs her name. Signing features such as speed, velocity, and pressure are as important as the finished signature's static shape. Signature verification enjoys a synergy with existing processes that other biometrics do not. People are used the signatures as a means of transaction-related identity verification, and most would see nothing unusual in extending this to encompass biometrics. Signature verification devices are reasonably accurate in operation and obviously lend themselves to applications where a signature is an accepted identifier. Surprisingly, relatively few significant signature applications have emerged compared with other biometric methodologies.

12.7 VOICE RECOGNITION

Voice authentication is not based on voice recognition but on voice-to-print authentication, where complex technology transforms voice into text. To enroll into the system a user is instructed to repeat a certain phrase several times. The computer would take the samples to digitalize and store each sample and then from the acquired data, build a voice signature that would be open to allow some voice variations of the produced signature. Voice biometrics has the most potential for growth, because it requires no new hardware, most PC's already contain a microphone. However, poor quality and ambient noise affect verification. In addition, the enrollment procedure has often been more complicated than with other biometrics, leading to the perception that voice verification is not user friendly. Therefore, voice authentication needs improvement. One day, voice may become an additive technology to finger-scan technology. Because many people see finger scanning as a higher authentication form, voice biometrics will most likely be relegated to replacing or enhancing PIN's, passwords, or account names.

12.8 USES FOR BIOMETRICS

Security systems use biometrics for two basic purposes: to verify or to identify users. Identification tends to be the more difficult of the two uses because a system must search database of enrolled users to find a match. The biometric that a security system employs depends in part on what the system is protecting and what it is trying to protect against.

Physical Access: For decades, many highly secure environments have used biometric technology for entry access. Today the primary application of biometrics is in physical security: to control access to secure locations (rooms or buildings). Biometric devices, typically hand geometry readers, are in office buildings, hospitals, casinos, etc. Biometrics are useful for high-volume access control. For example, biometrics control access of 65,000 people during the 1996 olympic games, and Disney World uses a fingerprint scanner to verify season-pass holders entering the theme park. Another example is an eye-ticketing, which Charlotte/Douglas International Airport in North Carolina evaluating. Eye-ticket links a passenger's frequent fly number to an iris scan. After the passenger enrolls in the system, the machine performs ticketing and check-in.

Virtual Access: For a long time, biometric-based network and computer access were areas often discussed but rarely implemented. Analysts see virtual access as the application that will provide the critical mass to move biometrics for network and computer access from the realm of science-fiction devices to regular system components. Physical lock downs, can protect hardware, and passwords are currently the most popular way to protect data on a network. Biometrics, however, can increase a company's ability to protect its data by implementing a more secure key than a password. Using biometrics also allows a hierarchical structure of data protection, making the data even more secure. Passwords supply a minimal level of access to network data, but biometrics on the other hand can supply high level of security if implemented correctly.

E-commerce Applications: E-commerce developers are exploring the use of biometrics and smart-cards to more accurately verify a trading party's identity. For

example, many banks are interested in this combination to better authenticate customers and ensure nonrepudiation of online banking, trading, and purchasing transactions. Engineers are working on the cardholder verification method, which would enlist biometrics to replace signature verification. Some companies are using biometrics to obtain secure services over the telephone through voice authentication.

Covert Surveillance: One of the more challenging research areas involves using biometrics for covert surveillance. Using facial and body recognition technologies, researchers hope to use biometrics to automatically identify known suspects entering buildings or traversing crowded security areas such as airports. The use of biometrics for covert identification as opposed to authentication must overcome technical challenges such as simultaneously identifying multiple subjects in a crowd and working with uncooperative subjects.

Biometric security is one area that no segment of the IT industry can afford to ignore. Biometrics provide security benefits accross the spectrum, from IT vendors to end users, and from a security system developers to security system users. So anyone who wants to use biometric security should think well and decide which biometric security standard he/she is going to use. Table below compares different kind of biometric security standards.

Table 12.1. Biometric security standards comparison

Characteristic	Fingerprints	Hand Geometry	Retina	Iris	Face	Signature	Voice
Ease of use	High	High	Low	Medium	Medium	High	High
Error incidence	Dryness, dirt, age	Hand injury, age	Glasses	Poor lighting	Lighting, age, glasses, hair	Changing signatures	Noise, colds, weather
Accuracy	High	High	Very High	Very High	High	High	High
Cost	-	-	-	-	-	-	-
User Acceptance	Medium	Medium	Medium	Medium	Medium	Medium	High

Required security level	High	Medium	High	Very High	Medium	Medium	Medium
Long-Term stability	High	Medium	High	High	Medium	Medium	Medium

Biometric security has been around for decades but has mainly been for highly secretive environments with extreme security measures. The technologies behind biometrics are still emerging. An increasing number of agencies and departments are turning to biometrics to achieve a higher level of security. Biometric traits, unlike passwords cannot be lost, stolen or easily duplicated.

CHAPTER TWO: WIRELESS NETWORKS

PART ONE: BLUETOOTH

1.1 INTRODUCTION

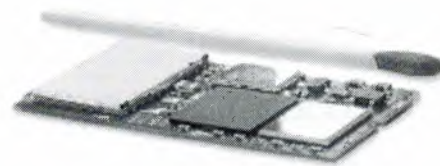
Bluetooth is a short range radio technology that enables wireless connectivity between mobile devices. The three main goals for Bluetooth are: small size, minimal power consumption, and low price. The Bluetooth specification intended to 'unite' separate personal computing devices such as laptops, PDA's, cell phones, and peripherals, like printers for example. In short, Bluetooth is wireless technology intended for short-range radio links to replace cables. It's primary features are voice and data capabilities, low complexity, low power and low cost.

Why is it called Bluetooth? Harald Bluetooth was king of Denmark in the late 900s. He managed to unite Denmark and part of Norway into a single kingdom, then introduced Christianity into Denmark. He left a large monument, the Jelling rune stone, in memory of his parents. He was killed in 986 during a battle with his son. Choosing this name for the standard indicates how important companies from the Baltic region are to the communications industry, even if it says little about the way the technology works.

Ericsson of Sweeden invented Bluetooth in 1994. It operates in the 2.4 GHz radio frequency band, offers 712 kb data rates and has a range of approximately 10 meters. Application of Bluetooth technology has also been extended to offer wireless access to LAN's, the mobile phone network and the Internet. Since it operates in the 2.4 GHz band, the Bluetooth standard is targeted for worldwide approvals so that, anywhere in the world, any Bluetooth enabled device can connect to other bluetooth devices in its proximity, regardless of manufacturer. According to the specification, Bluetooth devices communicate wirelessly in short-range, ad-hoc networks called *piconets*. Each device can simultaneously communicate with up to seven other devices in the piconet. Also, each device can be a participant in several piconets. These piconets are established automatically as devices enter and leave the radio network. The broad foundation of the Bluetooth specification accounts for the wide interest in potential

applications. These range from straightforward cable replacement to sophisticated networking applications. Examples:

- ▶ Wireless headsets for cell phones for hands-free, wire-free phone calls.
- ▶ PC mouse/keyboard using Bluetooth wireless connection to the PC.
- ▶ Wireless printing between a PC or laptop and a Bluetooth enabled printer.
- ▶ Networking and file-sharing between PC's, PDA's and laptops in a meeting.
- ▶ Internet access for Bluetooth enabled devices via the nearest Bluetooth enabled device on the Internet.
- ▶ Synchronize contact information between a cell phone, PDA, notebook and desktop wirelessly.



Bluetooth Module

Figure 1.1 Examples of Bluetooth usage

The most obvious benefit from Bluetooth is the original goal of simple cable replacement between two devices. For many situations, this alone is compelling based on the physical elimination of inconvenient cables that take space, create clutter and limit device placement. In industrial and commercial applications, the presence of wires creates potential safety and task interference issues. The wide range of device types and standard interface afforded by Bluetooth allows selection of devices optimized each for their particular function and ergonomics. The multi-point capabilities of Bluetooth communications allow one interface to support communications with multiple devices: printers, scanners, PDA's, other PC's, etc. Bluetooth wireless networking, in general, provides a simple and fast path to ad-hoc networks with minimal equipment and overhead.

1.2 HOW BLUETOOTH WORKS?

The Bluetooth system consists of a radio unit, a link control unit, and a support unit for link management and host terminal interface functions. The *Host Controller Interface (HCI)* provides the means for a host device to access Bluetooth hardware capabilities. For example, a laptop computer could be the host device and a PC card inserted in the PC is the Bluetooth device. All commands from the host to the Bluetooth module and events from the module to the host go through the HCI interface. The protocol stack is above the radio and baseband hardware, partly residing in the Bluetooth unit and in the host device.

The Bluetooth network is called a *piconet*. A piconet is defined as a group of devices consisting of at least one master and one slave unit which all share the same frequency hopping sequence. A piconet is formed by a master and up to seven active slaves. The slaves in the piconet only communicate with the master. In the simplest case it means that two devices are connected. The device that initiates the connection is called a *master* and the other devices are called *slaves* (figure 3). The majority of Bluetooth applications will be point-to-point applications. Bluetooth connections are typically ad-hoc connections, which means that the network will be established just for the current task and then dismantled after the data transfer has been completed. A master can have simultaneous connections (point-to-multipoint) to up to seven slaves. However, the data rate is limited. One device can also be connected in two or more piconets. The setup is called *scatternet*. A scatternet can be formed by linking two or more piconets. When a device is present in more than one piconet, it must time-share and synchronize to the master of the piconet with which it is currently communicating. A device can, however, only be a master to one piconet at a time. The master/slave roles are not necessarily fixed and can also be changed during the connection if, for example, the master does not have enough resources to manage the piconet. Master/slave switch is also needed in the scatternet. Most of current Bluetooth implementations support piconets only. Point-to-multipoint support depends on the implementation.

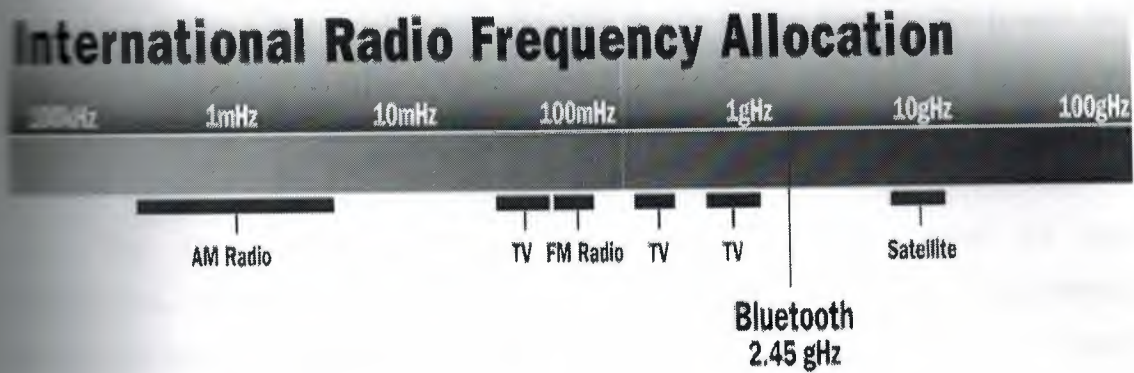


Figure 1.2. Bluetooth offers 2.45 GHz bandwidth

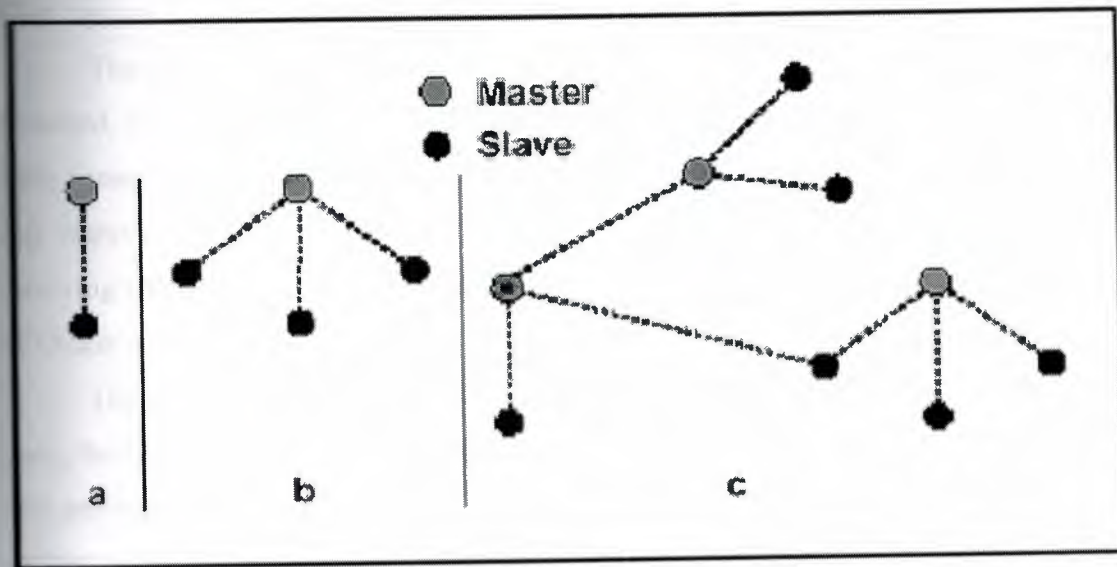


Figure 1.3. Bluetooth piconet and scatternet scenarios:

- a) Point to point connection between two devices.
- b) Point to multipoint connection between a master and three slaves.
- c) Scatternet that consists of three piconets.

Bluetooth technology uses a frequency hopping technique, which means that every packet is transmitted on a different frequency. The master sets the hopping sequence, and the slaves synchronize to the master. In most countries, 79 channels can be used. With a fast hop rate (1600 hops per second), good interference protection is achieved. Another benefit is a short packet length. If some other device is jamming the transmission of a packet, the packet is resent in another frequency determined by the frequency scheme of the master. Subsequent time slots are used for transmitting and receiving. A packet nominally covers a single slot, but can be extended to cover three or five slots. In multi slot packets, the frequency remains the same until the entire packet is

sent. When using a multi-slot packet, the data rate is higher, because the header and long switching time after the packet are needed only once in each packet. On the other hand, the robustness is reduced, in a crowded environment the long packets will more probably be lost.

The *Asynchronous Connectionless Links (ACL)* are defined for data transmission, primarily packet data. They support symmetrical and asymmetrical packet-switched connections. Multi-slot packets use the ACL link type and can reach the maximum data rate of 723 kbps in one direction and 57.6 kbps in the other direction. The master controls the ACL link bandwidth and decides how much of the bandwidth a slave can use in a piconet. Broadcast messages are supported in the ACL link.

The *Synchronous Connection Oriented Links (SCO)* support symmetrical, circuit switched, point-to-point connections and are therefore primarily used for voice traffic. Two consecutive time slots at fixed intervals are reserved for an SCO link. The SCO link reserves every sixth slot for a transmitting channel and the subsequent slot for a receiving channel, so there can be up to three simultaneous SCO links. The data rate for SCO link is 64 kbps.

Data is transmitted in packets. Each packet consists of three entities; the access code, the header and the payload. The size of the access code and the header are fixed. The payload may range from 0 to 2745 bits per packet. The control packets may also consist of the access code only, or of the access code and header only. In ACL packets all three entities are needed.

Three methods are used for ensuring reliable data transfer in crowded environments. In the *Forward Error Correction (FEC)* scheme, additional check bits are added in the packet header or the payload. In the *Automatic Repeat Request (ARQ)* scheme, the data payload is retransmitted until the recipient sends an acknowledgment. Acknowledgment information is included in the header of the return packet. To determine whether the payload is correct or not, a *Cyclic Redundancy Check (CRC)* code is added to the packet.

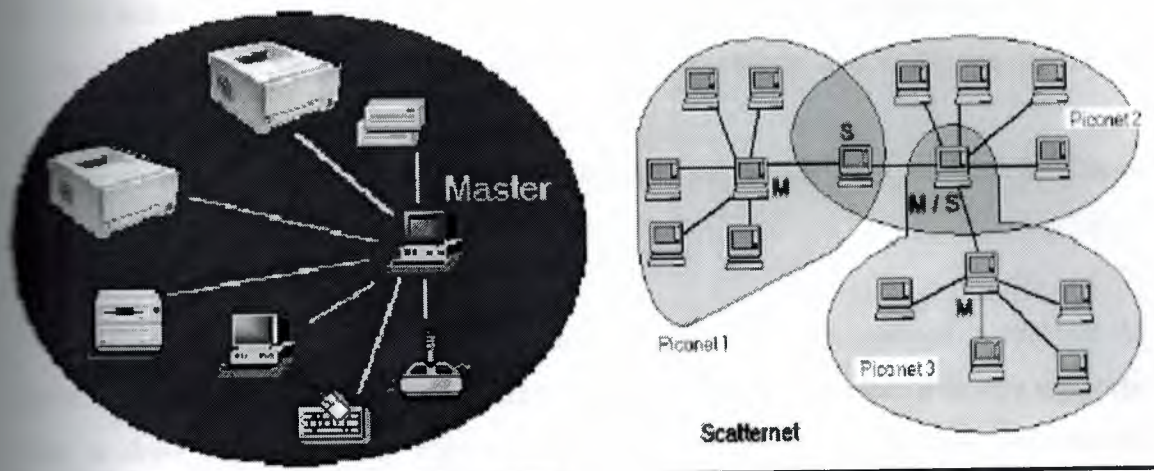


Figure 1.4. Master and Slave Connections in Bluetooth

1.3 CONNECTION ESTABLISHMENT AND BLUETOOTH PROFILES

As applications need to connect to one another, it is probably appropriate to introduce how devices are connect to each other. Unlike the wired technology it is designed to replace, a Bluetooth device does not have to be aware of the devices and capabilities they are attaching to. There is a built in mechanism to inquire for devices, connect to them and once connected, discover the services they possess in their database. In its simplest form the devices needing to connect proceed as follows:

- 1) The master enters inquiry mode and sends out an inquiry to discover devices available to connect to.
- 2) Potential slaves make themselves discoverable by entering inquiry scan mode and listen for an inquiry from a master.
- 3) On receiving an inquiry, the slave responds to the master with a Frequency Hop Synchronization packet (FHS). The FHS contains information that is needed to create a connection to the device. This information includes its Bluetooth address and class of device.

- 4) The master collects the FHS information from each device discovered. To connect to one of these devices, the master goes into page mode and will page the device using corresponding Bluetooth address.
- 5) The slave being paged by a master will need to be in page scan mode to be able to connect to a master.

Once a connection is created between two devices, the application can use the *Service Discovery Protocol (SDP)* to find out what particular services a device supports. This is done via a L2CAP channel to the service discovery server. The potential Bluetooth services available sit on top of the Bluetooth protocol stack.

Profiles specify how the interoperable solution for the functions described in the usage models is provided; in another words, a profile defines the protocols and protocol features supporting a particular usage model. Bluetooth version 1.1 profiles are shown in the figure below. Some profiles are dependent on other profiles. For example; file transfer profile, object push profile, and synchronization profile are dependent on the generic object exchange profile. All profiles are dependent on the generic access profile.

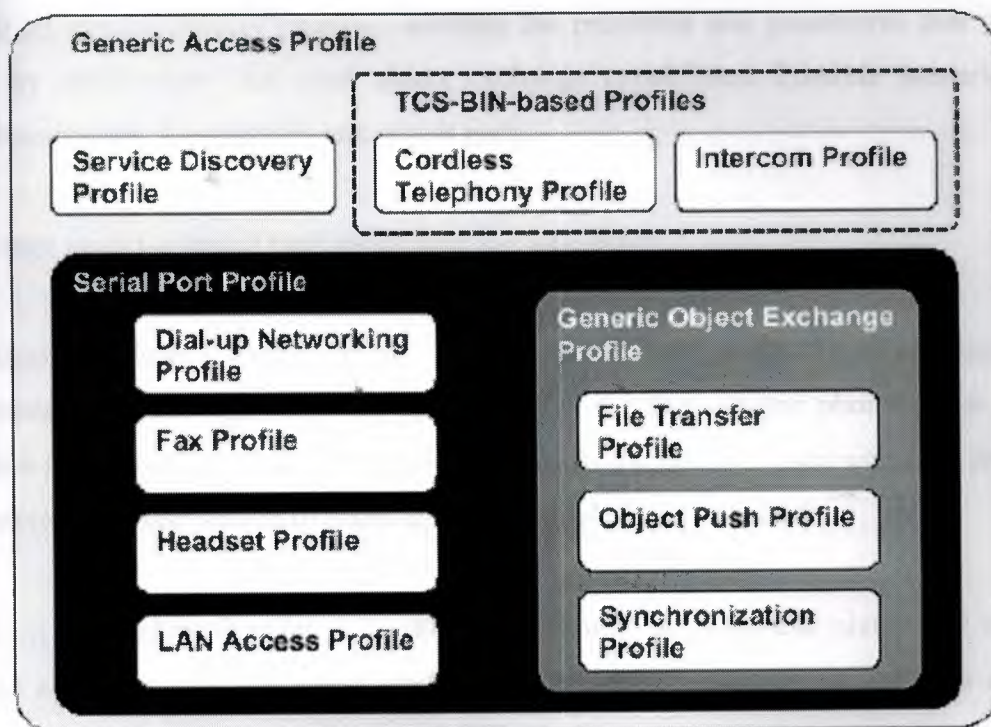


Figure 1.5. Bluetooth Profiles

Bluetooth products support different sets of profiles. In order to support certain profile, features of the profile must be implemented. The four general profiles in the Bluetooth are:

Generic Access Profile: Defines the generic procedures related to discovery of Bluetooth devices and link management aspects of connecting to Bluetooth devices (connecting mode procedures). It also defines procedures related to use of different security levels. In addition this profile includes common format requirements for parameters accessible on the user interface level. Every Bluetooth device has to support the generic access profile.

Service Discovery Application Profile: Defines the features and procedures for an application in a Bluetooth device to discover services of another Bluetooth device.

Serial Port Profile: Defines the requirements for Bluetooth devices necessary for setting-up emulated serial cable connections between two peer devices.

Generic Object Exchange Profile: Defines the protocols and procedures that will be used by applications that need object exchange capabilities. Possible scenarios are synchronization, file transfer, and object push.

→ Usage model-oriented profiles include the following:

Cordless Telephony Profile: Define the features and procedures required for interoperability between different units active in the three-in-one phone usage model (walkie-talkie, cellular phone). The cordless telephony profile is used when the phone is connected to a base station of fixed telephony network via Bluetooth.

Dial-Up Networking Profile: Describes how to use a cellular phone or a modem beside a computer as a wireless modem to receive data calls, to connect to a dial-up Internet access server, or to use other dial-up services.

Fax Profile: Defines how a computer can use a Bluetooth cellular phone or modem as a wireless fax modem to send or receive a fax message.

Headset Profile: Defines the requirements for Bluetooth devices necessary to support the headset use case. Wireless headsets can be used with cellular phones and laptops.

LAN Access Profile: Defines how Bluetooth enabled devices can access the services of a local area network using PPP (point-to-point protocol) over RFCOMM (Bluetooth protocol that emulates RS-232 signal) and how the same PPP mechanisms are used to form a network consisting of two Bluetooth enabled devices.

File Transfer Profile: Covers the scenarios that enable the user to browse and edit objects (files and folders) in the file system of another Bluetooth device and to transfer objects between two Bluetooth devices. The most common devices are, PC's, notebooks, and PDA's.

Object Push Profile: Covers the scenarios that enable users to push, pull, and exchange simple objects such as business cards between two Bluetooth devices such as, notebook, PC's, PDA's, and mobile phones.

Synchronization Profile: Covers the following scenarios: PIM data exchange between two devices and automatic synchronization of data (e.g. calendar items) when a device enters the proximity of the computer. Synchronization can be used between notebooks, PDA's, and mobile phones.

Table 1.1 Commonly used and implemented Bluetooth profiles

Profile	User Application	Description
Generic Access Profile	Foundation for all other profiles.	Provides access and security functions to other profiles.

Service Discovery Application Profile	Used to discover other devices and the services on those devices.	Allows a Bluetooth requesting device to discover the services offered by a mother device.
Serial Port Profile	Used to synchronize a PDA with a portable computer. Also used for other serial applications.	Provides serial cable emulation for Bluetooth devices.
Dial-up Networking Profile	Allows a mobile phone or dial-up modem to be used to access a network.	Connects a computer device to WAN through a dial-up networking gateway device such as a mobile phone or dial-up modem.
Object Push Profile	Used to exchange business cards, appointments, or to "push" objects.	Uses the generic Object Exchange Profile to transfer specific objects between devices. Also used to send objects to a Bluetooth printer.
Generic Object Exchange Profile	Performs tasks such as printing or file and business card transfers.	Used with an application layer to provide a method for file transfer, object push, or synchronization between two devices.

Headset Profile	Allows Bluetooth audio headsets to link wirelessly to telephony devices.	Uses a Synchronous Connection Oriented (SCO) channel for full-duplex telephony audio and an Asynchronous Connectionless Link (ACL) for control signaling.
PAN Profile	Used to connect Bluetooth devices in a personal area network.	While the new PAN profile is not yet widely available, it is a fundamental profile being developed by the Bluetooth Special Interest Group (SIG) that will be supported in future Microsoft operating systems.
File Transfer Profile	Used to transfer files between Bluetooth devices.	Allows files or folders to be browsed between two devices. Files can also be transferred using the Object Push or PAN profiles.
Human Interface Device (HID) Profile	Provides support for devices such as a Bluetooth mouse, keyboard, joystick, or gamepad.	Uses HID protocol from the USB specification. Provides quality of service for low-latency performance. On the PC this is a new Bluetooth profile.

<p>Hard Copy Cable Replacement (HCRP) profile</p>	<p>Allows two Bluetooth devices to establish the equivalent of a wired connection (for example, a Windows driver-to-printer connection).</p>	<p>Provides application layer connectivity. The application and the device must handle the protocol.</p>
--	--	--

1.4 BLUETOOTH PROTOCOLS

Protocols are needed to implement different profiles and usage models. Every profile uses at least part of the protocol stack. In order to achieve interoperability between two Bluetooth devices, they both must have the same vertical profile of the protocol stack. Bluetooth products support different sets of protocols. In order to support a certain Bluetooth profile, the mandatory features of certain protocols must be implemented.

Baseband and Link Control Protocol: This protocol controls the Bluetooth unit's synchronisation and transmission frequency hopping sequence. This layer is responsible for synchronizing the transmission-hopping frequency and clocks of different Bluetooth devices.

Audio Protocol: Audio transmissions can be performed between one or more Bluetooth units, using many different usage models. Audio is routed directly to and from baseband. Any two Bluetooth devices supporting audio can send and receive audio data between each other just by opening an audio link.

Link Manager Protocol (LMP): The Link Manager Protocol (LMP) is responsible for link set-up between Bluetooth units. It handles the control and negotiation of packet sizes used when transmitting data. The link manager protocol also handles management of power modes, power consumption, and state of a Bluetooth unit in a piconet. Finally this layer handles generation, exchange and control of link and encryption keys for authentication and encryption.

Logical Link Control and Adaptation Protocol (L2CAP): The Bluetooth Logical Link and Adaptation Protocol (L2CAP) is situated over the baseband layer and beside the Link Manager Protocol in the Bluetooth protocol stack. The L2CAP layer provides connection-oriented and connectionless data services to upper layers. The four main tasks for L2CAP are:

- ❖ Multiplexing – L2CAP must support protocol multiplexing since a number of protocols can operate over L2CAP.
- ❖ Segmentation and Reassembly – Data packets exceeding the maximum transmission unit (MTU), must be segmented before being transmitted. This and reverse functionality, reassemble, is performed by L2CAP.
- ❖ Quality of Service – The establishment of an L2CAP connection allows the exchange of information regarding current quality of service for the connection between the two Bluetooth units.
- ❖ Groups – The L2CAP specification supports a group abstraction that permits implementations for mapping groups onto a piconet.

Service Discovery Protocol (SDP): The Service Discovery Protocol (SDP), defines how a Bluetooth client's application shall act to discover available Bluetooth servers services and their characteristics. The protocol defines how a client can search for a service based on specific attributes without the client knowing anything of the available services. The SDP provides means for the discovery of new services becoming available when the client enters an area where a Bluetooth server is operating. The SDP also provides functionality for detecting when a service is no longer available.

Cable Replacement Protocol (RFCOMM): The RFCOMM protocol is a serial port emulation protocol. The protocol covers applications that make use of the serial ports of the unit. RFCOMM emulates RS-232 control and data signals over the Bluetooth baseband. It provides transport capabilities for upper level services (e.g. OBEX that use serial line as the transport mechanism).

Telephony Control Protocol: The telephony control protocol, is a bit-oriented protocol which defines the call control signalling for the establishment of speech and data calls between Bluetooth units. The protocol defines the signalling for establishment

and release of calls between Bluetooth units. Furthermore, it provides functionality to exchange signalling information unrelated to ongoing calls. Establishment of a voice or data call in a point-to-point configuration as well as in a point-to-multipoint configuration is covered in this protocol.

Host Controller Interface (HCI): The Host Controller Interface, HCI, provides a uniform interface method for accessing the Bluetooth Hardware capabilities. It contains a command interface to the baseband controller and link manager and access to Hardware status. Finally, it contains control and event registers.

Point-To-Point Protocol (PPP): The Point-To-Point Protocol in the Bluetooth technology is designed to run over RFCOMM to accomplish point-to-point connections. PPP is a packet oriented protocol and must therefore use its serial mechanisms to convert the packet data stream into a serial data stream.

TCP/UDP/IP: The TCP/UDP/IP standards are defined to operate in Bluetooth units allowing them to communicate with other units connected, for instance, to the Internet. Hence, the Bluetooth unit can act as a bridge to the Internet.

OBEX Protocol: OBEX is an optional application layer protocol designed to enable units supporting infrared communication to exchange a wide variety of data and commands in a resource sensitive standardized fashion. OBEX uses a client-server model and is independent of the transport mechanisms and transport API. The OBEX protocol also defines a folder-listing object, which is used to browse the contents of folders on remote device. RFCOMM is used as the main transport layer for OBEX.

Wireless Application Protocol (WAP): The Wireless Application Protocol (WAP) is a wireless protocol specification that works accross a variety of wide-area wireless network technologies bringing the Internet to mobile devices. Bluetooth can be used like other wireless networks with regard to WAP, it can be used to provide a bearer for transporting data between the WAP client and its adjacent web server. Furthermore Bluetooth's ad-hoc networking capability gives a wap client unique possibilities regarding mobility compared with other WAP bearers. The traditional form of WAP communications involves a client device that communicates with a server/proxy device

using the WAP protocols. Bluetooth is expected to provide a bearer service as specified by the WAP architecture.

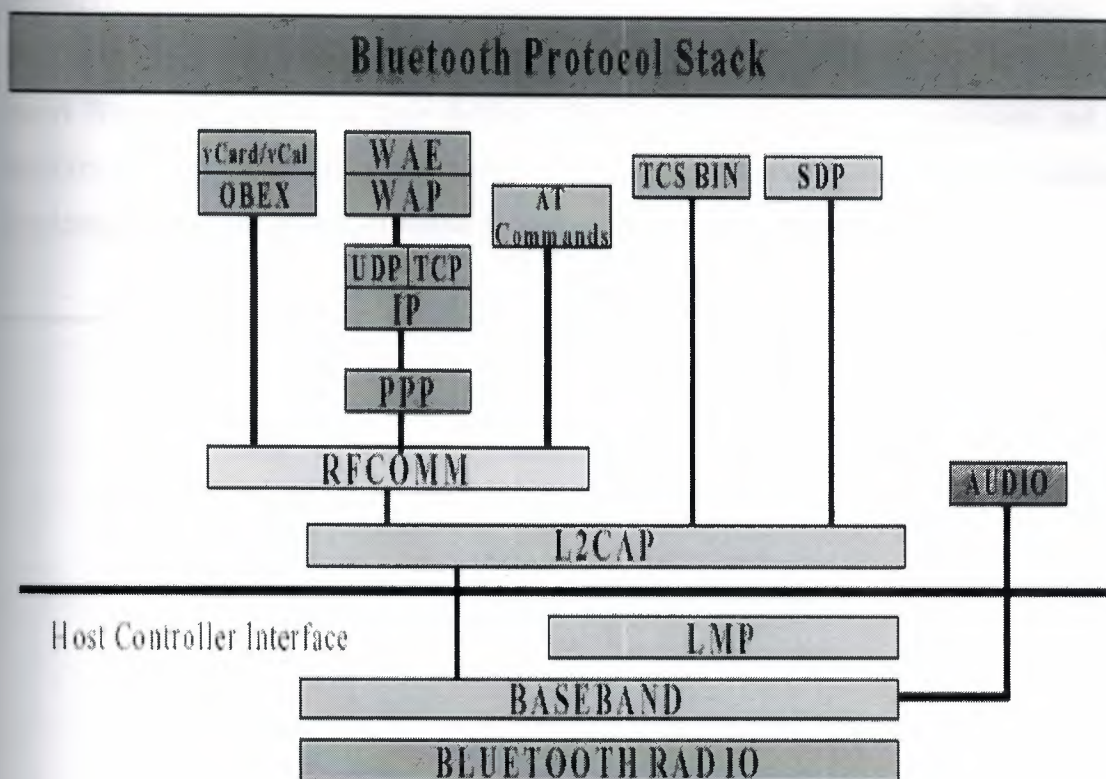


Figure 1.6. Bluetooth protocol stack

1.5 BLUETOOTH STRENGTHS AND FUTURE

The main advantages of Bluetooth are; the minimal hardware dimensions, low price on Bluetooth components, and the low power consumption for Bluetooth connections. The advantages make it possible to introduce support for Bluetooth in many types of devices at a low price. Both hardware and device manufacturers will work for the introduction of Bluetooth in many different devices. The capabilities provided by Bluetooth, approximately 720 kbit/s, can be used for cable replacement and several other applications such as speech, LAN, and so on. It is estimated that, Bluetooth will be a built-in feature in more than 100 million mobile phones and in several million other communication devices, ranging from headsets and portable PC's to desktop computers and notebooks. The first Bluetooth products were basic cable

replacement products. However, when the Bluetooth chips have entered the mass market and chips are found in a different number of devices, several new markets will open for Bluetooth solutions. A few software development kits (SDK) have now been introduced on the market. More competition on SDK market and lower prices on Bluetooth chips will make manufacturers of electronic equipment easy to convince to insert Bluetooth support in their devices. The Bluetooth hardware dimensions and its uniform method for building applications will ensure a Bluetooth market with matching implementations regardless of brand and what country the product is designed for.

CHAPTER TWO: WIRELESS NETWORKS

PART TWO: IEEE 802.11 STANDARDS

2.1 THE IEEE 802.11 STANDARD

A *Wireless LAN (WLAN)* is a data transmission system designed to provide location independent network access between computing devices by using radio waves rather than a cable infrastructure. The widespread acceptance of WLAN's depends on industry standardization to ensure product compatibility and reliability among the various manufacturers. The Institute of Electrical and Electronics Engineers (IEEE) ratified the original 802.11 specification in 1997 as the standard for wireless LAN's. 802.11 defines two pieces of equipment, a wireless station, which is usually a PC equipped with a wireless network interface card (NIC), and an access point (AP), which acts as bridge between the wireless and wired networks. An access point usually consists of a radio, a wired network interface, and bridging software conforming to the 802.11 bridging standard. The access point acts as the base station for the wireless network, aggregating access for multiple wireless stations onto the wired network. Wireless end stations can be 802.11 PC card, PCI, or NICs, or embedded solutions in non-pc clients.

The 802.11 standard defines two modes: *Infrastructure Mode* and *Ad-Hoc Mode*. In infrastructure mode (figure 1), the wireless network consists of at least one access point connected to the wired network infrastructure and a set of wireless end stations. This configuration is called a Basic Service Set (BSS). An Extended Service Set (ESS) is a set of two or more BSS's forming a single subnetwork. Since most corporate WLAN's require access to the wired LAN for services (file servers, printers, Internet links) they will operate in infrastructure mode. Ad-hoc mode (also called peer-to-peer mode or an Independent Basic Service Set or IBSS) is simply a set of 802.11 wireless stations that communicate directly with one another without using an access point or any connection to a wired network (figure 2). This mode is useful for quickly and easily setting up a wireless network anywhere that a wireless infrastructure does not exist or is not required for services, such as a hotel room, convention center, or airport, or where access to the wired network is barred.

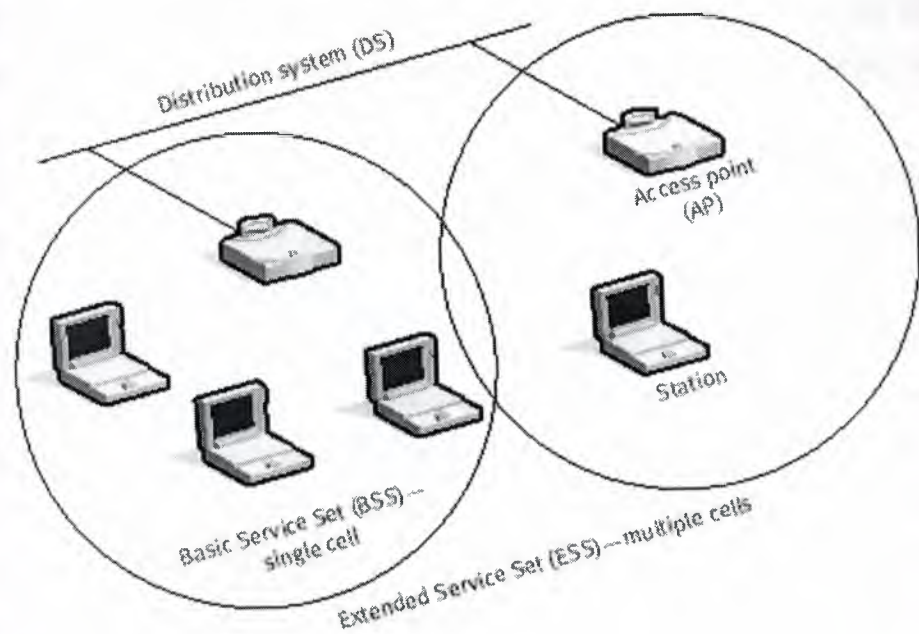


Figure 2.1. 802.11 Infrastructure mode

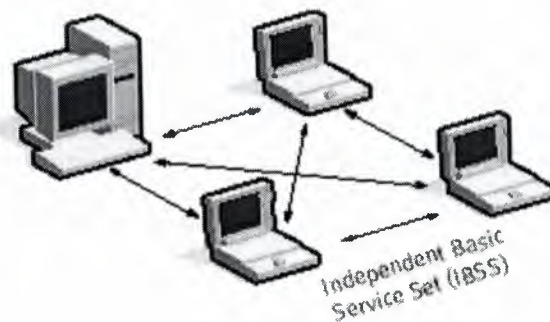


Figure 2.2. 802.11 Ad-Hoc Mode

The three physical layers originally defined in 802.11 included two spread-spectrum radio techniques and a diffuse infrared specification. The radio based standards operate within the 2.4 GHz ISM band. These frequency bands are recognized by international regulatory agencies. 802.11 based products do not require user licensing or special training. Spread-spectrum techniques, in addition to satisfying regulatory requirements, increase reliability, boost throughput, and allow many unrelated products to share the spectrum without explicit cooperation and with minimal interference. The original 802.11 wireless standard defines data rates of 1 Mbps and 2 Mbps via radio waves using *Frequency Hopping Spread Spectrum (FHSS)* or *Direct*

Sequence Spread Spectrum (DSSS). It is important to note that FHSS and DSSS are fundamentally different signalling mechanisms and will not interoperate with one another.

Using the frequency hopping technique, the 2.4 GHz band is divided into 75, 1-MHz subchannels. The sender and receiver agree on a hopping pattern, and data is sent over a sequence of the subchannels. Each conversation within the 802.11 network occurs over a different hopping pattern, and the patterns are designed to minimize the chance of two senders using the same subchannel simultaneously. FHSS techniques allow for a relatively simple radio design, but are limited to speeds of no higher than 2 Mbps. This limitation is driven primarily by FCC regulations that restrict subchannel bandwidth to 1 MHz. These regulations force FHSS systems to spread their usage across the entire 2.4 GHz band, meaning they must hop often, which leads to a high amount of hopping overhead. In contrast, the direct sequence signaling technique divides the 2.4 GHz band into 14, 22-MHz channels. Adjacent channels overlap one another partially, with three of the 14 being completely non-overlapping. Data is sent across one of these 22 MHz channels without hopping to other channels. To compensate for a noise on a given channel, a technique called “chipping” is used. Each bit of user data is converted into a series of redundant bit patterns called “chips”. The inherent redundancy of each chip combined with spreading the signal across the 22 MHz channel provides for a form of error checking and correction, even if part of the signal is damaged, it can still be recovered in many cases, minimizing the need for retransmissions.

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. Because wireless LAN's 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: This 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: This type of 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 the shared key authentication requires implementation of encryption via the Wired Equivalent Privacy (WEP) algorithm.

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 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.

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. 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 Wired Equivalent Privacy Algorithm (WEP).

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.

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 system. The mobile system invokes the association service once and only once, typically when the station enters the IBSS. Each station can associate with one access point though an access point can associate with multiple stations.

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 could 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.

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, losses contact with the access point with which it is associated, and needs to become associated with new access point. By using a 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.

Distribution: Distribution is the primary service used by an IEEE 802.11 station. A station uses the distribution service every time it sends Media Access Control (MAC) frames across the distribution system. The distribution service provides the distribution with only enough information to determine the proper destination IBSS 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.

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. The portal is an abstract architectural concept that typically resides in an access point though it could be part of the 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.

The 802.11 *Media Access Control* or *MAC* layer provides functionality to allow reliable data delivery for the upper layers over the wireless physical media. The data delivery itself is based on an asynchronous, connectionless delivery of MAC layer data. There is no guarantee that the frames will be delivered successfully. The 802.11 MAC provides a controlled access method to the shared wireless media called Carrier-Sense Multiple Access with Collision Avoidance (*CSMA/CA*). *CSMA/CA* is similar to the collision detection access method deployed by 802.3 ethernet LANs. The third function of the 802.11 MAC is to protect the data being delivered by providing security and privacy services. Security is provided by the authentication services and by Wireless

Equivalent Privacy (*WEP*), which is an encryption service for data delivered on the WLAN.

The fundamental access method of 802.11 is *Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA)*. It works by a “listen before talk scheme”. This means that a station wishing to transmit must first sense the radio channel to determine if another station is transmitting. If the medium is not busy, the transmission may proceed. The CSMA/CA protocol avoids collisions among stations sharing the medium by utilizing a random backoff time if the station’s physical or logical sensing mechanisms indicates a busy medium. The period of time immediately following a busy medium is the highest probability of collisions occurring, especially under high utilization. The CSMA/CA scheme implements a minimum time gap between frames from a given user. Once a frame has been sent from a given transmitting station, that station must wait until the time gap is up to try to transmit again. Once the time has passed, the station selects a random amount of time (the backoff interval) to wait before “listening” again to verify a clear channel on which to transmit. If the channel is still busy, another backoff interval is selected that is less than the first. This process is repeated until the waiting time approaches zero and the station is allowed to transmit. This type of multiple access ensures judicious channel sharing while avoiding collisions.

The 802.11 *Physical Layer (PHY)* is the interface between the MAC and the wireless media where frames are transmitted and received. The PHY provides three functions. First, the PHY provides an interface to exchange frames with the upper MAC layer for transmission and reception of data. Second, the PHY uses signal carrier and spread spectrum modulation to transmit data frames over the media. Thirdly, the PHY provides a carrier sense indication back to the MAC to verify activity on the media. 802.11 provides three different PHY definitions: FHSS, DSSS, and another extension to the standard that defines 11Mbps and 5.5Mbps data rates utilizing an extension to DSSS called High Rate DSSS (HR/DSSS).

Spread spectrum is a technique trading bandwidth for reliability. The goal is to use more bandwidth than the system really needs for transmission to reduce the impact of localized interference on the media. Spread spectrum spreads the transmitted bandwidth of the resulting signal, reducing the peak power but keeping total power the same. Frequency Hopping utilizes a set of narrow channels and “hops” through all of them in a predetermined sequence. For example; the 2.4 GHz frequency band is divided into 75 channels of 1-MHz each. Every 20 to 400 msec the system “hops” to a new

channel following a predetermined cyclic pattern. The 802.11 Frequency Hopping Spread Spectrum (FHSS) uses the 2.4 GHz radio frequency band, operating with at 1 or 2 Mbps data rate. The principle of Direct Sequence (DSSS) is to spread a signal on a larger frequency band by multiplexing it with a signature or code to minimize localized interference and background noise. To spread the signal, each bit is modulated by a code. In the receiver, the original signal is recovered by receiving the whole spread channel and demodulating with the same code used by the transmitter. The DSSS also uses the 2.4 GHz frequency band.

2.2 THE IEEE 802.11a/802.11b STANDARDS

The IEEE 802.11b standard is an extension to 802.11 that applies to WLANs and provides 11Mbps transmission in the 2.4 GHz band. 802.11b uses only DSSS and was a ratification to the original 802.11 standard allowing wireless functionality. 802.11b is currently the most successful implementation of 802.11 standards. It's fast and easy to implement, offering a full range of standards-based product infrastructure. The key contribution of the 802.11b addition to the wireless LAN standard was to standardize the physical layer support of two new speeds, 5.5 Mbps and 11 Mbps. To accomplish this, DSSS had to be selected as the sole physical layer technique for the standard since frequency hopping cannot support the higher speeds without violating current FCC regulations. The implication is that 802.11b systems will interoperate with 1 Mbps and 2 Mbps 802.11 DSSS systems, but will not work with 1 Mbps and 2 Mbps 802.11 FHSS system.

The original 802.11 DSSS standard specifies an 11-bit chipping, called a *Barker sequence*, to encode all data sent over the air. Each 11-bit chip sequence represents a single data bit (1 or 0), and is converted to a waveform, called a *symbol*, that can be sent over the air. These symbols are transmitted at a 1 MSps (1 million symbols per-second) symbol rate using a technique called *Binary Phase Shift Keying* (BPSK). In the case of 2 Mbps, a more sophisticated implementation called *Quadrature Phase Shift Keying* (QPSK) is used. It doubles the data rate available in BPSK, via improved efficiency in the use of the radio bandwidth.

To increase the data rate in the 802.11b standard, advanced coding techniques are employed. Rather than the two 11-bit Barker sequences, 802.11b specifies

Complementary Code Keying (CCK), which consists of a set of 64 8 bit code words. As a set these code words have unique mathematical properties that allow them to be correctly distinguished from one another by a receiver even in the presence of substantial noise and multipath interference (e.g. interference caused by receiving multiple radio reflections within a building). The 5.5 Mbps rate uses CCK to encode 4 bits per carrier, while the 11 Mbps rate encodes 8 bits per carrier. Both speeds use QPSK as the modulation technique. To support very noisy environment as well as extended range, 802.11b WLAN's use *Dynamic Rate Shifting*, allowing data rates to be automatically adjusted to compensate for the changing nature of the radio channel. Ideally, users connect at the full 11 Mbps rate. However when devices move beyond the optimal range for 11 Mbps operation, or if substantial interference is present, 802.11b devices will transmit at lower speeds, falling back to 5.5, 2, and 1 Mbps. Likewise, if the device moves back within the range of a higher-speed transmission, the connection will automatically speed-up again. Rate shifting is a physical layer mechanism transparent to the user and the upper layers of the protocol stack.

802.11b WLAN's communicate using radio waves, because these waves penetrate off many indoor structures or can reflect around obstacles. WLAN throughput depends on several factors, including the number of users, microcell range, interference, multipath propagation, standards support, and hardware type. Of course, anything that affects data traffic on the wired portions of the LAN, such as latency and bottlenecks, will also affect the wireless portion. When it comes to range, more is not always better. For example, if the network requirement is for high performance (5.5 Mbps or 11 Mbps) and complete coverage, long range at lower network speeds (1 Mbps and 2 Mbps) may make it difficult to employ a channel reuse pattern while maintaining high performance.

802.11a is supplement to IEEE 802.11 standard, specified for high speed physical layer in the 5 GHz band. Unlike the other standards in the 802.11 family, which are located in the unlicensed 2.4 GHz band, this standard intends to implement data connections in the frequency range of 5 GHz, thus circumvents the coexistence with bluetooth systems and family appliances. The highlights of the technologies used in 802.11a standard are *Orthogonal Frequency Division Multiplexing (OFDM)*, convolutional coding, QAM modulation and data scrambling and interleaving. Data rates supported are; 6, 9, 12, 18, 24, 36, 48, and 54 Mbps, where support of 6, 12, and

24 Mbps is mandatory. The data rate is determined by the combination of modulation and error correction coding schemes.

The underlined technology is the OFDM used in the system. The system uses 52 subcarriers that are digitally modulated, among which 4 are pilot channels, and 48 are data channels. The pilot signals are deployed in order to make the coherent detection robust against frequency offsets and phase noise. The pilots shall be BPSK modulated binary sequence to prevent the generation of spectral lines.

802.11a with its greater throughput than 802.11b, is more appropriate for video and multimedia applications. Let's compare some basic characteristics of 802.11a and 802.11b by looking at few parameters:

- ❖ 802.11b uses the Direct Sequence Spread Spectrum (DSSS) modulation scheme, while 802.11a uses Orthogonal Frequency Division Multiplexing (OFDM). In terms of power efficiency, DSSS is more efficient than OFDM. Thus 802.11a devices consume more power compared to 802.11b devices.
- ❖ 802.11b uses the 2.4 GHz spectrum, which is overcrowded with devices like cordless phones and microwave ovens. Even Bluetooth devices uses the 2.4 GHz spectrum. 802.11a on the other hand, uses the less crowded 5 GHz spectrum. Though the 5 GHz spectrum is less crowded, the signals have higher absorption rate and this causes it to be easily blocked by walls and objects.
- ❖ Due to the higher absorption rate at the 5 GHz spectrum, 802.11a devices have shorter operating range of about 150 feet, compared to the 300 feet achievable by 802.11b. As a result, more transmitters are required for 802.11a networks.
- ❖ Components for 802.11a devices are more expensive to produce and hence their price tags are higher than 802.11b devices. Also, the increased number of transmitters required for 802.11a network will drive up the cost of implementing an 802.11a network.
- ❖ 802.11a is not compatible with the 802.11b protocol. Hence 802.11a devices cannot work with existing 802.11b wireless access points.
- ❖ 802.11a can accommodate more users due to the increase in radio frequency channels and increased operating bandwidth.

The main draw of migrating to an 802.11a network is no doubt increased bandwidth. With the increased in data rate (54 Mbps), applications like audio and video

streaming and networking games would be possible. However, the drawback in adopting 802.11a is compatibility. Business and institutions that have invested in 802.11b networks are reluctant to migrate to a faster, but incompatible 802.11a network. For these reasons, vendors are coming out with dual-band wireless points. These dual-band access points support both 802.11a and 802.11b devices. You can deploy both 802.11a and 802.11b devices all in the same environment. Best of all, since those two protocols are operating in different frequencies, interference is minimized. Below you can see some of the 802.11a and 802.11b devices available in the market.



Wireless Access Point (802.11a, 802.11b).



Wireless Router and Print Server
(802.11b).



Wireless Access Point with 4-port switch
(802.11b).



Wireless Adapter (802.11b).

Figure 2.3. Some of the 802.11a/b devices

2.3 THE IEEE 802.11g STANDARD

With the 802.11g standard approved by the IEEE, it is generating a great deal of interest among wireless users. This standard makes available high data rates comparable to the 802.11a standard, but most importantly provides backward compatibility to the widely implemented 802.11b standard. Just like 802.11b devices, 802.11g devices operate in 2.4 GHz band except it uses the *Orthogonal Frequency Division Multiplexing* (OFDM) technology, unlike the Complementary Code Keying (CCK) modulation used by the 802.11b standard. OFDM is also used by 802.11a devices that operate in the 5 GHz band. The 802.11g standard also supports *Barker Code and CCK* modulating giving 1, 2, 5.5, and 11 Mbps data rates for backward compatibility with the 802.11b standard. OFDM provides; 6, 9, 12, 18, 24, 36, 48, and 54 Mbps data rates. The optional Packet Binary Convolution Coding (PBCC) encoding method provides data rates of 22 and 33 Mbps. The standard only includes data rates; 1, 2, 5.5, 11, 6, 12, and 24 as being mandatory for transmission and reception. Similar to the 802.11b standard, 802.11g devices are limited to three non-overlapping channels and the new physical layer is called the Extended Rate Physical (ERP) layer. Traditionally, though 802.11b and 802.11g devices communicate using CCK and OFDM modulation schemes respectively and because of backward compatibility, 802.11g devices have to support both modulation schemes. 802.11g employs *Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)*. The CSMA/CA protocol allows a device that is transmitting data exclusive transmitting rights. No other 802.11 device will transmit at that time and will begin transmission only when the medium is clear. This avoids collisions after waiting for random intervals of time. This random back-off interval is calculated using slot times multiplied with a random number.

OFDM signals are not heard by the 802.11b devices which will incorrectly access the medium to be free from transmission leading to collisions and reducing throughput. The 802.11g standard requires devices to employ protection mechanisms to improve the performance in a mixed 802.11b/g environment. 802.11g standard allows the use of two protection mechanisms in such an environment; *RTS/CTS and CTS-to-Self*. These protection mechanisms prepare 802.11g devices for communication with 802.11b devices using the CCK modulation scheme. In the RTS/CTS protection mechanism, the device wanting to communicate sends a *request-to-send (RTS)* message

to the destination node. It then receives a *Clear-to-Send (CTS)* message from the destination node indicating that the RTS message was received and that the data packet can be sent. This CTS message is heard by all the devices on the network and know that some transmission is taking place and will cease their own transmission appropriately. After receiving the CTS signal, the device sends data and waits for an ACK signal to verify that a successful transmission has transpired without collisions. In the CTS-to-Self protection mechanism the 802.11g access point will send a CTS message when it desires to send data, even though there was no RTS message received. Both these mechanisms help reduce collisions.

The immense benefit of backward compatibility is overshadowed by the lowernig of performance of the wireless network in a mixed environment. A WLAN with 802.11g only devices will provide higher throughput than what it will in a mixed environment with 802.11b devices. 802.11a devices do not impact the performance as they operate in the seperate 5 GHz spectrum. Different configurations on the 802.11g access points and 802.11b/g client devices will cause variety of problems on the WLAN. Problems could range from minor ones like devices sending frames at low speeds and use of non-standard speed transmissions rates, to major problems such as having a few client devices not using mechanisms to operate efficiently in a mixed environment. Also in certain cases these mechanisms, which are discussed later in the note, may cause more problems than what it attempts to solve. This makes it very important to understand minor details of various factors during implementation and maintaining 802.11g networks that will ultimately govern the solution which is unique for every 802.11g WLAN.

OFDM is a “multi-carrier” modulation scheme. The data is split up among several closely spaced subcarriers. By doing so, OFDM systems are able to provide very reliable operation even in environments that result in a high degree of signal distortion due to multipath. In addition, OFDM systems can support higher data rates than single carrier systems without incurring a huge penalty in terms of system complexity. For data rates up-to 11 Mbps, CCK is a very good choice. However, as data rates go higher, OFDM becomes a better choice. OFDM was selected for use in the 5 GHz bands primarily because it enables data rates up-to 54 Mbps to be realized.

IEEE 802.11g brings higher data rates (up to 54 Mbps) to the 2.4 GHz band and of equal importance, it ensures backward compatibility with existing wireless equipment. For owners of existing wireless equipment, 802.11g provides a smooth

migration path to higher data rates, thus extending the life of 2.4 GHz equipment. With high data rates and backwards compatibility for 802.11b devices, the 802.11g standard has a promising future in enterprise.

802.11h: This specification is a *European* variant of 802.11a with additional optimization features. These two 5GHz standards are nearly identical, except that 802.11h adds *TPC (Transmit Power Control)* which limits the wireless network card from emitting more radio signal than is needed, and *DFS (Dynamic Frequency Selection)*, which lets the device listen to what is happening in the airspace before picking a channel. TPC and DFS are required features in Europe.

2.4 COMPARISON OF BLUETOOTH AND IEEE 802.11 STANDARDS

→ The following table shows the general characteristics about Bluetooth and 802.11 standards:

Table 2.1. General characteristics of Bluetooth and 802.11 standards

	Bluetooth 1.1	Bluetooth Medium Rate	Bluetooth High Rate	802.11b	802.11g	802.11a
Application	General Wireless	General Wireless	General Wireless	Wireless Ethernet for PC	Wireless Ethernet for PC	Wireless Ethernet for PC
Speed	1 Mbps	2 Mbps	10 Mbps	11 Mbps	36-54 Mbps	24-54 Mbps
Range	10-100 m.	-	-	100-300 m.	-	-
Power Consumption	Low	Low	Medium	High	High	Very High
Bandwidth	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz	5 GHz

Bluetooth has specializations for very low cost, low power applications with simple interfaces to printers especially useful for non-PC devices like cell phones and PDAs. Audio-video interfaces for use with consumer devices such as headsets, speakers, music players, remote controls, and video conferencing are being added to the Bluetooth profiles. Bluetooth is intended for portable products, short ranges, and limited battery power. As a result, it offers exceptionally low power consumption and, in some cases, will not measurably effect battery life. On the other hand, 802.11 is designed for longer range transmission (up to 300 meters) and, by definition, must consume significantly more power. 802.11 is the WLAN of choice where range and throughput are more important than size and cost. Bluetooth is preferred where size, cost, and mobility are more important than range and throughput. 802.11 is too big for most mobile applications. It consumes 21 times more space than the first generation Bluetooth solutions. The key word for Bluetooth is mobile. While Bluetooth can be used in almost unlimited applications, its strongest benefit is for mobile workers and business travelers by permitting a notebook PC to connect to any wireless network via a cellular telephone. The key words for 802.11 are range and throughput. It is the WLAN of choice where these features are needed more than low cost, small size, efficient battery use, and mobility. In contrast, 802.11 remains relatively high cost (for enterprises, access points cost between \$500 and \$1000 and network interface cards range from \$100 to \$200). However, if range and throughput is more important than size and cost, then 802.11 is a clear winner over Bluetooth. Finally, both 802.11 and Bluetooth will continue refinement.

Table 2.2. General Characteristics of Bluetooth and 802.11 Standard

802.11	Bluetooth
<i>Optimized for:</i> Home/campus/office WLAN.	<i>Optimized for:</i> Cable replacement with limited wireless network capacity.
<i>Range:</i> 15-30 meters indoors and 300 meters outdoors.	<i>Range:</i> 10 meter range, 100 meter range with higher transmit power.
<i>Data rate:</i> Faster data rate. 802.11a: 24 to 54 Mbps. 5 GHz band. 802.11b: 2.4 GHz band. 802.11g: Extends existing 802.11b to 36-54 Mbps with full backwards compatibility.	<i>Data rate:</i> Slower data rate. 1 Mbps data rate for Bluetooth 1.1. 2 Mbps data rate for Bluetooth medium rate. 10 Mbps data rate for Bluetooth high rate.

<p><i>Frequency:</i> 802.11a: 5GHz OFDM in 20 MHz channels. 802.11b: 2.4 GHz, fixed 11 MHz channel (DSSS). 802.11g: 2.4 GHz, fixed 11 MHz channel (OFDM).</p>	<p><i>Frequency:</i> 2.4 GHz, 1600 hops/sec radio (FHSS) 1 MHz channels over 79 MHz.</p>
<p><i>Size:</i> Larger</p>	<p><i>Size:</i> Very highly integrated, smaller.</p>
<p><i>Cost:</i> Higher (average chip set price= \$30 to \$35).</p>	<p><i>Cost:</i> Lower (average chip set price= \$5)</p>
<p><i>Current consumption:</i> Shorter battery life for handheld/portable devices. Averages about 10x the power consumption of Bluetooth. Does not have a power saving protocol.</p>	<p><i>Current Consumption:</i> Longer/unaffected battery life.</p>
<p><i>Co-Existence with Bluetooth:</i> May interface if they are active in the same area. 802.15 working group is developing co-existence methods between 802.11 and Bluetooth wireless technology.</p>	<p><i>Co-Existence with 802.11:</i> May interface if they are active in the same area. 802.15 working group is developing co-existence methods between 802.11 and Bluetooth wireless technology.</p>
<p><i>Popular applications:</i> Desktop PCs, Notebook PCs, Wireless LANs, high-end palmtops</p>	<p><i>Popular applications:</i> Palmtops, Notebook PCs, printers, cellular telephones.</p>
<p><i>General:</i></p> <ul style="list-style-type: none"> ❖ Corporate wireless networks (LANs). ❖ Limited/expensive applications. <ul style="list-style-type: none"> ❖ Primarily desktop PCs/notebook PCs. 	<p><i>General:</i></p> <ul style="list-style-type: none"> ❖ Replace cables for keyboards, mouse, PDAs, cell phones, headsets, Notebook PCs, personal stereos (CD/MP3), speakers. ❖ Print/fax documents from cell phones or PDAs; synchronize PDAs with PCs. ❖ Send data to multible devices simultaneously. ❖ Wireless, multiplayer games. ❖ Automotive systems-arm/disarm security systems.

CHAPTER TWO: WIRELESS NETWORKS

PART THREE: WIRELESS SECURITY

3.1 SECURITY RISKS OF 802.11

Security is a principal consideration when planning, designing, implementing, and managing a network infrastructure. This is especially true for wireless LANs, which present a unique set of challenges to IT and security professionals. In addition to the typical problems that new network and device technologies engender, including incompatibilities and ongoing support issues, non-secure wireless LANs can expose an organization's network traffic and resources to unauthorized outsiders. Such individuals may capture data and exploit network based resources, including internet access, servers, and disk storage. More importantly, wireless access to a network can represent the entry point for various types of attacks, which can crash an entire network, render services unavailable, and potentially subject the organization to legal liabilities.

Wireless LANs are easy to find. To enable clients to find them, networks must transmit Beacon frames with network parameters. Of course, the information needed to join a network is also the information needed to launch an attack on a network. Beacon frames are not processed by any privacy functions, which means that 802.11 network and its parameters are available for anybody within an 802.11 card. Attackers with high-gain antennas can find networks from nearby roads or buildings and may launch attacks without having physical access to facility. Ensuring security on a wireless network is partly a matter of design. Networks should place access points outside of security perimeter devices such as firewalls,* and administrators should consider using *Virtual Private Network (VPN)* to provide access to the corporate network. Strong user authentication should be deployed, preferably using new products based on the IEEE 802.11x standard. 802.11x defines new frame types for user-based authentication. The 802.11x standard use *Remote Authentication Dial-in User Service (RADIUS)* server in conjunction with two data communication protocols: Extensible Authentication Protocol (EAP) and Transport Layer Security (TLS). A RADIUS server requires a user to login with a user name and password as well as answer an encryption key question. 802.11x implementations also improve data encryption through rotation of the WEP 128 key. It changes all the locks periodically and hacker could not steal the key.

802.11x is based upon an existing authentication protocol known as the Extensible Authentication Protocol (EAP) which in itself is an extension of Point-to-Point (PPP) Protocol. Figure 2 graphically shows the authentication process of 802.11x. Inappropriate configurations may be a major source of security vulnerability, especially if wireless LANs have been deployed without oversight from security engineers.

Easy access to wireless LANs is coupled with easy deployment. When combined, these two characteristics can cause headaches for network administrators and security officers. Any user can run to a nearby computer store, purchase an access point, and connect it to the corporate network without authorization. Many access points are now priced low. Departments may also be able to roll out their own wireless LANs without authorization from a central IT organization. End users are not security experts, and may not be aware of the risks posed by wireless LANs. The obvious way to find unauthorized networks is to do the same thing that attackers do: use an antenna and look for them so that you find unauthorized networks before attackers exploit them. Many tools can be used to perform site audits and track access points. An analyzer used for site audit work should be capable of simultaneously scanning for unauthorized access points.

VPN: *Virtual Private Network (VPN)* solutions are widely deployed to provide remote workers with secure access to the network via the Internet. In this remote user application, the VPN provides a secure, dedicated path (or tunnel) over an untrusted network, in this case, the Internet. Various tunneling protocols, including the *Point-to-Point Tunneling Protocol (PPTP)* and *Layer 2 Tunneling Protocol (L2TP)* are used in conjunction with standard, centralized authentication solutions, such as *Remote Authentication Dial-In User Service (RADIUS)* servers. The same VPN technology can also be used for secure wireless access. In this scenario, the untrusted network is the wireless network. The access points are configured for open access with no WEP encryption, but wireless access is isolated from the enterprise network by the VPN server. Authentication and full encryption over the wireless network is provided through the VPN servers that also act as firewalls/gateways to the internal private network. The VPN approach has number of advantages:

- ❖ Traffic to the internal network is isolated until VPN authentication is performed.
- ❖ WEP key and MAC address list management is not needed because of security measures created by the VPN channel itself.

- ❖ Addresses general remote access with a consistent user interface in different locations such as at home, at work, and in airport.

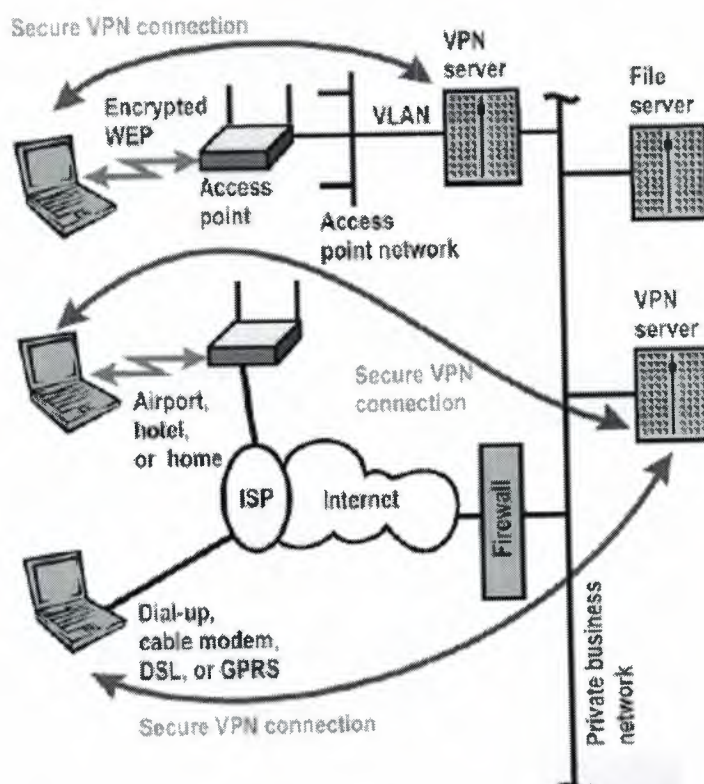


Figure 3.1. 802.11 VPN wireless security

WEP: Majority of access points are put in service with only minimal modifications to their default configuration. Nearly all of the access points running with default configurations have not activated *Wireless Encryption Privacy (WEP)* or have a default key used by all the vendor's products. A main function of WEP is to prevent unauthorized access to a wireless network. It relies on secret key that is shared between a wireless station and an access point. The secret key is used to encrypt packets before they are transmitted and an integrity check is used to ensure the packets are not modified in transit. Without WEP, network access is usually there for the taking. WEP is based on the RC4 stream cipher, a symmetric cipher where the same key is used for both encryption and decryption. The original 802.11 standard uses 40-bit WEP key. Cryptographically stronger 104-bit keys implementations are provided by a number of WLAN vendors. Figure 1 graphically illustrates WEP. Two problems can result from such open access. In addition to bandwidth charges for unauthorized use, legal problems

may result. The obvious defense against unauthorized use is to prevent unauthorized users from accessing the network. Strong, cryptographically protected authentication is a precondition for authorization because access privileges are based on user identity. VPN solutions deployed to prevent traffic in transit across the radio link provide strong authentication. Once a network has been successfully deployed, it is vital to ensure that authentication and authorization policies are followed. As with the access point problem, the solution is to perform regular audits of the deployed wireless network equipment to ensure that strong authentication mechanisms are in use and that network devices are properly configured.

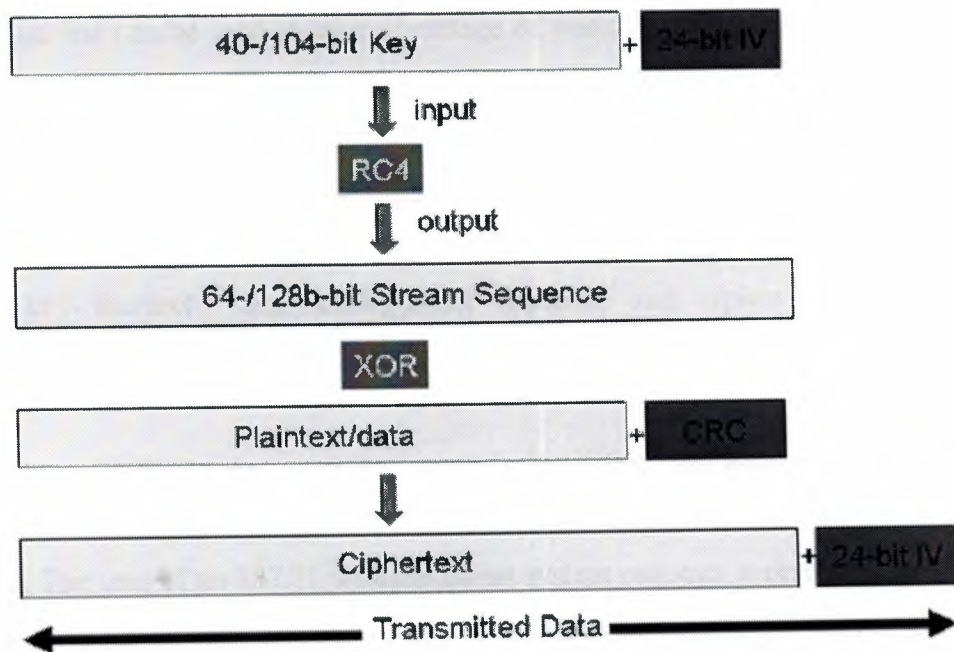


Figure3.2. Illustration of WEP mechanism

→ WEP functions as follows:

1. A secret key (either 40 or 104 bits) is concatenated with a 24-bit *Initialization Vector (IV)* resulting in a 64 or 128 bit key. An IV is added to the secret key in each packet to ensure that each packet has a different RC4 key (given that the secret key doesn't change frequently).
2. The key input into the RC4 Pseudorandom Number Generator (PRNG), resulting in pseudorandom keystream of the same length as the initial key (either 64 or 128 bits).

3. The plaintext (data) is run through an integrity checking algorithm resulting in a checksum (CRC). This checksum (the CRC in figure) is concatenated onto the plaintext so that the integrity of the information may be checked by the decrypting party.
4. The data vector, i.e., data + checksum vector from step 3, is encrypted by doing a bitwise XOR with the keystream from step 2 above, which results in the ciphertext.
5. The IV is appended to the ciphertext and the result is transmitted via wireless.

802.11i: 802.11i is a new security standard being developed by the IEEE taskgroup. 802.11i addresses the weaknesses of WEP based wireless security. Scripting tools exist that can be used to take advantage of weaknesses in the WEP key algorithm to successfully attack to a network and discover the WEP key. The 802.11i standard addresses the user authentication and encryption weaknesses of WEP based wireless security. The components of 802.11i include 802.11x port-based authentication framework, the *Temporal Key Integrity Protocol (TKIP)*, the *Advanced Encryption Standard (AES)*, key hierarchy and management features, and cipher and authentication negotiation.

802.11x: 802.11x is used to securely establish an authenticated association between the client and the access point. Generally, the scenario would be as shown in figure 3. The user of an 802.11 wireless client system requests access to an access point. The access point passes the request to a centralized authentication server that handles the authentication exchange and, if successful, provides an encryption key to the access point. The access point uses the key to securely transmit key to the client. At this point, the client has access to the network, transmissions between the client and access point are encrypted, and user may log on to the network domain. During the session, new keys are generated between the client and access point (referred to as dynamic WEP key exchange) to help stop exposure to WEP attacks. 802.11x does not require a specific protocol for authentication. Instead, it specifies that the Extensible Authentication Protocol (EAP) will be used. EAP is an encapsulation protocol that allows different authentication protocols to be selected and used. Effectively, EAP serves as a conduit for other authentication protocols. The 802.11x approach has the following advantages:

- ❖ Client keys are dynamically generated and propagated.

- ❖ Scalable to large enterprise networks by simply adding access points and, as needed, additional RADIUS servers.
- ❖ Flexible authentication; administrators may choose the type of authentication method used.

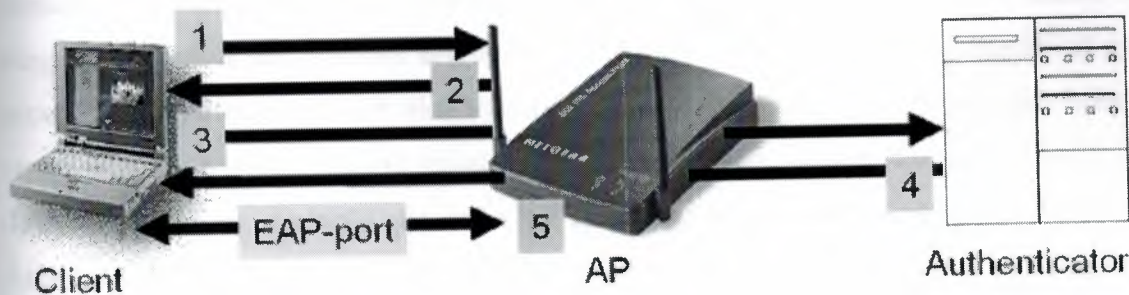


Figure 3.3. Authentication process of 802.11x

→ 802.11x authentication occurs as follows:

1. The client sends a request for authentication to the Access Point (AP).
2. Access point replies with a request that the client provide identification, and blocks all other traffic, such as HTTP and POP3 packets, until the AP can verify the client's identity using the authentication server.
3. The client sends a response containing the identity to the authentication server.
4. The authentication server receives the request and uses an appropriate authentication algorithm to verify the client's identity. If the user can be identified, an accept message is sent to the AP, otherwise a reject message is sent.
5. If the authentication server accepts the client, then the AP will transition the client's port to an authorized state and forward additional traffic.

802.11x can also provide dynamic key management as a delivery mechanism. Through dynamic key exchange the authentication server can return session keys to the AP along with the accept message. In step 3 above, rather than returning a simple accept or reject message, the authenticator returns both the results of authentication plus a session key. The AP uses the session keys from the authentication server to sign and encrypt a message that is forwarded to the client after sending the success message (step 4). The client then can use contents of the key message to define appropriate encryption

keys (step 5 and thereafter). The mechanism for dynamic key management provides a more secure mechanism than the manual maintenance of keys. The 802.11x mechanism allows clients, through the use of dynamic key management, to automatically change encryption keys as often as necessary to minimize the possibility of a passive attack.

Wireless LANs have limited transmission capacity, an attacker might launch a denial of service (DoS) attack on the limited resources. If an attacker were to launch a ping flood from a fast ethernet segment, it could easily overwhelm the capacity of an access point. Attackers could also inject traffic into the radio network without being attached to a wireless access point. The 802.11 *Media Access Control (MAC)* is designed to allow multiple networks to share the same space and radio channel. Attackers wishing to take out the wireless network could send their own traffic on the same radio channel, and the target network would accommodate the new traffic as best it could using the CSMA/CA mechanisms in the standard. Malicious attackers who transmit spoofed frames can also overwhelm limited capacity. Attackers may also use simple radio jamming techniques and send high noise transmissions at a target wireless network. The only practical defense against flooding attacks is to locate attackers and apply an appropriate solution.

802.11 networks do not authenticate frames. Attackers can use spoofed frames to redirect traffic, at much simpler level, attackers can observe the MAC addresses of stations in use on the network and adopt those addresses for malicious transmissions. To prevent this class of attacks, user authentication mechanisms are being developed for 802.11 networks. By requiring authenticating by potential users, unauthorized users can be kept from accessing the network. 802.11x can be used to require that users authenticate before accessing the network. Attackers can use spoofed frames in active attacks as well. In addition to hijacking sessions, attackers can exploit the lack of authentication of access points. Attackers can easily pretend to be an access point because nothing in 802.11 requires an access point to prove it really is an access point. Using methods based on *Transport Layer Security (TLS)*, access points will need to prove their identity before clients provide authentication credentials, and credentials are protected by strong cryptography for transmission. Until the retification of 802.11i, MAC spoofing will be a threat. Network engineers must focus on containing any damage done by MAC spoofing by isolating wireless networks from the more vulnerable core network.

Network access control can be implemented using an *Service Set Identification (SSID)* associated with an access point or group of access points. The SSID provides a mechanism to segment a wireless network into multiple networks serviced by one or more access points. Each access point is programmed with an SSID corresponding to a specific wireless network. To access this network, client computers must be configured with the correct SSID. A building might be segmented into multiple networks by floor or department. Typically, a client computer can be configured with multiple SSIDs for users who require access to the network from a variety of different locations. Because a client computer must present the correct SSID to access the access point, the SSID acts as a simple password and, thus, provides a measure of security. However, this minimal security is compromised if the access point is configured to broadcast its SSID. When this broadcast feature is enabled, any client computer that is not configured with a specific SSID is allowed to receive the SSID and access the access point. In addition, because users typically configure their own client systems with the appropriate SSIDs, they are widely known and easily shared.

802.11 provides no protection against attacks which passively observe traffic. The main risk is that 802.11 does not provide a way to secure data in transit against attackers. Frame headers are always “in the clear” and are visible to anybody with a wireless network analyzer. Security against traffic analyzers was supposed to be provided by the *Wired Equivalent Privacy (WEP)* specification. The current WEP security is keys and dynamic re-keying. WEP has been extensively studied and the security protocols have been fortified against all known attacks. A critical component of this fortification is the short re-keying time, which prevents attackers from learning a great deal about the properties of a WEP key before it is replaced. Strong cryptographic solutions like, SSH, SSL, and IP Security were designed to transmit data securely over public channels and have proven resistant to attack over many years, and will almost certainly provide a higher level of security.

3.2 WHY IS 802.11 WIRELESS NETWORKING TECHNOLOGY INSECURE?

- ❖ Wireless access points are open – Wireless access points are essentially radio antenna hard wired to network servers. The signal emitted from the access point

radiates in a circle pattern of roughly 300 feet radius. Anyone within the range of a given access point can potentially access the network. Access points are, by their nature open. The default settings for the access point hardware, authorization and management are almost always the most insecure settings. Free software is readily available that will let anyone use their laptop or other device to listen to 802.11 traffic. Since 802.11 signals can pass well outside the walls of a building, potentially anyone in their car outside the building can pick up on 802.11 network communications.

- ❖ Wired equivalency privacy (WEP) encryption protocol has been broken – The standard encryption protocol built into 802.11 has been proven to be insecure. The key structure used by WEP can be broken by brute force attack. WEP will discourage the casual attacker but cannot be trusted to protect the network from the determined attacker. Work is being done to make WEP more secure as well as to create a new security protocol for wireless communications.

3.3 WAYS TO SECURE AN 802.11 NETWORK

- ❖ Virtual private network – By far, the most effective and complete way to secure 802.11 network is through the use of a virtual private network. VPN technology was created to provide secure communications over the public Internet system by encrypting data sent over the public system. Since 802.11 communications are essentially public, it makes sense that VPN technology would be an appropriate solution for wireless communications as well. Placing a firewall between an access point and network, thereby creating a *DeMilitarized Zone (DMZ)*, and requiring all communications to come via VPN is the *MOST* effective way to protect a network.
- ❖ Check network for unauthorized access points – Since 802.11 technology is so cheap and simple to deploy, it is not difficult for employees (or non-employees) with physical access to your network to create an unauthorized access point that is not properly protected and could potentially undo all your careful security plans. Policy is critical here, employees should know that insertion of a wireless access point is subject to constraints and security concerns.
- ❖ Enable WEP – By default, WEP is disabled on wireless network equipment. Even though WEP has known problems it will help to protect your network from hackers.

- ❖ Utilize a RADIUS server – User based authentication provides a centrally managed method of authenticating users attempting to access the wireless network. A RADIUS (Remote Authentication Dial-In User Service) server provides this functionality. Utilizing a RADIUS server does not address the security of communications while they are “in the air”, it only prevents unauthorized people from accessing the “wired” network.

3.4 SECURITY RISKS OF BLUETOOTH

The Bluetooth wireless technology system contains a set of profiles. A profile defines a selection of messages and procedures (generally termed capabilities) from the Bluetooth specifications. This gives an unambiguous description of the air interface for specified devices and use cases. A security architecture defines the protocols and functionality required to implement the security elements within a specific application category. The rules that determine the access rights to different resources on the devices are called the access policy. The access policy together with the description of the usage of basic security mechanisms like authentication and encryption make up the *security policy*. The security policy is part of the security architecture for a Bluetooth application profile. There are many security threads inside the Bluetooth profiles, we will discuss them one by one.

The service discovery application profile describes the features and procedures used to discover services registered on other Bluetooth wireless devices using the Bluetooth *Service Discovery Protocol (SDP)*. The SDP itself does not require the use of authentication and/or encryption for SDP transactions. If authentication is performed on the Bluetooth wireless devices to be involved in an SDP procedure, then the devices must pass the authentication test to perform SDP procedures. Therefore any security procedures applied to the SDP are determined by those used to negotiate the connections between the specific Bluetooth wireless devices. SDP is not available to devices that do not pass this test. Since SDP security is based on device access to the SDP service, security may be provided by restricting access to trusted devices. In the case of a connection between untrusted or unknown devices, the service record is freely available, since no security is applied. This, however, is acceptable in many situations

since the SDP only provides a record indicating what services are available, not a mechanism to access these services.

Another Bluetooth profile is called Headset profile. As shown in figure 3, Bluetooth profiles, the "*Headset Profile*" depends on both the "*Serial Port Profile*" and the "*Generic Access Profile*". The serial port profile provides RS-232 serial cable emulation for Bluetooth wireless devices. The generic access profile (GAP) describes several security aspects of Bluetooth wireless connections. Since the headset profile inherits characteristics from the GAP, these aspects also apply to the headset profile. A typical headset configuration consists of two devices; a *Headset (HS)* and an *Audio Gateway (AG)*. The AG is typically a cellular phone, laptop, PC, or any other type of audio player device, such as a radio, CD player, etc. For reasons, which include personal privacy and preventing infringement on others, it is recommended that communication between the HS and AG be protected by the Bluetooth baseband authentication and encryption mechanisms. How and when these mechanisms should be used is determined by policy rules, which may be preset or configurable by the end user. Since the HS will normally not have a user interface, it is appropriate to assume that an external device, such as the AG, may control some of the basic settings of the HS (devices to be connected). Apart from the pure authentication, encryption and key storage functions, the HS and AG entities need to use an access policy to provide, for example, for audio connections and for the remote control of the HS. To provide alternative means for modifying AG and HS functionality, such as application program updates, security access or control policy changes, etc., the device manufacturer may provide a serial port interface. Use of such a wired means of connection to the AG or HS provides a highly secure means for initializing or modifying device operating parameters.

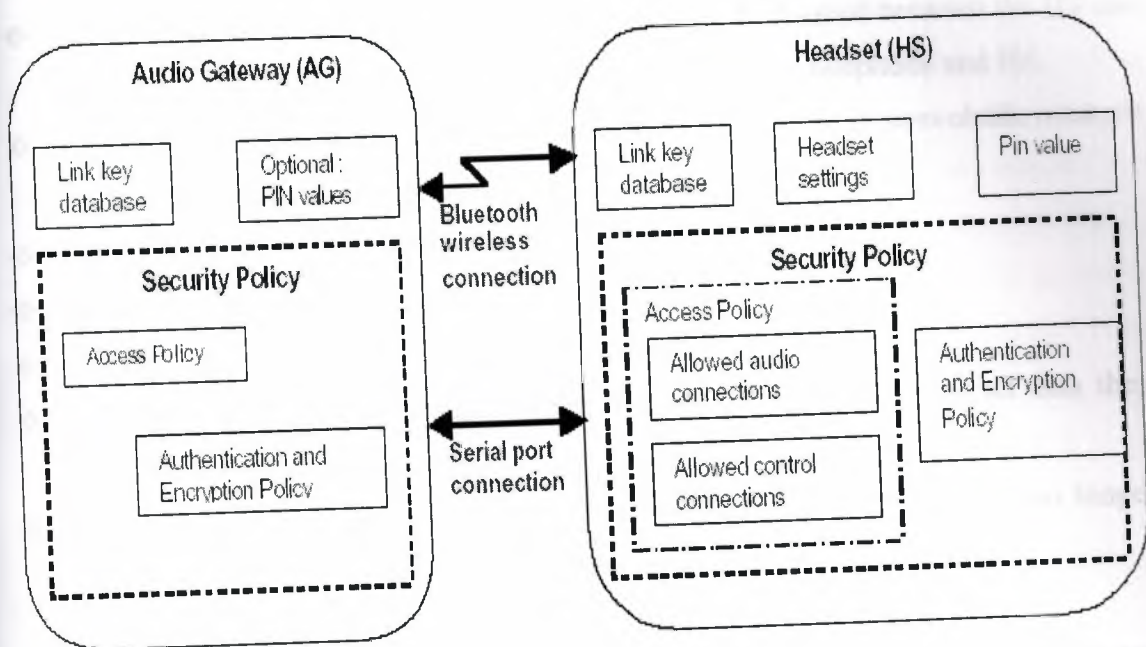


Figure 3.4. Bluetooth headset security architecture

A pairing and connection example is provided below. There are several ways of implementing HS security and HS control. Here, the use of Bluetooth wireless connection is assumed. Assume a new HS is delivered to a customer. The customer would like to use the HS together with his mobile phone acting as the AG. The HS is delivered with a preset Bluetooth *passkey* known to the customer. This passkey is intended for use in the Bluetooth *Link Manager Protocol (LMP)* link key. The customer and the pairing units perform the following steps before the customer is able to use the HS together with the mobile phone:

- ❖ Customer sets the HS into pairing mode by pressing a button on the HS.
- ❖ The HS indicates to the user that it is ready for pairing.
- ❖ The customer prepares his mobile phone for discovery of a new Bluetooth HS device.
- ❖ The phone performs a Bluetooth inquiry and gets a response from the HS.
- ❖ As part of the LMP channel set-up, the HS demands authentication of the phone.
- ❖ The phone detects that it does not have any previous link key with the HS. The Bluetooth pairing is requested.
- ❖ The phone prompts the user to enter the passkey for the HS.

- ❖ The customer inputs the passkey. A key exchange is performed between the HS and the phone. A link key is derived that is shared between the telephone and HS.
- ❖ The new link key between the HS and the telephone is stored in nonvolatile memory in both the phone and the HS unit.
- ❖ The HS authenticates the phone.
- ❖ The phone authenticates the HS.
- ❖ The HS and the phone perform an encryption key exchange.
- ❖ The LMP set-up is now complete. The HS and the phone encrypt all data they exchange from now on.
- ❖ The customer now switches the HS out of the pairing mode so it will no longer accept any new inquiries or pairing requests.

At this point, the HS will only accept connections from the telephone with it was paired. The HS will also require authentication and encryption before any LMP channel set-up can be completed. Authentication is now based on the link key. If the HS is stolen, the illegal user can try to set-up a connection with it.

The Bluetooth wireless technology offers authentication and encryption mechanisms on the Baseband level 1. They can be used to protect Bluetooth point-to-point links. The baseband security is based on the link keys that are determined for each particular Bluetooth device pair. The link key is derived during a pairing procedure. At the pairing step the Bluetooth wireless device user must either enter a Bluetooth passkey, or the same Bluetooth passkey must be available at the two Bluetooth units by some other means. This part describes how the built-in Bluetooth baseband security can be used for the *Dial-up Networking Profile*.

In the dial-up networking profile Bluetooth baseband security is required. The profile contains a basic description of the application of Bluetooth security procedures. The dial-up networking profile only supports one connection at a time. The profile is typically used for private modem connections. Depending on the type of Gateway (the device that provides the access to the public network) device the user security configuration possibilities might vary. A mobile phone can have a rather advanced security policy, while a modem might have very limited configuration possibilities and hence does not allow the implementation of an advanced security policy. Typically the modem does not have a user interface. Therefore it is appropriate, to assume that an external device, such as the Data Terminal (the device that uses the dial-up service of

the gateway) controls some of the modem settings. The an architecture similar to the one described for the headset profile applies. It is important to provide basic protection of the Bluetooth over-the-air transmission. The Bluetooth baseband authentication and encryption should be used to protect the link. In order to set-up secure connections, the Data Terminal and Gateway need to store the necessary link keys. Assume a user with a laptop borrows a mobile phone from a friend. The user would like to use the phone to get Internet access through the mobile phone using the dial-up networking profile. Here a short description of the different steps at connection set-up is given:

- ❖ The phone owner switches the phone on, sets the phone into discoverable mode, and switchs it into "one time secure connection" mode. This is done through a dedicated security menu.
- ❖ The Data Terminal user switches the Data Terminal into the "one time secure connection" mode. This is done through menu on the Data Terminal.
- ❖ The user then asks the Data Terminal to discover neighboring Bluetooth devices.
- ❖ The Data Terminal tries to set-up a connection with the mobile phone. As part of the Link Manager Prtocol (LMP) channel set-up, the Data Terminal demands authentication of the phone.
- ❖ The Data Terminal and mobile phones are bonded and the user is asked to enter the same Bluetooth passkey into both devices. The user creates the Bluetooth passkey preferably using a dedicated Bluetooth passkey generation application in the Data Terminal or in the phone.
- ❖ A pairing between the two devices is performed and a common Bluetooth combination key is calculated.
- ❖ A common link key is stored in the Data Terminal and the Gateway.
- ❖ Authentication is performed and the encryption key is exchanged between the devices.
- ❖ The LMP connection establishment is completed.
- ❖ A serial port emulation connection is established between the Data Terminal and the phone.
- ❖ The Data Terminal uses commands to set-up the desired dial-up connection.
- ❖ Dial-up data can securely flow between the Data Terminal and the phone.

- ❖ When the call is finished, the Bluetooth wireless connection is released. The recently derived encryption keys in the Data Terminal and phone Bluetooth modules are deleted.

LAN access is one example where one Bluetooth wireless device might be connected to several different types of devices. This section shows how the Bluetooth baseband security mechanisms can be utilized to secure communication for LAN access. There are two different roles defined in the LAN profile; the *LAN Access Point (LAP)* and the *Data Terminal (DT)*. The LAP is the Bluetooth wireless device that provides access to the LAN (e.g. Ethernet, USB, cable modem). The DT uses the services of the LAP. Typical devices acting as data terminals are laptops, notebooks, desktop PC's and PDA's. The following scenarios are described in the LAN access profile:

1. A single DT uses a LAP as a wireless means for connecting to a LAN.
2. Multiple DT's use a LAP as a wireless means for connecting to a LAN.
3. PC-to-PC connection where two Bluetooth wireless devices can form a single connection with each other.

For a communication service several different security aspects must be taken into account. These aspects cover everything from protection of communication links (provided by encryption and/or data integrity protection), authentication of devices or users and access control. Different security mechanisms can be applied at different layers in the communication stack. Furthermore, protection at one layer does not exclude protection at another level. The security demands depend on the application, the environment and usage scenario. Authentication and encryption can be provided on IP or application level. In order to set-up secure connections, the LAP and DT need to store necessary keys. In order to obtain high security level the LAP should be either be physically protected so that unauthorized users cannot manipulate it or that the LAP some build in security mechanisms that prevent unauthorized access to it. The LAP can be managed by a user interface or through network management. It is important to provide basic protection of the Bluetooth air transmission. It is required that the air interface is as secure as the fixed LAN connections on the other side of the LAP. Hence, the Bluetooth baseband authentication and encryption can preferably protect the link. In

order to set-up secure connections, the LAP and DT need to store necessary keys. Bluetooth link keys are suitable for this purpose. The LAN access profile is defined over Point-to-Point (PPP) protocol. PPP supports a number of different authentication and encryption protocols. These security protocols can be used in addition to the protection provided by the baseband. For example, PPP authentication might be required to authenticate DT users. The LAP and DT should use long random Bluetooth passkeys, because using short passkey values is a potential security risk.

Assume a user would like to register his DT for getting LAN access through LAPs installed by a certain LAN access service provider or organization. A user who registers the DT at the LAN access provider might do this using a non-Bluetooth wireless procedure. When a DT user subscribes to a LAN access service it gets a unique ID that identifies the service provider. Together with the ID the user also receives a secret Bluetooth *passkey*. The passkey is used to perform a Bluetooth bonding. In order to provide high security for the system a long passkey value must be used. The Bluetooth passkey is generated by the LAN access service provider using a secure random generator and is unique for each DT subscriber in the LAN. The DT user needs to manually enter into in a well protected LAN access service database the two values:

- 1) LAN access service ID.
- 2) Bluetooth passkey for the particular LAN access service.

Also at the registration the user is given to the LAN access provider a unique DT ID. This ID can be LAN access specific or it can be the DT Bluetooth wireless device address. As part of the subscription, the LAN access provider stores the Bluetooth passkey and corresponding DT ID in a central secure database. All LAPs in the access need to have secure access to this database as described in figure 3.

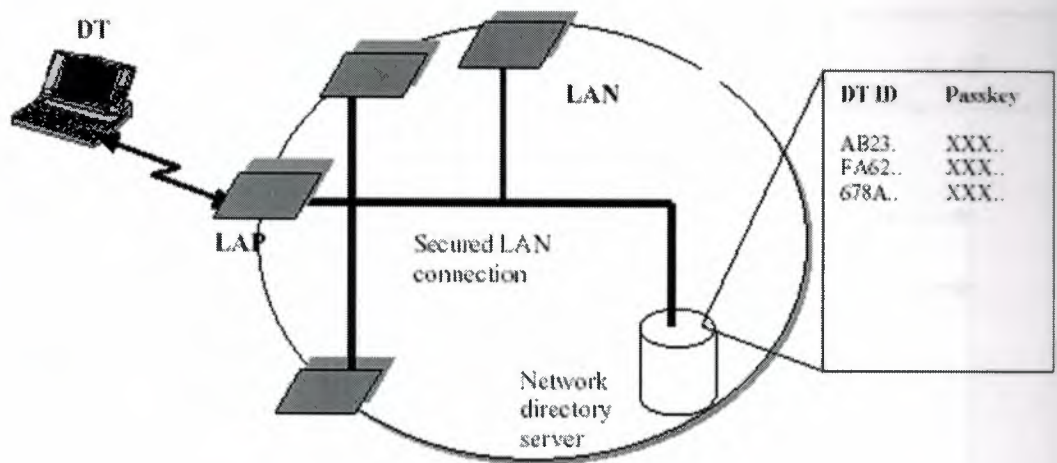


Figure 3.5. LAN access network architecture with subscription server containing users Bluetooth keys.

→ Below is a detailed description of each step in the pairing procedure:

- 1) The DT connects to the LAP or the LAP connects to the LP.
- 2) The DT searches for LAN access service record on the LAP. The DT receives the service ID of the LAP.
- 3) The DT checks that it knows the service ID received over the SDP protocol. Otherwise, the DT interrupts the connection procedure.
- 4) The DT asks the internal service database for the Bluetooth passkey corresponding to the service ID.
- 5) The LAP makes a secure network connection towards the network directory server to obtain the Bluetooth passkey corresponding to the received DT ID.
- 6) The DT and LAP performs a Bluetooth bonding using the Bluetooth passkey obtained from the databases. As a result of the bonding the DT and LAP share a common link key.
- 7) The DT uses the HCI command “write-stored-link-key”, to store the derived key in the Bluetooth wireless module. The key is also stored as a group key for the LAP service in the key database of the host.
- 8) The LAP uses the HCI command “write-stored-link-key”, to store the derived key in the Bluetooth wireless module. The key is stored as a group key for the DT in the network directory server. The key might be identified by the *BD_ADDR* of the DT.

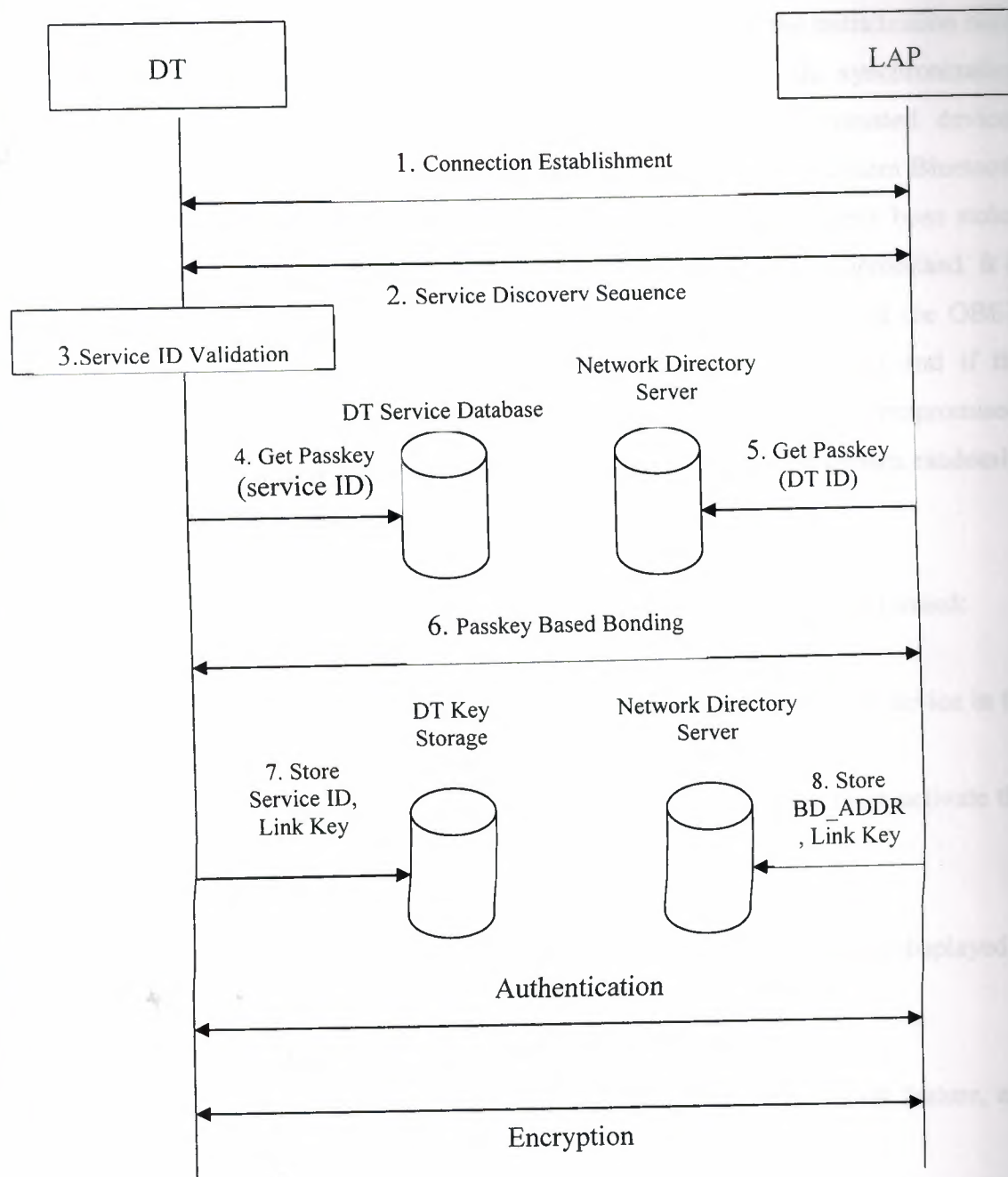


Figure 3.6. Steps in the pairing procedure

This section describes security solutions and usage models for the Bluetooth synchronization profile. The synchronization profile defines requirements, protocols, and procedures to be used for applications that implements this usage model. This profile allows security mechanisms at baseband, OBEX, IrMC client/server and UI level. At baseband we have the usual Bluetooth authentication and encryption. On user interface level authorization for access security can be used. The OBEX authentication depends on an initialization procedure in which a common password must be entered

and stored in both devices. If OBEX authentication is to be used, the initialization must occur before the first OBEX connection is established. Access to the synchronization service shall be automatically granted to trusted devices. For untrusted devices authorization on the IrMC server device is required. If short or non-random Bluetooth passkeys are used, and if it is possible that the exchanged messages have been stolen during the pairing, it is possible that the derived link key has been compromised. It is important to protect the connection identifier of the OBEX session, and the OBEX authentication response messages. If weak OBEX passwords are used and if the Bluetooth link keys have been compromised, then the overall security is compromised. For this reason it is also required that OBEX passwords be long and chosen randomly, independent from the Bluetooth passkey.

→ Here is an example showing how the initial synchronization may be performed:

1. The user performs bonding of the devices. He may register the server device in the trusted devices Database of the client device, and vice versa.
2. The IrMC server must be in connectable mode. If not, the user must activate this mode on the device.
3. The user activates an application for synchronization.
4. A list of devices in the Radio Frequency proximity of the IrMC client is displayed to the user.
5. The user selects a device to be connected and synchronized.
6. The user is alerted if the device does not support the synchronization feature, and the user may select another device.
7. If the IrMC client device is untrusted, the user is alerted that synchronization will be performed. By some device specific interaction the user accepts this.
8. The first synchronization is processed.
9. The user may be notified of the result of the operation.

Like most advances, wireless LANs pose both opportunities and risks. The technology can represent a powerful complement to an organization's networking capabilities, enabling increased employee productivity and reducing IT costs. To minimize the attendant risks, IT administrators can implement a range of measures, including establishment of wireless security policies and practices, as well as

implementation of various LAN design and implementation measures. Achieving this balance of opportunity and risk allows enterprises to confidently implement wireless LANs and realize the benefits this increasingly viable technology offers.

CONCLUSION

This project has discussed about the Information Security & Wireless Networks. How hackers attack, ways to secure our systems and various kinds of protection/encryption mechanisms are explained in the first chapter. The goals of hacker is to gain access to individual computer, discover passwords and credit card numbers, and also cause damage by deleting files. As a conclusion I can say that there is no way to completely secure our systems from hackers, but we can reduce attacks by implementing/using firewalls, antivirus programs and update them regularly. A person should use hard to crack passwords in the systems. Different strong encryption techniques are also used for privacy. SSL, PGP, SSH, and Kerberos are widely used protection mechanisms, which I explained their structure and functions in detail. How viruses and trojans spread, spam, and spyware explained in the first chapter.

In the second chapter I discussed about wireless technologies and their security concerns. Bluetooth and IEEE 802.11 standards are widely used for wireless communications. With Bluetooth and 802.11, users will be able to connect a wide range of computing and telecommunications devices easily and simply, without the need for connecting cables. I briefly explained how they work, their structure and also compared both two standards. At the end I explained how to protect wireless devices from hackers. The most effective and complete way to secure wireless networks is to use Virtual Private Network (VPN). Wireless Encryption Privacy (WEP) should also be enabled on wireless devices.

REFERENCES:

- [1] William Stallings, Cryptography and Network Security Principles and Practice, Prentice Hall Inc. 1999.
- [2] Various kinds of documents about Information Security and Wireless Networks, retrieved March-April 2004 from "<http://www.itpapers.com>".
- [3] Computer Viruses and Trojans, retrieved March-April 2004 from "<http://www.symantec.com>".
- [4] Methods employed by Hackers and Crackers, retrieved April 2004 from "<http://www.astalavista.com>".
- [5] Various kinds of documents about Information Security, retrieved March 2004 from "<http://www.faqs.org>".
- [6] Various kinds of documents about Information Security, retrieved March-April 2004 from "<http://www.windowsecurity.com>".