University of Technology
الجامعة التكنولوجية

Computer Science Department
قسم علوم الحاسوب

أمنية البيانات

أ . د. هالة بهجت عبد الوهاب

cs.uotechnology.edu.iq

University of Technology
Computer Sciences Department
Network Management
Branch

Chapter 1

1. Introduction to Data Security

Security is the sum of all measures taken to prevent loss of any kind. Loss can occur because of user error, defects in code, malicious acts, hardware failure, and acts of nature. *Security is a compromise between opening the systems up to the world and locking them down so no one can use them.*

2. Requirements for computer protection

Computer security is based on the same concepts that physical security is: trust, knowledge of a secret to prove authenticity, possession of a key to open locks, and legal accountability. The metaphors are so apt that most computer security mechanisms even have the same names as their physical counterparts. The basic concepts in data security are:

2.1 Security

Information systems security is the ability to provide the services required by the user community while simultaneously preventing unauthorized use of system resources. Providing the system resources to those who need them is just as much a part of system security as protection and prevention of undesired use of those resources.

2.2 Confidentiality

- The concept of *Confidentiality* in information security pertains to the protection of information and prevention of unauthorized access or disclosure.
- The ability to keep data confidential, or secret, is critical to staying competitive in today's business environments
- Loss of confidentiality jeopardizes system and corporate integrity.

Threats to confidentiality

- Hackers , A hacker is an individual who is skilled at bypassing controls and accessing data or information that he or she has not been given authorization to do so.
- Masqueraders , Authorized users on the system that have obtained another persons credentials.
- Unauthorized Users, Users that gain access to the system even if "company rules" forbid it.
- Unprotected Downloads , Downloads of files from secure environments to non-secure environments or media.
- Malware , Virus and worms and other malicious software
- Software hooks (Trapdoors) , During the development phase software developers create "hooks" that allow them to bypass authentication processes and access the internal workings of the program. When the product development phase is over developers do not always remember the hooks and may leave them in place to be exploited by hackers.

2.3 Integrity

- Integrity deals with prevention of unauthorized modification of intentional or accidental modification.
- This concept further breaks down into authenticity, accountability, and non-repudiation.
  - *Authenticity* means that the information is from whomever we expect it to be and whatever we expect it to be.
  - *Accountability* means that the information has an owner or custodian who will stand by its contents.
  - *Non-repudiation* is a property achieved through cryptographic methods which prevents an individual or entity from denying having performed a particular action related to data

2.4 Availability

- Availability assures that the resources that need to be accessed are accessible to authorized parties in the ways they are needed. Availability is a natural result of the other two concepts.
- If the confidentiality and integrity of the systems are assured their availability for the purpose they are intended for is a direct consequence.
- Threats to Availability
  - Availability can be affected by a number of events which break down into human and non human influenced factors. These further break down to unintentional and intentional acts.

- Examples of unintentional (non-directed) acts can be overwriting, in part or whole, of data, compromising of systems, or network infrastructure by organizational staff.
- Intentional acts can be conventional warfare (bombs and air-strikes), information warfare *denial of service (DoS)* and *distributed denial of service (DDoS)*.
- Non-human factors include loss of availability due to fires, floods, earthquakes and storms.

2.5 Authentication

- Authentication is the process by which the information system assures that you are who you say you are; how you prove your identity is authentic.
- Methods of performing authentication are:
  - user ID and passwords. The system compares the given password with a stored password. If the two passwords match then the user is authentic.
  - Swipe card, which has a magnetic strip embedded, which would already contain your details, so that no physical data entry takes place or just a PIN is entered.
  - digital certificate, an encrypted piece of data which contains information about its owner, creator, generation and expiration dates, and other data to uniquely identify a user.

  - key fob, small electronic devices which generate a new random password synchronized to the main computer
  - Biometrics - retinal scanners and fingerprint readers. Parts of the body are considered unique enough to allow authentication to computer systems based one their properties.
- For a very secure environment, it is also possible to combine several of these options, such as by having fingerprint identification along with user ID and key fob.

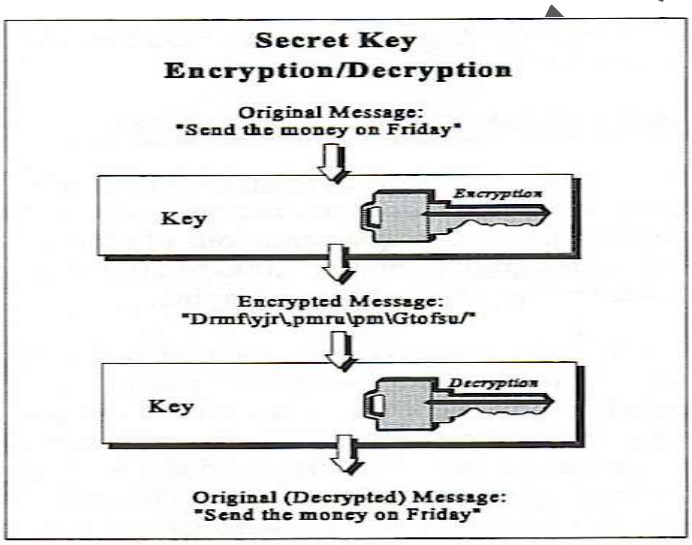
2.6 Non-repudiation

- Data flows around the internet at the speed of light, or as close to it as the servers allow. There are hackers, spoofers, sniffers, and worse out there just waiting to steal, alter, and corrupt your information.
- Data consumers need to be able to trust that the data has not been altered, and that its source is authentic.
- Through the use of security related mechanisms, producers and consumers of data can be assured that the data remains trustworthy across untrusted networks such as the internet, and even internal intranets.

2.7 Assuring data validity

- The identity of the data producer can be assured if the data is *signed* by its source.
- Data is signed through its encryption using a shared secret such as a numerical crypto key or using a “public/private” key pair

- The consumer of the data can validate the signature of the data and thereby be assured that the data has remained unaltered in transmission.
- Since the data could be decrypted into something intelligible, the content is valid.



2.8 Authorization

- Authorization is the granting or denial of resource access to a user.
- It is dependent on the access rights to a resource existing on the system.
- Identification and authorization work together to implement the concepts of Confidentiality, Integrity, and Availability.

- *Confidentiality* - A user's identity is authenticated by the system. That user is subsequently represented in the system by a token - either character or numerical data. By using this token, access to data and resources can be allowed or denied.
- *Integrity* - Authorization provides the mechanism to prevent the disruption of data by known users with out the appropriate authority.
- *Availability* - the ability to touch resources that you are permitted to touch, is backed by the ability to authorize users to resources.

3. Security mechanisms:

3.1Trust

All computer security springs from the concept of inherent or original trust. Just as a child inherently trusts its parents, a secure computer system inherently trusts those who set it up. While this may seem rather obvious, it is an important concept because it is the origination of all subsequent security measures. There's more inherent trust in computer security than simply the original establishment of a system. For example, you trust that there are no "back doors" in the software you use that could be exploited by a knowledgeable person to gain access. You trust that the login screen that you are looking at is actually the system's true login screen and not a mimic designed to collect your password and then pass it to a remote system. Finally, you trust that the designers of the system have not made any serious mistakes that could obviate your security measures.

3.2 Authentication

Authentication **is** the process of determining the identification of a user. *Authentication* is the process of determining the identity of a user. Forcing the user to prove that they know a secret that should be known only to them proves that they are who they say they are.

1- **user account:** A record containing information that identifies a user, including a secret password. *User accounts* are associated with some form of secret, such as a password, PIN, biometric hash, or a device like a *smart card* that contains a larger, more secure password than a human could remember. To the system, there is no concept of a human; there is only secret, information tied to that secret, and information to which that secret has access.

2- **Smart card:** An electronic device containing a simple calculator preprogrammed with a code that cannot be retrieved. When given a challenge, it can calculate a response that proves it knows the code without revealing what the code is.

Authentication is only useful in so far as it is accurate. Passwords are probably the least reliable form of authentication in common use today, but they're also the most easily implemented—they require no special hardware and no sophisticated algorithms for basic use. However, they are easily guessed, and even when they're carefully chosen it's still possible to simply guess the entire range of possible passwords on many systems in short order. A less common but more secure method of authentication is to physically possess a unique key. This is analogous to most physical locks. In computer security systems, "keys" are actually large numbers generated by special algorithms that incorporate

information about the user and are stored on removable media like smart cards. The problem with keys is that, like physical keys, they can be lost or stolen. However, when combined with a password, they are very secure and difficult to thwart. Another form of authentication provides inherent identification by using a physical property of the user. This is called biometric authentication, and it relies upon unique and unchangeable physical properties of a human, such as handwriting characteristics, fingerprints, facial characteristics, and so forth. Biometric authentication has the potential to be the most reliable form of authentication because it's easy to use, nearly impossible to fake when correctly implemented, and can't be circumvented for the sake of convenience. Some forms of biometric authentication are easier to "forge" than others, and naïve implementations can sometimes be easily faked. But when well implemented, biometric authentication is the most secure form of authentication and the only form that can be truly said to uniquely and unmistakably identify a user.

3.3 Chain of Authority

Trust provider A trusted third party that certifies the identity of all parties in a secure transaction. Trust providers do this by verifying the identity of each party and generating digital certificates that can be used to determine that identity. Trust providers perform a function analogous to a notary public. During the installation of a security system, the original administrator will create the root account. From the root account (called the "administrator" account in Windows and the "Supervisor" account in NetWare), all other accounts, keys, and certificates spring. Every account on a system, even massive systems containing millions of accounts, spring from this chain of authority.

The concept of chains of authority isn't often discussed because it is inherent in a secure system. Certificate systems are also based on a chain of authority. Consider the case of separate businesses that do a lot of work together. It would be convenient if users from Business Alpha could automatically log on to computers at Business Beta. But because these two systems have two different chains of authority, there's no way for Business Alpha to trust that users who say they are from Business Beta actually are. This problem is solved by having both businesses trust a third-party *trust provider,* or a company that specializes in verifying identity and creating secure certificates that can be used to prove identity to foreign systems. As long as both businesses trust the same trust provider, they are rooted in the same chain of authority and can trust certificates that are generated by that trust provider. Trust providers are the digital equivalent of a notary public. Examples of trust providers
are VeriSign and Thawed.

3.4 Accountability

Accountability is where the secret meets the user. Users don't try to circumvent security because their identity would be known and they would be held legally accountable for their actions. It is accountability, rather than access controls, that prevents illegal behavior. In pure accountability-based systems, no access control mechanisms are present. Users simply know that their every action is being logged, and since their identity is known and their activities are tracked, they won't do things that could jeopardize their position (unless something happens to make them no longer care). The problem with accountability-based systems is twofold—they only work if identity can't be

faked, and there are rare occasions where users lose their inhibitions. Without access control, these users can destroy the entire system. For these reasons, accountability-based security is normally used to augment access control systems rather than to replace them.

4 Access Control

Access control is the security methodology that allows access to information based on identity. Users who have been given permission or keys to information can access it—otherwise, access is denied.

.1 Permissions-Based Access Control

File :A sequence of related information referenced by a filename in a directory.

Once the system knows the identity of an individual because they've been authenticated, the system can selectively allow or deny access to resources like stored files based on that identity. This is called permissions-based security because users are either granted or denied permission to access a *file* or other resource. The question of who has access to which files is typically either defined by administrators when the system is implemented or created according to some set of default rules programmed into the system; for instance, the original creator (owner) of a file is the only user who can change it. Access controls are typically implemented either as directory permissions that apply to all files within the directory or by an access control list, which is a component of a file that explicitly lists which users can access it. Typically, when a

Encryption-Based Access Control (Privacy)

private key : The key used to decode public key messages that must be kept private.

A totally different way to control access is to simply encrypt data using public key encryption. Access to the encrypted data is given to those who want it, but it's worthless to them unless they have the

private key required to decode it. Using PKE to secure data works very well, but it requires considerably more processing power to encode and decode data.

References:

1- Managing Cisco network security: building rock-solid networks 2000
2- Cryptography and Network Security, William Stalling , 2003
3- Asst.prof.Dr.Soukaena.H,"Ch-1,Lec-1," data security" computer secines department , university of technology.

Chapter 2

2.1 Understanding Hacking

Know thy enemy. Hackers are the reason you need to implement computer

security, and an in-depth defense against any adversary requires an in-depth understanding of that adversary. This chapter describes hackers,their motivations, and their methods. By knowing a hacker's motivations, you can predict your own risk level and adapt your specific defenses to ward off the type of hackers you expect to attack your network while retaining as much usability as possible for your legitimate users.

2.2 What Is Hacking?

Hacking is quite simply the attempt to gain access to a computer system without authorization. Originally, the term *hacker* simply referred to an adept computer user, and gurus still use the term to refer to themselves in that original sense. But when breaking into computer systems (technically known as *cracking*) became popular, the media used the *hacker* to refer only to computer criminals, thus popularizing only the negative connotation. In this book, we refer only to that negative connotation as well. Hacking is illegal. Title 18, United States Code, Section 1030, first enacted by Congress in 1984, criminalized hacking. Technically, the code requires that the perpetrator actually "do" something other than simply obtain access and read information—but then, if that's all they did, you probably wouldn't know you'd been hacked anyway. The law specifically states that the perpetrator must "knowingly" commit the crime—thereby requiring that at least some sort of notification that unauthorized access is illegal be posted or that some authentication hurdle be established in order to make the activity prosecutable. According to the FBI, for a computer-related crime to become a federal crime, the attacker must be shown to have caused at least $5,000 worth of damage. This is why spammers who access open relay mail

servers get away with transmitting their floods of e-mail through other people's mail servers without being prosecuted— they're not doing enough financial damage to any one victim to really be prosecutable, and the SMTP servers are not performing authentication so there's no reasonable expectation of security. But, because spam has become such a plague lately, the 2004 CANSPAM Act specifically criminalizes the transmission of unsolicited commercial e-mail without an existing business relationship.

2.3 Types of Hackers

Learning to hack takes an enormous amount of time, as does perpetrating actual acts of hacking. Because of the time it takes, there are only two serious types of hackers: the underemployed and those hackers being paid by someone to hack. The word *hacker* conjures up images of skinny teenage boys aglow in the phosphor of their monitors. Indeed, this group makes up the largest portion of the teeming millions of hackers, but they are far from the most serious threat.Hackers fall quite specifically into these categories, in order of increasing threat:

1-Security experts
2-Script kiddies
3-Underemployed adults
4-Ideological hackers
5-Criminal hackers
6-Corporate spies
7-Disgruntled employees

- **Security Experts**

Most security experts are capable of hacking but decline to do so for moral or economic reasons. Computer security experts have found that there's more money in preventing hacking than in perpetrating it, so they spend their time keeping up with the hacking community and current techniques in order to make themselves more effective in the fight against it. A number of larger Internet service companies employ ethical hackers to test their security systems and those of their large customers, and hundreds of former hackers now consult independently as security experts to medium-sized businesses. These experts often are the first to find new hacking exploits, and they often write software to test or exacerbate a condition. Practicing hackers can exploit this software just as they can exploit any other software.

- **Script Kiddies**

script kiddie A novice hacker.*Script kiddies* are students who hack and are currently enrolled in some scholastic endeavor—junior high, high school, or college. Their parents support them, and if they have a job, it's only part-time. They are usually enrolled in whatever computer-related courses are available, if only to have access to the computer lab. These hackers may use their own computers, or (especially at colleges) they may use the more powerful resources of the school to perpetrate their hacks. Script kiddies joyride through cyberspace looking for targets of opportunity and are concerned mostly with impressing

their peers and not getting caught. They usually are not motivated to harm you, and in most instances, you'll never know they were there unless you have software that detects unusual activity and notifies you or a firewall that logs attacks—or unless they make a mistake. These hackers constitute about 90 percent of the total manual hacking activity on the Internet. If you consider the hacking community as an economic endeavor, these hackers are the consumers. They use the tools produced by others, stand in awe of the hacking feats of others, and generally produce a fan base to whom more serious script kiddies and underemployed adult hackers play. Any serious attempt at security will keep these hackers at bay. In addition to the desire to impress their peers, script kiddies hack primarily to get free stuff: software and music, mostly. They share pirated software amongst themselves, make MP3 compressed audio tracks from CDs of their favorite music, and trade the serial numbers needed to unlock the full functionality of demo software that can be downloaded from the Internet.

- **Vectors That Hackers Exploit**

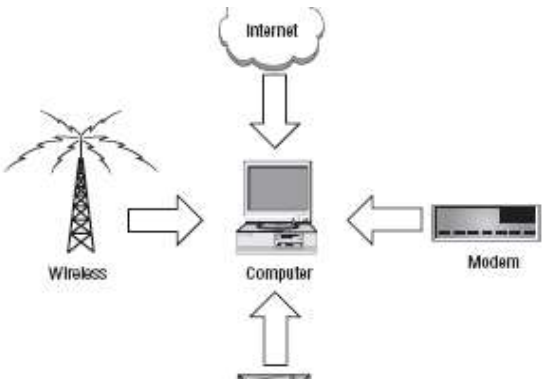
There are only four ways for a hacker to access your network:

1-By connecting over the Internet

2-By using a computer on your network directly

3-By dialing in via a Remote Access Service (RAS) server

4-By connecting via a nonsecure wireless network



- **Direct Intrusion**

Hackers are notoriously nonchalant and have, on numerous occasions, simply walked into businesses, sat down at a local terminal or network client, and begun setting the stage for further remote penetration. In large companies, there's no way to know everyone by sight, so an unfamiliar worker in the IT department isn't uncommon or suspicious at all. In companies that don't have ID badges or security guards, it isn't anybody's job to check credentials, so penetration is relatively easy. And even in small companies, it's easy to put on a pair of coveralls and pretend to be with a telephone or network wiring company or even pose as the spouse of a fictitious employee. With a simple excuse like telephone problems in the area, access to the server room is granted (oddly,these are nearly always colocated with telephone equipment). If left unattended,a hacker can simply create a new administrative user account. In less than a minute, a small external modem or wireless access point can be attached without even rebooting your server. Solving the direct intrusion problem is easy: Employ strong physical security at your premises and treat any cable or connection that leaves the building as a security concern. This means putting

firewalls between your WAN links and your internal network or behind wireless links. By employing your firewalls to monitor any connections that leave the building, you are able to eliminate direct intrusion as a vector.

- **Dial-Up**

Dial-up hacking, via modems, used to be the only sort of hacking that existed, but it has quickly fallen to second place after Internet intrusions. (Hacking over the Internet is simply easier and more interesting for hackers.) This doesn't mean that the dial-up vector has gone away—hackers with a specific target will employ any available means to gain access.

Although the dial-up problem usually means exploiting a modem attached to a Remote Access Service (RAS) server, it also includes the problem of dialing into individual computers. Any modem that has been set to answer for the purpose of allowing remote access or remote control for the employee who uses the computer presents a security concern. Many organizations allow employees to remotely access their computers from home using this method. Containing the dial-up problem is conceptually easy: Put your RAS servers outside your firewall in the public security zone, and force legitimate users to authenticate with your firewall first to gain access to private network resources. Allow no device to answer a telephone line behind your firewall. This eliminates dial-up as a vector by forcing it to work like any other Internet connection.

- **Internet**

Internet intrusion is the most available, most easily exploited, and most problematic vector of intrusion into your network. This vector is the primary topic of this book. If you follow the advice in this section, the Internet will be

the only true vector into your network. You already know that the Internet vector is solved by using firewalls, so there's no point in belaboring the topic here. The remainder of this book is about solving the Internet intrusion vector.

2.4 Hacking Techniques

Hacking attacks progress in a series of stages, using various tools and techniques. A hacking session consists of the following stages:

1-Target selection

2-Information gathering

3-Attack

The hacker will attempt to find out more about your network through eachsuccessive attack, so these stages actually feed back into the process as more information is gathered from failed attacks.

Chapter 3

Firewall

3.1 Introduction

Firewall process is the same as process of farmer who is attempting to protect his hen house from the fox. The lay of land places the hen house in the center of the barnyard, which is surrounded by a fence to keep the animals in place. Another fence is in place at the perimeter around the farmer's property. Keep in mind where the chickens are housed. The farmer places a chair at the front gate on the outer perimeter, and sits down to watch for the fox. Now he is in position to act the same as firewall would act in a network environment. Both are protecting
the main point of entry into the system .

3.2 Firewall Definition

A firewall protects networked computers from intentional hostile intrusion that could compromise confidentiality or results in data corruption or denial of service]. It may be a hardware device as shown in figure (2-1), or a software program running on secure host computer as shown in figure (2-2).

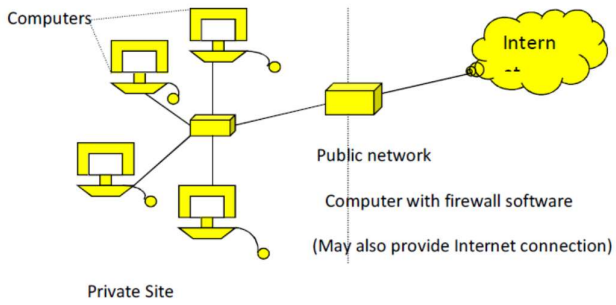


Figure (2-2) Computer with Firewall software.

A firewall is placed between the site and the rest of the Internet; as shown in figure (2-3).

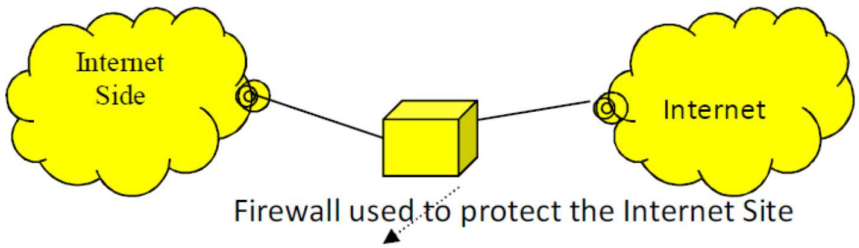


Figure (2-3) Position of the firewall used to protect the Internet site.

The term firewall is derived from the fireproof physical boundary placed between two structures to prevent fire from moving between them .If the site has multiple Internet connections,a firewall must be placed at each, and all the site's firewalls must be configured to enforce the site's security policy.

3.3 Firewall Concept, Conditions, Security and Transparency:

Firewall Concept

All traffic from inside to outside, and vice versa through the main entry, must pass through the firewall. This is achieved by physically blocking all access to the site except via the firewall. Various configurations are possible. Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies. In general Firewall is one of the famous types of intruder detection. Firewall effective against worms because it is much difficult for a worm to authenticate itself to a firewall operating a tight exclusive policy than to a general, widely used system. It is effective against Trojan horse . However, even firewall are not invincible.

Firewall Conditions:

There are some conditions must be considered in firewalls the most important conditions are declared in the following points:

1- The firewall must be ease to use, has powerful graphical user Interfaces (GUI), which simplifies the job of installation,configuration, and management.

2- The firewall must has high performance, should be fast enough so users do not feel the screening of packets. The volume of data throughput and transmission speed associated with the product should be consistent with the company's
bandwidth to the Internet.

3- The firewall must be flexible , should be open enough to accommodate the security policy of the company, as well as to allow for changes in the features. Remember that a security policy should a very seldom change, but security procedures should always be reviewed, especially in light of new Internet
and web centric application.

Firewall Security:

The firewall itself is immune to penetration. This implies the use of a trusted system with a secure operating system.

Firewall Transparency:

Transparent firewall, this is important because if a confusing system is adopted, users will develop resistance to it and end up not using it. Conversely, the more transparent the firewall is to users, the more likely they will use it appropriately.

3.4 Firewall Limitation

Security of firewalls neither provides perfect security nor is free of operational difficulties. The most important limitations of the firewall are interpreted in the following points:

1. Firewalls do not protect against malicious insiders.

2. Firewalls have no protection against connections that circumvent the firewall (i.e., Modems) attached to computers inside the firewall.

3. Firewalls can not protect against the transfer of virusinfected programs or files. Because of the variety of operating systems and applications supported inside the perimeter, it would be impractical and perhaps impossible for the firewall to scan all incoming files, E-mail, and message for viruses .

4. Firewalls can not protect against completely new threats, firewalls designed to protect against known threats. Nofirewall can automatically defend against every new threat that arises .

3.5 Firewalls types

There are many types of firewalls, they tend to differ in their approach but can be characterized as firewalls, which block traffic, and firewall which permit traffic. The common types of the
firewalls are:

a- Packet Filtering Firewall.

b- Application Level Firewall.

c- Circuit Level Firewall.

d- Stateful Multilayer Inspection Firewall.

e- Firewall Reference Model.

References:

1- Managing Cisco network security: building rock-solid networks 2000
2- Cryptography and Network Security, William Stalling , 2003.

Chapter Four

Cryptography Algorithms

4.1 Introduction

Suppose a sender wants to send a message to a receiver. Moreover, this sender wants to send the message securely: she wants to make sure an eavesdropper can not read the message.

A message is *plaintext* [sometime called clear text]. The process of disguising a message in such a way as to hide its substance is *encryption*. An encrypted message is *cipher text*. The process of truing cipher text back into plaintext is *decryption*; this is all shown in figure (1).

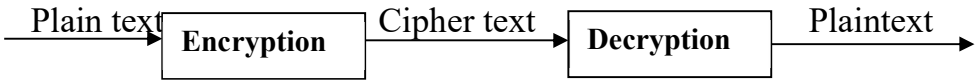


Figure (1): Encryption and Decryption

Plain text is denoted by M, for message or P for plaintext. It can be a stream of bits, a text file, a bitmap, a stream of digitized voice, a digital video image. Whatever. As far as a computer is concerned, M is simply binary data.

Cipher text is denoted by C. it is also binary data: some the same size as M. some time larger (by combining encryption with compression. C may be smaller than M.)

The encryption functions E operate on M to produce C, or, in mathematical notation.

$$E(M) = C$$

In the reverse process, the decryption functions D operate on C to produce M:

$$D(C) = M$$

Since the whole point of encryption and then decryption a message is to recover the original plain text, the following identity must hold true.

$$D(E(M)) = M$$

4.2 The components of the cryptographic system. (Cryptosystem)

1) A plain text message space , M.

2) A cipher text message space, C.

3) A key space, K.

4) A family of enciphering transformations, E_k: m → c, where k€ K.

5) A family of deciphering transformations, D_k: c → m, where k€ K.

Example//

$$D_k(E_k(m))=m$$

4.3Cryptosystem must satisfy three general requirements:

1) The enciphering and deciphering transformations must be efficient for all keys.

2) The system must be easy to use.

3) The security of the system should depend only on the secrecy of the keys and not on the secrecy of the algorithms E or D.

4.4 Types of attacks

There are three general types of cryptanalytic attack. Of course, each of them assumes that the cryptanalyst has complete knowledge of the encryption algorithm used.

1- Cipher text – only attack: the cryptanalyst has the cipher text of several message , all of which have been encrypted using the same encryption algorithm, the cryptanalyst job is to recover the plain text of many message as possible, or better yet to deduce the key (or keys) used to encrypted the message. In order to decrypted other message encrypted with the same keys.

Given: $C_1=E_k(p_1)$, $C_2=E_k(p_2)$,….. $C_i=E_k(p_i)$

Deduce: either p_1, p_2, p_i, k; or an algorithm.

To infer p_{i+1} from $C_i=E_k(p_{i+1})$

2- Known – plain text attack: the cryptanalyst has access not only to the cipher text of several message, but also to the plain text of those message. His job is to deduce the key (or keys) used to encrypt the message or an algorithm to decrypt any new message encrypted with the same key, (or keys).

Given: $P_1,C_1=E_k(p_1)$, $P_2,C_2=E_k(p_2)$,….. $P_i,C_i=E_k(p_i)$

Deduce: either k or an algorithm.

To infer P_{i+1} from $C_i=E_k(P_{i+1})$

3- chosen– plain text attack: the cryptanalyst not only has access to the cipher text and associated plain text for several message. But be also chooses the plain text that gets encrypted.

This is more powerful than a known-plain text attack, because the cryptanalyst can choose specified plaintext blocks to encrypted, ones that might more information about the key. His job is to deduce the key (or keys) used to encrypt the message or an algorithm to decrypt any new message encrypted with the same key (or keys).

Given: $P_1, C_1=E_k(p_1)$, $P_2, C_2=E_k(p_2)$,…. $P_i, C_i=E_k(p_i)$

where the cryptanalyst gets to choose p_1, p_2, p_i

Deduce: either k or an algorithm.

To infer P_{i+1} from $C_i=E_k(P_{i+1})$

Other

4- chosen– cipher text attack.

5- Adaptive-chosen-plaintext attack: this is a special case of a chosen-plaintext attack.

Note only can the cryptanalyst choose the plaintext that is encrypted , but he can also modify his choice based on the result of pervious encryption- in a

chosen-plain text attack,a cryptan-alyst might just be able to choose one large block of plain text to be encrypted ; in an adaptive-chosen-plaintext attack he can choose a smaller block of plaintext and then choose another based on the result of the first, and so forth.

4.5 Traditional Systems Cipher

Traditional systems related to all ciphers used before and they are divided in two:

1) Transposition.
2) Substitution.

1) Transposition: means a management of letters according to some schema. i.e. rearrange bits or characters in the data.

Plaintext —written→ figure —take off→ cipher text

((Transposition))

a) The plaintext was written into the figure according to some (written) path.
b) The cipher text was taken off the figure according to some (take off) path.
c) The key consisted of the figure together with the written in & take off path.

4.5.1 Transposition Types

1) Simple transposition (columner transposition):

- The plaintext can be written into a matrix by rows the cipher text can obtain by taking the columns in some order.

Example 1 :

CRYPTOGRAPHY = plain text
3 1 4 2 = key

(Encipherment process)

3	1	4	2					
C	R	Y	P					
T	O	G	R	+ key	➔	ROP	PRY	CTA YGH
A	P	H	Y			1	2	3 4

(Decipherment process)

key = 3 1 4 2
cipher text = ROP PRY CTA YGH
1 2 3 4

3	1	4	2		
C	R	Y	P		
T	O	G	R	➔	CRYPTOGRAPHY
A	P	H	Y		

Example 2:

Plaintext = this is transposition
Key = code

A B 1C D2 E3 F G H I J K L M N O4 P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Code=1 4 2 3

(Encipherment process)

1	4	2	3
t	h	i	s
i	s	t	r
a	n	s	p
o	s	i	t
i	o	n	*x*

cipher text = tiaoi itsin srptx hsnso

(Decipherment process)

cipher text = tiaoi itsin srptx hsnso
1 2 3 4

key= code = 1 4 2 3

1	4	2	3
t	h	i	s
i	s	t	r
a	n	s	p
o	s	i	t
i	o	n	*X*

→

= this is transposition

2) *A fixed period d with a permutation function.* F: Zd➔Zd

Example1 :

Plain text: CRYPTOGRAPHY , d=4 , f=2 4 1 3

(Encipherment process)

d=4 d=4 d=4
CRYP TOGR APHY
1 2 3 4 1 2 3 4 1 2 3 4

Cipher text :RPCY ORTG PYAH

(Decipherment process)

Cipher text :RPCY ORTG PYAH , d=4, f= 2413

2 4 1 3 2413 2413

Plaintext : CRYP TOGR APHY

1 2 3 4 1 2 3 4 1 2 3 4

Note: d represent block length

Problems

Q1/

Encipher the following plain text="work hard for future" using the key =513462 based on simple transposition method (columnar transposition) and test the result.

Sol : Encryption Process

Key= 5 1 3 4 6 2

w o r k h a
r d f o r f ----------→123456
u t u r e x

C=odt afx rfu kor wru hre

Test by using decryption process

Decryption process:

C=odt afx rfu kor wru hre
1 2 3 4 5 6

Key= 5 1 3 4 6 2

w o r k h a
r d f o r f ----------→work hard for future
u t u r e x

Q2/

Encipher the following plain text= "work hard for future", using the key =463125,d=6 based on transposition method (fixed period d with a permutation function) and test the result.

Sol: Encipher process

d1=6	d2=6	d3=6
w o r k h a	r d f o r f	u t u r e x
1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6

Key=463125→
C= karwoh offrdr rxuute

Test by using decryption process→

C= karwoh o f f r d r r x u u t e
Key= 463125 4 6 3 1 2 5 4 6 3 1 2 5 → 123456
Plaintext = work hard for future x

Sol: *work hard for future*

Q3/

Decipher the following cipher text ="axtodfkrurfoheuwrr", when you know two encryption transposition methods are used first method is (columnar transposition) where key1=513462 and the second method is(fixed period d with a permutation function) where key2=463125, d=6

Sol:

Firstly: we decipher the ciphertext ="axtodfkrurfoheuwrr" based on fixed period d with a permutation function) where key2=463125, d=6

	d1=6	d2=6	d3=6
C=	a x t o d f	k r u r f o	h e u w r r
K=	463125	463125	463125

C1=odtafx rfukor wruhre

Secondly": we decipher the ciphertext C1="odtafx rfukor wruhre" based on is (columnar transposition) where key1=513462.

C1= odt afx rfu kor wru hre
1 2 3 4 5 6

Key= 5 1 3 4 6 2
w o r k h a
r d f o r f ----------→work hard for futurex
u t u r e x

Sol: *work hard for future*

Mathematical Background

- ***Number theory***

(1) **Modular Arithmetic**

a=b mod n
a=b+k*n
where k is integer,
b= a/n or a-b/n without reminder,
a is integer number (positive)
And we can write **a≡ b mod n**

Example :

1) 23 mod 12=11
23= 11*1+12 ≈ a=b+k*n
23-11=1*12 ≈ a-b=k*n
23-11/12=12/12=1

2) 17 mod 5=2
17=2+3*5 ≈ a=b+k*n
17-2=3*5 ≈ a-b=k*n
17-2/5=15/5=3

(2) **Greatest Common Divisor (GCD)**

Two numbers are relatively prime when they share no factors in common other than 1. In other words, if the greatest common divisor of a and n is equal to 1. This is written

Gcd(a, n) = 1

Example:

1- gcd(27,18)
27= 18*1 +9
18= 9*2+0
Gcd(27,18)=9

2- gcd(5,3)
5= 3*1+2
3=2*1+1
2=1*1+1
1=1*1+0
Gcd(5,3)=1

3- gcd(123,4567)
4567=123*37+16
123=16*7+11
16=11*1+5
11=5*2+1
5=1*5+0
gcd(123,4567)=1

(4) **Computing Inverse**

Example:

(1) 5^{-1}(mod 23)

23 mod 5 =3
5 mod 3 =2
3 mod 2 =1
2 mod 1=0

a mod n=b
b=a-k.n
3=23-4.5
2=5-1.3
=5-1.(23-4.5)
=5-23+4.5
=5.5-23
1=3-1.2
=(23-4.5)-1.(5.5-2)
=23-4.5-5.5+23
=2.23-9.5

x=23-9=14

x=14

14*5 mod 23=1

14=x inverse 5 mod 23

(2) $10^{-1} \bmod 26$

26 mod 10 =6 6= 26-2.10

10 mod 6 =4 4= 10-1.6

=10-1.(26-2.10)

=10-26+2.10

=3.10-26

6 mod 4 =2 2= 6-1.4

=(26-2.10)-1.(3.10-26)

=26-2.10-3.10+26

= 2.26-5.10

4 mod 2=0

x=26-5=21

10*21 mod 26=

210 mod 26≠1

Inverse not exist

University of Technology
Computer Sciences Department
Network Management
Branch

Substitution Ciphers

4.5. 2 Substitution Ciphers

Four types:

a) simple
b) homophonic
c) polyalphabetic
d) polygram

a) Simple Substitution

- replace each plaintext letter with a corresponding cipher text letter.

1- Keyword

Example1: Encipher the following plaintext= "CRYPTOGRAPHY", using Keyword= CRYPTOGRAPHIC SYSTEM

Sol:

P: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

K: C R Y P T O G A H I S E M B D F J K L N Q U V W X Z

To encipher the plaintext= CRYPTOGRAPHY
ciphertext=YKXFNDGKCFAX

Example2: Decipher the ciphertext="sj oi jq fjs sj oi " using keyword="good mind good health"

Sol:

plain A B C D E F g H I J K L M N O P Q R S T U V W X Y Z

key	g	o	d	m	i	n	h	e	a	l	t	b	c	f	j	k	p	q	r	s	u	v	w	x	y	z

Ciphertext= sj oi jq fjs sj oi
Plaintext= TO BE OR NOT TO BE

2- Shift alphabets

F(a)=(a+k)mod n encipher

M=(c-k)mod n decipher

i.e. c=(p+k)mod 26, where a =0,……..z=25

- this is called shifted alphabet by k position
- example (caeser cipher use k=3)

P= a b c d…….z

C= d e f g……c

Example: Encipher the following plaintext= " This day is sunny day" using Shift alphabets where key =5.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

C=(p+k)mod 26 Encryption Process

T=19→ (19+5) mod 26=24=y
H=7 → (7+5) mod 26=12=m
I=8 → (8+5) mod 26=13=n
S=18→ (18+5) mod =23=x
D=3 → (3+5) mod 26=8=i
A=0 → (0+5) mod 26=5=f
Y =24→ (24+5) mod 26=3=d
U=20→ (20+5) mod 26=25=z
N=13→ (13+5) mod 26=18=s

Ciphertext= ymos ifd nx xzssd ifd

P=(c-k) mod 26 Decryption Process

Y=24→ (24-5) mod 26= 19 =T
M=12→ (12-5) mod 26=7=H
13=14→ (13-5) mod 26=8=I
X=23→ (23-5) mod26=18=S
I=8→ (8-5) mod26=3=D
F=5→ (5-5) mod 25=0=A
D=3→ (3-5)mod 26= -2 mod26→ -2+26=24→Y
Z=25→ (25-5) mod 26 =20→U
S=18 → (18-5) mod 26=13→N

Plain text= This Day is Sunny Day

3- Multiplication

F(a)=(a.k)mod n

Or

C=(P.k)mod 26 , where GCD(k,26)=1 …. Encryption process

P=(c.k-1) mod 26 , where GCD(k,26)=1…. Decryption Process

Example1: Encryption Process

K=9

P: A B C D E F G H I J K ……..Z

C: A J S B K T C L U D……..R

Example2: Decrypt the following ciphertext="fq dmqz nqf fq dm" when you know that simple substitution (multiplication) method is used and key=3 .

Sol: Decryption process → $P=(c.k^{-1}) \bmod 26$, where GCD(k,26)=1

1-Find Inv(3) mod 26 →

3^{-1}(mod 26)

26 mod 3 =2
3 mod 2 =1

a mod n=b
b=a-k.n
2=26-8.3 ----------(1)
1=3-1.2
=3-1.(26-8.3)
=3-1.26+8.3
=1.3-1.26+8.3
1=9.3-26

2 mod 1=0 ---End

Sol→ $3^{-1}=9$

2-

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

f=5 → (5*9) mod 26= 45 mod 26=19→T
q=16→ (16*9)mod26=144 mod 26=14→o
d=3→ (3*9) mod 26=1→b
m=12→(12*9)mod 26=108 mod 26 =4→e
z=25→(25*9)mod 26= 255 mod 26=17→r
n=13→(13*9)mod 26=177 mod 26 =13→n

Plaintext = To be or not to be

4- Affine Transformation (shift + multiplication)

Like : ((1)) keywords as above

((2)) multiplication

C= (PK) mode 26 when gcd(k,26)=1

C=(p+k) mod 26

Encipher process ;

C=(PK1+K0) mod 26 where k0 is additive key,

and k1 is multiplication key and gcd(k1,26)=1

Decipher process:

P= ((C-K0)* inv(k1)) mod 26

Example : Encrypt the following plaintext=" work hard " using affine transformation method where the additive key=5 and multiplication key=3, and test the result .

Sol:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Encryption process:

W=22→((22*3)+5) mod 26=71 mod 26=19→t
O=14→((14*3)+5) mod 26=47 mod 26=21→v
R=17→((17*3)+5) mod 26=56 mod 26=4→e
K=10→((10*3)+5) mod 26=35 mod 26=9→j
H=7→((7*3)+5) mod 26=26 mod 26=0→a

A=0→((0*3)+5) mod 26=5 mod 26=5→f
D=3→((3*3)+5) mod 26=14 mod 26=14→o

Ciphertext= tvejafeo

→ *Test the results using decryption process*

Decipher process:

Sol:

1- Find Inv (3) mod 26 →

3^{-1}(mod 26)

26 mod 3 =2
3 mod 2 =1

a mod n=b
b=a-k.n
2=26-8.3 ----------(1)
1=3-1.2
=3-1.(26-8.3)
=3-1.26+8.3
=1.3-1.26+8.3
1=9.3-26

2 mod 1=0 ---End

Sol→ 3^{-1}=9

2-

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Ciphertext= tvejafeo

Additive key (k0)=5, multiplication key(k1)=3 and 3^{-1}=9 (the inverse)

P= ((C-K0)* inv(k1)) mod 26 , Where gcd(k1,26)=1

t=19→((19-5)*9) mod 26=22→w
v=21→((21-5)*9) mod 26=14→o
e=4→((4-5)*9) mod 26=17→r
j=9→((9-5)*9) mod 26=10→k
a=0→((0-5)*9) mod 26=7→h
f=5→((5-5)*9) mod 26=0→a
D=3→((3*3)+5) mod 26=3→d

Plaintext= work hard

b) *Homophonic substitution ciphers:*

- a *homophonic substitution* cipher maps each plaintext letter into a set of ciphertext elements(called homophones)

Example:

A	B	C	D	. . .	M	N	. . .	Z
3	17	60	4	. . .	71	11	. . .	31
25	55	80			26	50		83
31	90							
36								
79								

- notice : no. of homophones assigned to each plain text letter on proportional to its frequency.

- *Beal ciphers*:
  I use the first letter of each word from a plain text (ex : a book)
  II number the words rx: 1 ,2,3,4,…, text length as words.
  III to encipher a plain text a letter use its no, from II.
- Example

  01 02 03 04 05 06 07
  - The plain text a book : when , in the course of human event ….
  - To encipher: THE = 03 06 07

- ***Higher Order Homophonic Algorithm***

1- matrix n ×n
2- fill the matrix with numbers 1 ➔ n^2
3- encipher the plaintext with a long with a dummy message x
M= m1, m2, ….
X= x1, x2,….
Ci= K[m1 xi], i= 1,2, ……

Example
Let n=5 , plain alphabet { E, I, L , M, S}

	E	I	L	M	S
E	10	22	18	02	11
I	12	01	25	05	20
L	19	06	23	13	07
M	03	16	08	24	15
S	17	09	21	15	04

M= S M I L E
X= L I M E S (DUMMY MESSAGE)
C= 21 16 05 19 11

c- Polyalphabetic Ciphers

Another way to improve on the simple monoalphabetic technique is to use different monoalphabetic substitutions as one proceeds through the plaintext message. The general name for this approach is polyalphabetic substitution cipher. All these techniques have the following features in common:

1. A set of related monoalphabetic substitution rules is used.
2. A key determines which particular rule is chosen for a given transformation.

The best known, and one of the simplest, such algorithm is referred to as the Vigenère cipher. In this scheme, the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers, with shifts of 0 through 25. Each cipher is denoted by a key letter, which is the ciphertext letter that substitutes for the plaintext letter a. Thus, a Caesar cipher with a shift of 3 is denoted by the key value d.

To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating keyword. For example, if the keyword = deceptive, the message="we are discovered save yourself" is encrypted as follows:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	a																									z
B	b																									a
C	c																									b
D	d																									c
E	e																									d
F	f																									e
G	g																									f
H	h																									g
I	i																									h
J	j																									i
K	k																									j
L	l																									k
M	m																									l
N	n																									m
O	o																									n
P	p																									o
Q	q																									p
R	r																									q
S	s																									r
T	t																									s
U	u																									t
V	v																									u
W	w																									v
X	x																									w
Y	y																									x
Z	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

Vigener Table-1

Example1:

plaintext: wearediscoveredsaveyourself

key: deceptivedeceptivdeceptive

ciphertext: zicvtwqngrzgvtwavzhcqyglmgj

Decryption is equally simple. The key letter again identifies the row. The position of the ciphertext letter in that row determines the column, and the plaintext letter is at the top of that column. The strength of this cipher is that there are multiple ciphertext letters for each plaintext letter, one for each unique letter of the keyword. Thus, the letter frequency information is obscured. However, not all knowledge of the plaintext structure is lost.

To aid in understanding the scheme and to aid in its use, a matrix known as the Vigenère table is constructed (Table 1). Each of the 26 ciphers is laid out horizontally, with the key letter for each cipher to its left. A normal alphabet for the plaintext runs across the top. The process of encryption is simple: Given a key letter x and a plaintext letter y, the ciphertext letter is at the intersection of the row labeled x and the column labeled y; in this case the ciphertext is V.

Example2: Encipher the following plaintext= **" Good Luck To All"** , using Vigenère table , when you know **the keyword = yes** and test the results:

Sol: build Vigenère table using keyword= yes

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
y	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
e	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	y	v	w	x	y	z	a	b	c	d
s	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r

M= Good Luck To Al l
K= yesy esye sy esy

C=esgb pmao lm edj **Encrption process**

When Test the results using Vigenère table

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
y	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
e	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	y	v	w	x	y	z	a	b	c	d
s	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r

C=esgb pmao lm edj
K= yesy esye sy esy
M= Good Luck To Al l Decryption process

Polygram ciphers

1-Playfair Cipher

The best-known multiple-letter encryption cipher is the Playfair, which treats digrams in the plaintext as single units and translates these units into ciphertext

digrams. The Playfair algorithm is based on the use of a 5 x 5 matrix of letters constructed using a keyword. Here is an example, solved by Lord Peter Wimsey in Dorothy Sayers's Have His Carcase

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

In this case, the keyword is monarchy. The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetic order. The letters I and J count as one letter. Plaintext is encrypted two letters at a time, according to the following rules:

1. Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo on.

2. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ar is encrypted as RM.

3. Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, mu is encrypted as CM.

4. Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, hs becomes BP and ea becomes IM (or JM, as the encipherer wishes).

The Playfair cipher is a great advance over simple monoalphabetic ciphers. For one thing, whereas there are only 26 letters, there are 26 x 26 = 676 digrams, so that identification of individual digrams is more difficult. Furthermore, the relative frequencies of individual letters exhibit a much greater range than that of digrams, making frequency analysis much more difficult. For these reasons, the Playfair cipher was for a long time considered unbreakable. It was used as the standard field system by the British Army in World War I and still enjoyed

considerable use by the U.S. Army and other Allied forces during World War II.

Despite this level of confidence in its security, the Playfair cipher is relatively easy to break because it still leaves much of the structure of the plaintext language intact. A few hundred letters of ciphertext are generally sufficient.

2-Hill Cipher

This cipher is somewhat more difficult to understand than the others in this chapter, but it illustrates an important point about cryptanalysis that will be useful later on. This subsection can be skipped on a first reading.

Another interesting multiletter cipher is the Hill cipher, developed by the mathematician Lester Hill in 1929. The encryption algorithm takes m successive plaintext letters and substitutes for them m ciphertext letters. The substitution is determined by m linear equations in which each character is assigned a numerical value (a = 0, b = 1 ... z = 25). For m = 3, the system can be described as follow

$c_1 = (k_{11P1} + k_{12P2} + k_{13P3}) \bmod 26$

$c_2 = (k_{21P1} + k_{22P2} + k_{23P3}) \bmod 26$

$c_3 = (k_{31P1} + k_{32P2} + k3_{3P3}) \bmod 26$

This can be expressed in term of column vectors and matrices:

Or C = KP mod 26

where C and ***P are column vectors of length 3***, representing the plaintext and ciphertext, and K is a 3 x 3 matrix, representing the encryption key. Operations are performed mod 26.

For example, consider the plaintext "pay more money" and use the encryption key

The first three letters of the plaintext are represented by the vector the ciphertext for the entire plaintext is LNSHDLEWMTRW.

Decryption requires using the inverse of the matrix K. The inverse K^1 of a matrix K is defined by the equation $KK^1 = K^1K = I$, where I is the matrix that is all zeros except for ones along the main diagonal from upper left to lower right.

The inverse of a matrix does not always exist, but when it does, it satisfies the preceding equation. In this case, the inverse is:

This is demonstrated as follows:

It is easily seen that if the matrix K^1 is applied to the ciphertext, then the plaintext is recovered. To explain how the inverse of a matrix is determined, we make an exceedingly brief excursion into linear algebra. For any square matrix (m x m) the determinant equals the sum of all the products that can be formed by taking exactly one element from each row and exactly one element from each column, with certain of the product terms preceded by a minus sign. For a 2 x 2 matrix.

Example: Encipher the following plaintext= **" Hats"** using hill cipher method , when you know the **key** =$\begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}$, and test the results using decryption process**.**

Sol:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

M= hats → ha ts

ha=[7 0] , **ts**= [19 18]

key =$\begin{bmatrix} \mathbf{3} & \mathbf{3} \\ \mathbf{2} & \mathbf{5} \end{bmatrix}$

encryption process→ ha= $\begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} * \begin{bmatrix} 7 \\ 0 \end{bmatrix} = \begin{bmatrix} 21 \\ 14 \end{bmatrix} = \begin{bmatrix} v \\ o \end{bmatrix}$

Note: (3*7+3*0) mod 26 = 21→v
(2*7+5*0) mod26=14→o

ts= $\begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} * \begin{bmatrix} 19 \\ 18 \end{bmatrix} = \begin{bmatrix} 7 \\ 24 \end{bmatrix} = \begin{bmatrix} h \\ y \end{bmatrix}$

Note: (3*19+3*18) mod 26= 111mod 26=7→ h
(2*19+5*18) mod 26=128 mod 26 = 24→y

ciphertext=vohy

Decryption process→

ciphertext=vohy

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

C= vo ro → vo = [21 14] , hy= [7 24]

1-Find the determine for the **key** =$\begin{bmatrix} \mathbf{3} & \mathbf{3} \\ \mathbf{2} & \mathbf{5} \end{bmatrix}$→ 3*5-2*3= 15-6=9

2- find 9^{-1}mod 26→ inv=3 (حسب طريقة احتساب المعكوس)

$\begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix}$ mod 26 = $\begin{bmatrix} 5 & 23 \\ 24 & 3 \end{bmatrix}$ معالجة الاشارة السالبة من خلال الجمع ب 26

$\begin{bmatrix} 5 & 23 \\ 24 & 3 \end{bmatrix}$*3 mod 26=$\begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix}$

$\begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix}$*$\begin{bmatrix} 21 \\ 14 \end{bmatrix}$ mod 26 = $\begin{bmatrix} 7 \\ 0 \end{bmatrix}$= h a

Note: (15*21+17*14) mod 26= 553 mod 26 =7 →h
(20*21+9*14) mod 26 =546 mod 26 =0→a

$\begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix}$*$\begin{bmatrix} 7 \\ 24 \end{bmatrix}$ mod 26 = $\begin{bmatrix} 19 \\ 18 \end{bmatrix}$ =ts

Note : (15*7+17*24) mod 26 = 513mod 26= 19 →t

(20*7+9*24) mod 26 =356 mod 26=18→s

Plaintext= **Hats**

References:

1- Managing Cisco network security: building rock-solid networks 2000
2- Cryptography and Network Security, William Stalling , 2003

Chapter 5

Viruses

1- Understanding Viruses

Macro ;A list of instructions stored as data that is interpreted by a scripting host. To combat viruses effectively, you need to understand how they propagate

and what defenses are available. Computers store two entirely different types of information:

a- executable*code* and *data.*

Code specifies how the computer should operate, while data provides the information on which the operation is performed. For example, in the equation 1+1=2, the 1, 1, and 2 are the data and the + and = are code. The difference between code and data is crucial to virus defense because only code is susceptible to virus infection. Viruses can corrupt data but cannot propagate using pure data because data does not provide an execution environment for viruses to run in.

b- interpreter

A programming language application that loads scripts as data and then interprets commands step-by-step rather than by compiling them to machine language. But it's not always clear what is data and what is code. Is a Word document code or data? It's mostly data, but because Word documents can contains *macros* that Word interprets in order to perform complex operations, it can also contain code. The same goes for any macro-enabled application that stores a mixture of code and data in a single document. Applications that look at data and then perform wide-ranging operations based on that data are called *execution environments, interpreters,* Or *scripting hosts.*

c- scripting hosts

Execution environments that can be called from applications in order to execute scripts contained in the application's data. You could consider any use of data to be interpretation, but for viruses to propagate, the execution

environment needs to be complicated enough to allow for self-replication. This is a very high-level function that is typically only available when a scripting host running an interpreted computer language is built into
an application. What does all this really mean? Simple. You don't have to worry about viruses in pictures, audio files, online movies, and other programs that merely edit or display content, but you do need to worry about programs that use content to control the program's behavior. If the control mechanisms are complex enough to allow self-replication to occur, the application could host viruses. When you search the Web, you'll often see the term *virii* used as the plural for virus.Since *virus* comes from Greek and not Latin, its correct plural is *viruses* , not *virii.* But most hackers don't know that, so they use the term "virii" for the plural.

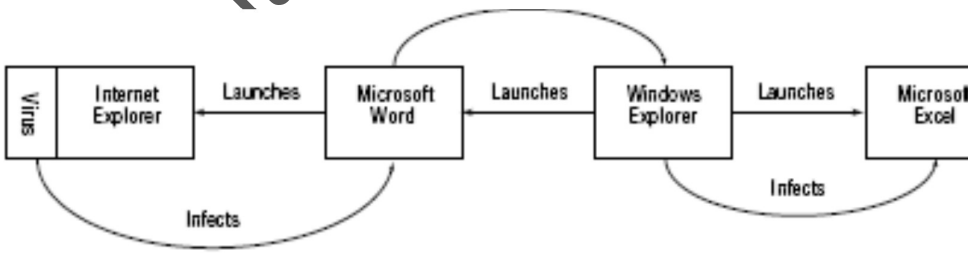
2- Understanding Virus Propagation

When you launch a program on any computer, you are directing an existing running program to launch it for you. In Windows, this program is (usually) Windows Explorer. It could technically be many other programs, like the command prompt or Internet Explorer (if you download the program and choose to open it from its current location). On a Macintosh it's usually the Finder, and in Unix it's the command shell or the X Windows manager. The important concept is that every program is started by another program in an unbroken chain all the way back to the initial bootstrap code stored permanently in the computer's motherboard. So how do viruses wedge themselves into this process? How do they know which programs will spread them and which will simply cause them to lie dormant? Usually, they don't. Viruses only know the

program that was running when they were first executed (like Internet Explorer) and the program that started that program. When a virus is first executed, it attaches itself to the beginning of the current program. The next time that program is run, the virus takes the information about the program that started the current program to determine what file it should infect next and attaches itself to the beginning of that program. In this way, the virus propagates one step closer to the beginning of the chain each time the programs in the startup chain are launched.

3- worms

Viruses that spread over a network automatically without human intervention. Viruses also attach themselves to each program that the current program launches. In most cases this is nothing—most applications can't launch other programs. But some do, and when those applications are found, the virus automatically spreads. The graphic that follows shows how a virus attached to Internet Explorer can propagate back to the program that launched it (Word), then back to Windows Explorer, and from there to other applications.



It's important to note that viruses require human activity—booting a floppy, executing a program, or opening an attachment—in order to activate and spread. Viruses that can spread without human activity are referred to as

worms. Worms typically exploit buffer overrun attacks against common Internet services like mail or web.

4- Common Types of Virus Attacks

Types of viruses are defined mostly by the propagation method they use. In many cases, an entire class of viruses is composed of permutations of just a single virus,so they're nearly equivalent. Viruses are categorized by their ultimate target, asdescribed in the following sections.

- **Boot Sector Viruses:**The first executable code that is stored on a disk and is used to load the operating system.*Boot sector* viruses were the original viruses, and they spread by the only common means of sharing information in the early days of computers—on floppy disks. Twenty years ago, networks were uncommon. Most data was shared by copying it to floppy disks. It was common at that time to boot floppy disks for special purposes like playing games or simply because the floppy had been left in the drive

when the computer was turned off. Boot sector viruses would copy themselves to the boot sector of the host when the floppy was booted and then subsequently infect every floppy that was inserted into the computer.

Thanks to the proliferation of networks, these viruses are practically extinct.

- **Executable Viruses:**

a- **Shell :**A command-line interface to an operating system. Executable viruses infect the startup code for programs and propagate back to the *shell* or desktop application of the computer in order to infect all programs launched from it. Because of the native immunity to this

activity in modern permissionsbased operating systems, these viruses have become rare, except in places where older operating systems are common.

b- **Macro Viruses::** Viruses that exist in the interpreted code embedded in Office documents. These viruses are not capable of escaping the confines of their interpreted environment, so they cannot infect executables. *Macro viruses* are written in a higher-level language, such as the Visual Basic scripting language built into Microsoft Office, so they are related to other interpreted language viruses like those that can infest Java applications. Macro viruses attach themselves to document templates and can spread to other documents that are saved after opening the infected one. They spread like wildfire through corporate networks where users share documents indiscriminately. Luckily, most Office document macro viruses are relatively harmless, and Microsoft has worked to close the security holes in the Office macro languages (the most common macro viruses are specific to Word and Excel). The latest version of Office ships with immunity enabled by default, so macro viruses will become obsolete when this software becomes widespread.

5- E-Mail Viruses

Unfortunately, the same application language has been built into Microsoft Outlook, the e-mail software that comes with Office (and its free-with-the-operating system sibling, Outlook Express). Viruses written for Outlook can automatically read your list of contacts from the Address Book or Contacts folder and mail themselves to everyone you know, thus propagating extremely

rapidly. Currently, Outlook e-mail viruses are by far the fastest spreading and largest threat in the
virus world.

6- Understanding Worms and Trojan Horses

Worms are viruses that spread automatically, irrespective of human behavior, by exploiting bugs in applications that are connected to the Internet. You've probably heard the names of the most widely successful ones in the mainstream media: Code Red, Nimda, and Slammer. From an infected machine, the worm scans the network searching for targets. It then contacts the target, initiates a benign exchange, exploits a bug in the receiver's server software to gain control of the server momentarily, and uploads itself to the target. Once the target is infected, the process starts again on it. Worms usually carry a Trojan horse along with them as payload and set up a listening service on the computer for hackers to connect to. Once a worm is in the wild, hackers will begin port scanning wide ranges of computers looking for the port opened up by the worm's listening service. When a hacker (let's call him Sam) finds a compromised computer, he will typically create an administrative account for himself and then clean up the worm and patch the computer against further exploits—to keep other hackers out so that he can reliably use the computer in the future. The computer is now "owned" by Sam and has become his "zombie," in hacker terms. Because this all happens behind the scenes (and often at night), the real owner of the computer often never knows. But like a parasitic symbiote, people who have been "owned" are sometimes better off having a knowledgeable hacker protecting their zombie from further attacks. Your

computer has probably already been hacked if you have a broadband Internet connection and you don't have a cable/DSL router or a software firewall. It wouldn't show up on a virus scan because the hacker would have cleaned up the worm within a few hours of infection. To take back ownership of your computer, change the passwords on every account on the machine. Hackers like to collect from a few dozen up to (in some cases) a few thousand zombies so that they can perpetrate attacks from many different IP addresses on the Internet. Some hackers actually sell (using auction sites, believe it or not) large banks of zombies to spammers who use them to transmit bulk spam. Anti-hacking researchers leave unprotected computers out on the Internet to allow them to be exploited so that they can track down hackers by watching the activity on the exploited computers, so hackers will typically "bounce" through multiple zombies before perpetrating an attack to throw investigators off their trail. This is going on all around you on the Internet, right now. Worms are basically impossible for end users to prevent, and they typically exploit newly found bugs that are either unpatched or not widely patched in a
vendor's code. When they attack extremely common systems like Windows or Linux, they spread very quickly and can cause enormous damage before they're stopped. Here are some suggestions to defend against worms:
Avoid software that is routinely compromised, like Microsoft Internet
Information Server and Internet Explorer. (Mozilla, a free download
From www.mozilla.org is an excellent replacement for IE on Windows

computers.) Stay up-to-date on patches and security fixes for all your public computers. Strongly consider using automatic updates for any public server, and schedule them for a nightly reboot to make sure that patches become effective as quickly as possible. Keep client computers behind firewalls or cable/DSL routers. Run only those services you intend to provide on public servers—don't just install everything for the sake of convenience when you set up a public server. Use firewalls to prevent worms from reaching the interior of your network from the Internet. Keep your virus-scanning software updated. But even with all these precautions, you can only be protected against worms that the vendors know about, and it's quite likely that a worm will infest your public servers at some point, so keep good backups as well.

7- Protecting Against Worms

There are two common ways to protect against worms. Firewalling services that you don't use is the primary method. However, some services (like web and e-mail) must be open to the Internet and usually cannot be protected against by a firewall. In this case, using software specifically designed to filter the protocol—such as a proxy-based firewall, a supplemental security service like e-eye Secure IIS, or simple URL filtering on the characters used by hackers to insert buffer overruns— can stop the attacks before they get to the firewall. For mail servers, simply putting a mail proxy server from a different operating system in front of your actual mail server will prevent the interior mail server from being affected by any buffer overrun that can affect the proxy. Finally, virus scanners receive signatures that allow them to recognize and (sometimes) clean worms that have already infected a server. In cases where the virus

scanner cannot automatically clean up the worm, antivirus software vendors will provide a downloadable tool that will clean up the infection. Unfortunately, this method doesn't stop worm infection; it merely removes it.

References:

4- Managing Cisco network security: building rock-solid networks2000
5- Cryptography and Network Security, William Stalling , 2003