Lecturer: Dr. Asmaa Alqassab

Level: 4th class

Lab Lectures: Applying Classical and Modern Ciphers using Python

Lab Assistant: L. Ammar Adel Ahmed

INTRODUCTION

Information/Data hiding is an act of protecting information/data from any inadvertent change. It can also be defined as the process of hiding the details of an object. A common use of information hiding is to hide the data so that if it is changed, the change is restricted to a small subset of the total data. It helps to prevent programmers from intentionally or unintentionally changing parts of a program. Information hiding techniques are categorized as:

Steganography, Cryptography and Watermarking. The three main categories of information hiding techniques are shown in figure 1.

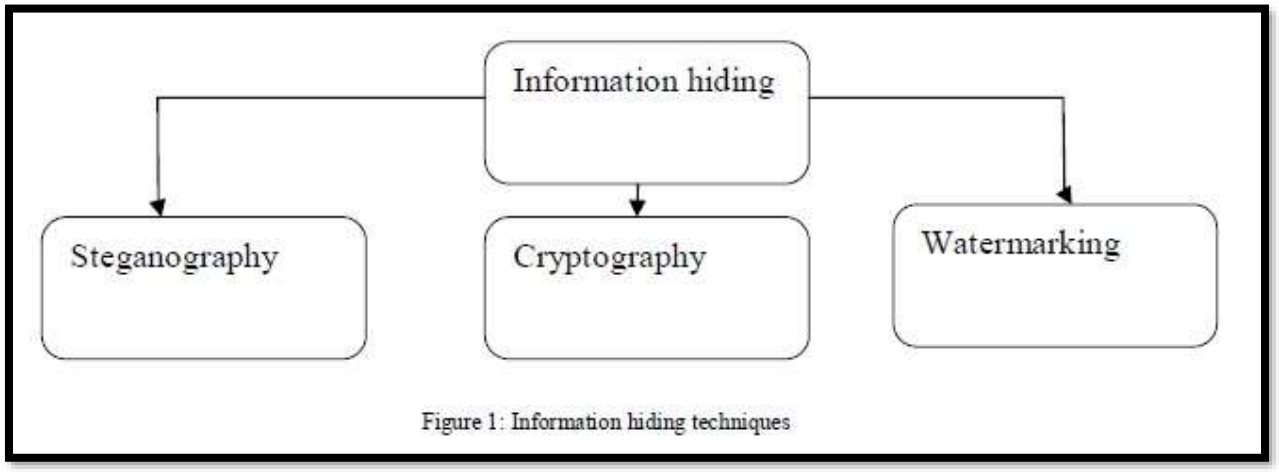


Figure 1: Information hiding techniques

STEGANOGRAPHY

It is a science of hiding information from unauthorized person. Steganography is a practice of concealing a file, message, image, or video within another file, message, image, or video.

The secret message and the cover message are first fed to encoder using steganography techniques and then the same encryption algorithm in reverse order is implemented to recover the original secret message and the original cover. Key plays very important role for retrieving the original message.

The advantage of steganography is that the message to be transmitted is not detectable to the casual eye. In fact, unauthorized people should not even suspect that a hidden message exists. Digital steganography is the art of hiding data within data. Goal of steganography is to hide data well enough that unintended recipients do not suspect the steganography medium of containing hidden data. Media files are ideal for steganography transmission because of their large size.

A basic steganography model is shown in figure 2.

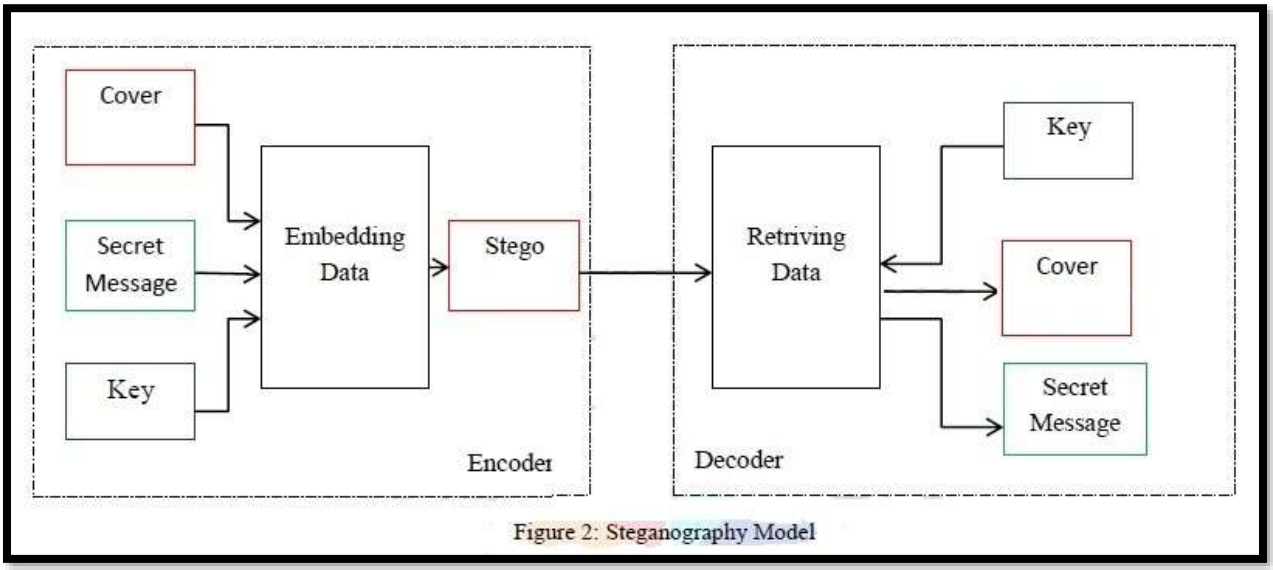


Figure 2: Steganography Model

CRYPTOGRAPHY

Cryptography comes from the Greek word "krypo", meaning hidden, and "graphia", meaning writing. Cryptography is a scheme of storing and transmitting data in a form that only those it is intended for can read and process. Unauthorized people could not extract any information, even if the scrambled messages fell in their hand. It is a skill of protecting information by encoding it into an unreadable format.

Encryption: The process of locking up information using cryptography is encryption. Locked information is encrypted message.

Key: A secret, similar to password that is used during encryption and decryption.

Decryption: The process of unlocking encrypted information using cryptography.

A cryptosystem is an implementation of cryptographic techniques and their accompanying infrastructure to provide information security services. Cryptosystems are divided as:

- Cryptography: It is an art of making cryptosystem capable of providing information security.
- Cryptanalysis: It is an art of breaking the cipher text.

Cryptography deals with the cipher text for transmission or storage. It involves the study of cryptographic mechanism with the intention to break them. Cryptanalysis is also used during the design of the new cryptographic techniques to test their security strengths.

Steps involved in cryptography are shown in figure 3.

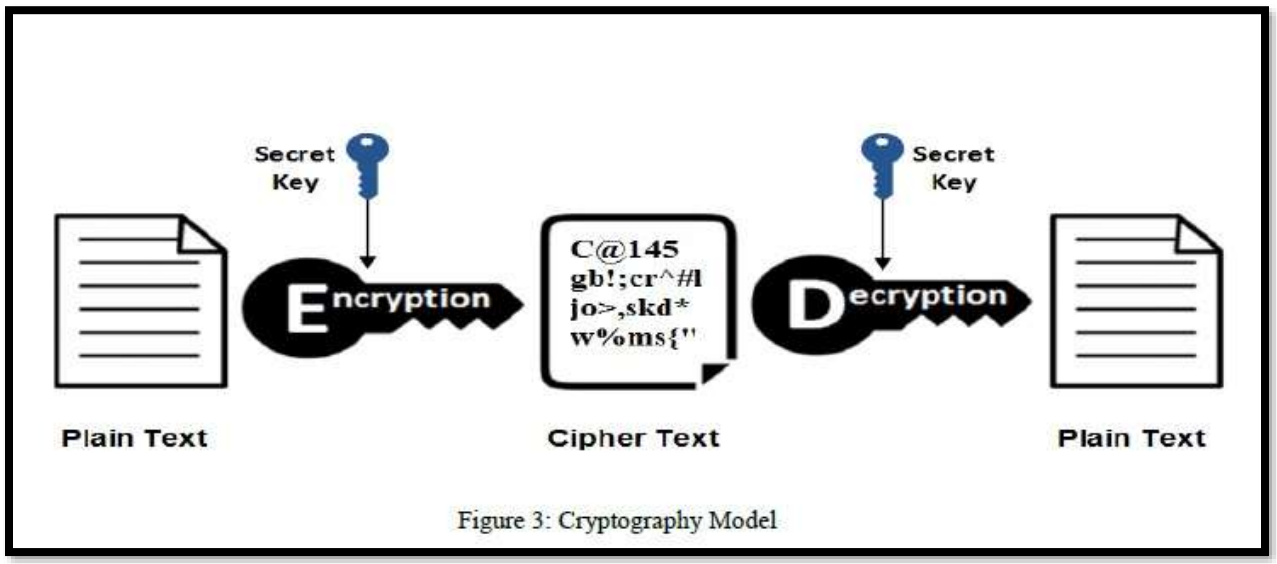


Figure 3: Cryptography Model

WATERMARKING

Digital watermarking is a method for inserting information (the watermark) into an image (visible or invisible). A watermark is a form of text or image that is impressed onto a text or image which provides evidence of its authenticity. Watermarking embeds a signal directly into the data and the signal becomes an integral part of the data, travelling with the data to its destination. Here, it can be assumed that the valuable data is protected as long as the watermark is present in it. Thus, the goal of a watermark must be to always remain present in the host data.

The data is embedded with the watermark and is allowed to travel through the channel (which is insecure). And finally, the watermark is extracted to check authenticity of the data.

Watermarking can be split into two types i.e., Fragile and Robust.

- Fragile: It involves embedding information into a file which is destroyed if the file is modified. This method is not suitable for recording the copyright holder of the file since it can be so easily removed. It is useful in situations where it is important to prove that the file has not been tampered. This technique is considered to be easier for implementation than robust methods.
- Robust: Embedded information cannot be easily destroyed. The watermark should be hidden in a part of the file where its removal would be easily perceived.

A digital watermark life cycle is shown in figure 4.

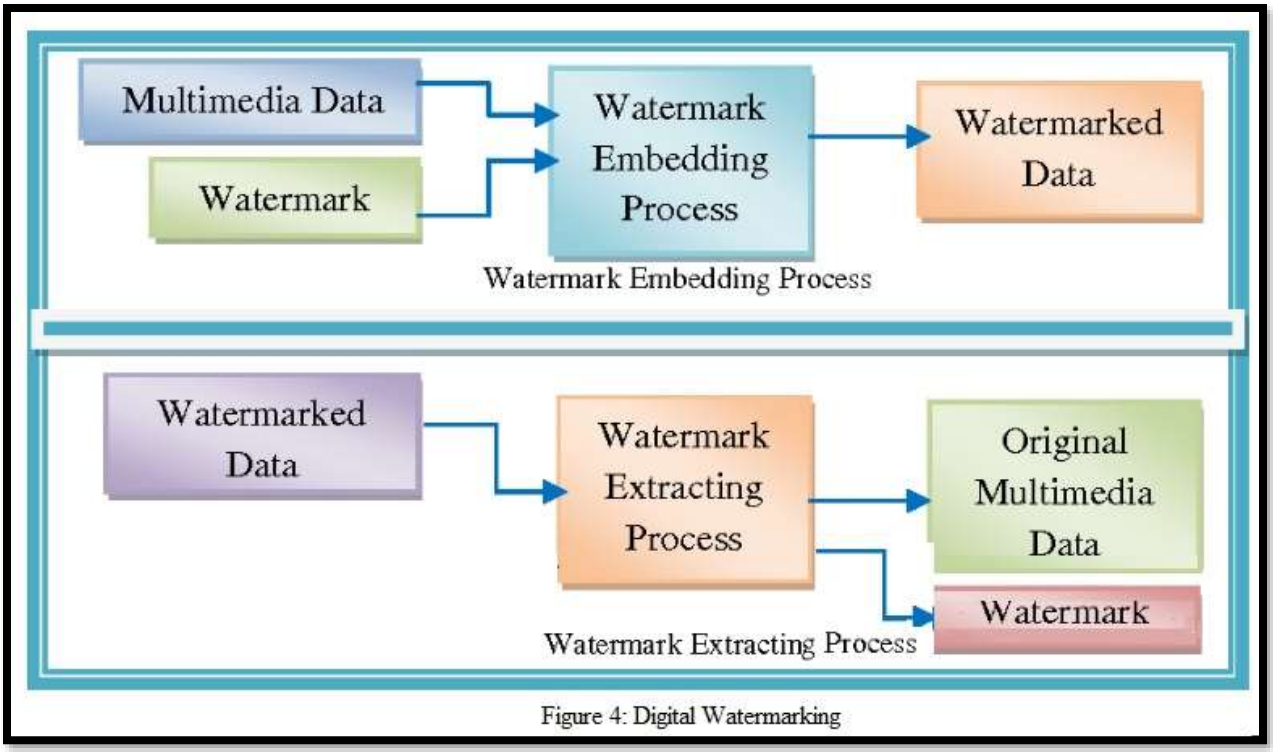


Figure 4: Digital Watermarking

STEGANOGRAPHY VS CRYPTOGRAPHY VS WATERMARKING

After analysis of the three techniques, the following thing we got, table 1.

Table 1: Comparison table of three techniques

Attribute	Steganography	Cryptography	Water Marking
Techniques	LSB, Spatial Domain, Jsteg, Outguess	Transposition, Substitution, RSA	compensated prediction, DCT
Naked eye Identification	No, as message is Hide within other carrier (cover image)	Yes, as message is convert in Other way, which sough something is hidden	Yes, as actual message is hiding by some watermark
Capacity	Differs as different Technology usually low hiding capacity	Capacity is so high, but as message is long it chances to be decrypt	Capacity depends on the size of hidden data.
Detection	Not easy to detect because to find steganographic image is hard.	Not easy to detect ,depend on technology used to generate	Not easy to detect
Strength	Hide message without altering the message, it conceals information	Hide message by altering the message by assigning key	Extend information and become an attribute of the cover image
Imperceptibility	High	High	High
Applicability	Universally	Universally	Universally
Robust	Yes	Yes	Yes

APPLICATIONS

Steganography is applicable to, but not limited to, the following areas:

 To establish secure and secret communications where cryptographic encryption methods are not available.

 Steganography is utilized military applications, where the two parties' communications are of large importance.

 The health care, and especially medical imaging systems, may very much benefit from information hiding techniques.

 These techniques are applied for protection of data alteration.

 Steganography is also applicable to media database systems.

Cryptographic techniques are utilized in following domains:

· Authentication/Digital Signatures: Authentication is the process by which claimed identities of users are verified.

· Secure Communication is the most straight forward use of cryptography.

· Another application of cryptography, called secret sharing, allows the trust of a secret to be distributed among a group of people.

· Time stamping which uses blind signature schemes that allow the sender to get a message receipted by another party without revealing any information about the message to the other party.

· Electronic money transfer techniques are utilizing cryptographic concept for secure transfer of funds.

· Secrecy in storage is usually maintained by a one-key system and the key is provided at the beginning of sessions.

Applications of watermarking:

· Temper proofing to find out if the data was tempered.

· Ownership Assertion to prove ownership

· Fingerprinting to identify the buyer of the content

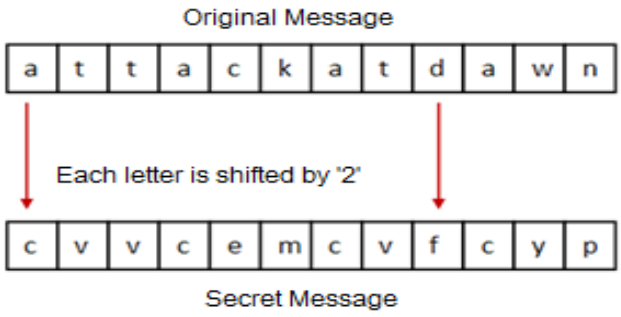
· Broadcast monitoring to determine royalty payments.

Chapter One

Hieroglyph – The Oldest Cryptographic Technique

The first known evidence of cryptography can be traced to the use of 'hieroglyph'. Some 4000 years ago, the Egyptians used to communicate by messages written in hieroglyph. This code was the secret known only to the scribes who used to transmit messages on behalf of the kings.

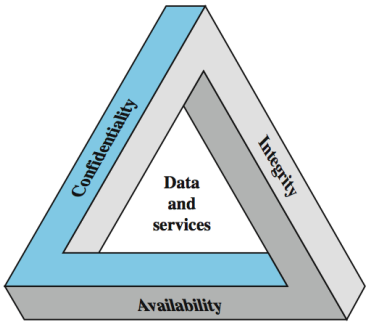
Later, the scholars moved on to using simple mono-alphabetic substitution ciphers during 500 to 600 BC. This involved replacing alphabets of message with other alphabets with some secret rule. This **rule** became a **key** to retrieve the message back from the garbled message.



Information 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.



Goals of Computer Security

Integrity

Integrity deals with prevention of unauthorized modification of intentional or accidental modification.

- **Data integrity**: assures that information and programs are changed only in a specified and authorized manner
- **System integrity**: Assures that a system performs its operations in unimpaired manner.

Confidentiality

The concept of *Confidentiality* in information security relate 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.

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 (confidentiality and integrity).

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.

Preventing Unauthorized Access

Authentication credentials

Information users provide to identify themselves for computer access

- **User knowledge** Name, password, PIN
- **Smart card** A card with an embedded memory chip used for identification
- **Biometrics** Human characteristics such as fingerprints, retina or voice patterns

Type of Attacks

1- Attacks on Hardwar

2- Attacks on Software

a- Software Deletion

b- Software Modification (Trojan horse),(trapdoor), (Program that leaks information)

c- Software Theft

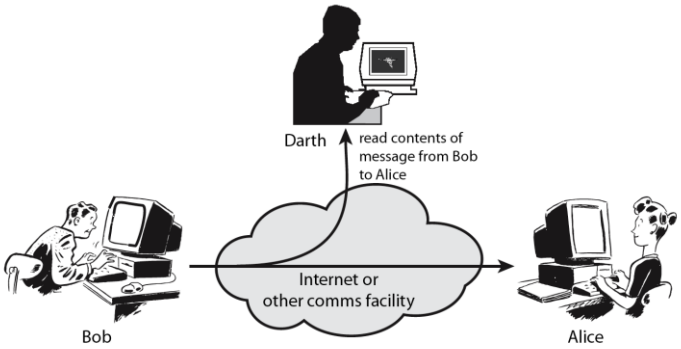
3- Attacks on Data

Security Attack

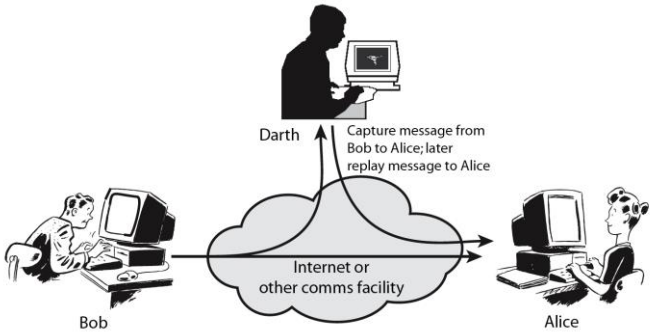
- any action that compromises the security of information owned by an organization
- information security is about how to prevent attacks, or failing that, to detect attacks on information-based systems

- often *threat* & *attack* used to mean same thing
- have a wide range of attacks
- can focus of generic types of attacks

- passive
- active



Passive Attacks



Active Attacks

Denial of Service, A "denial-of-service" attack is an attempt by attackers to prevent legitimate users of a service from using that service. Examples include

– Flooding the network to overwhelm the daemons and servers

– Disrupting connections between systems

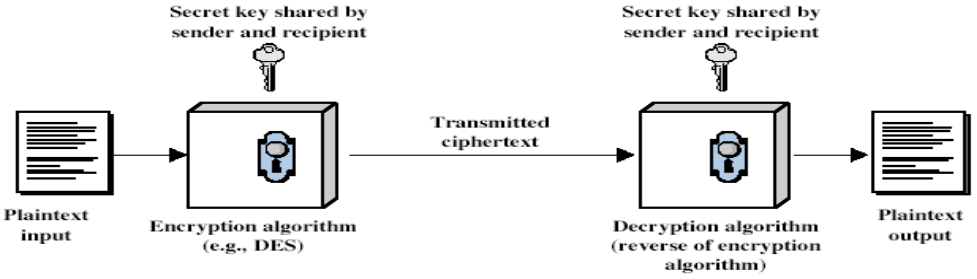
– Attempts to prevent a particular individual from accessing a service.

Basic Terminology

In cryptographic terminology, the message is called **plaintext** or **cleartext**. Encoding the contents of the message in such a way that hides its contents from outsiders is called **encryption**. The encrypted message is called the **ciphertext**. The process of retrieving the plaintext from the ciphertext is called **decryption**. Encryption and decryption usually make use of a **key**, and the coding method is such that decryption can be performed only by knowing the proper key.

Cryptography is the art or science of keeping messages secret. **Cryptanalysis** is the art of **breaking** ciphers, i.e. retrieving the plaintext without knowing the proper key. People who do cryptography are **cryptographers**, and practitioners of cryptanalysis are **cryptanalysts**.

Cryptography deals with all aspects of secure messaging, authentication, digital signatures, electronic money, and other applications. **Cryptology** is the branch of mathematics that studies the mathematical foundations of cryptographic methods.



There are two classes of key-based algorithms, **symmetric** (or **secret-key**) and **asymmetric** (or **public-key**) algorithms. The difference is that symmetric algorithms use the same key for encryption and decryption (or the decryption key is easily derived from the encryption key), whereas asymmetric algorithms use a different key for encryption and decryption, and the decryption key cannot be

derived from the encryption key.

Types of Cryptosystems

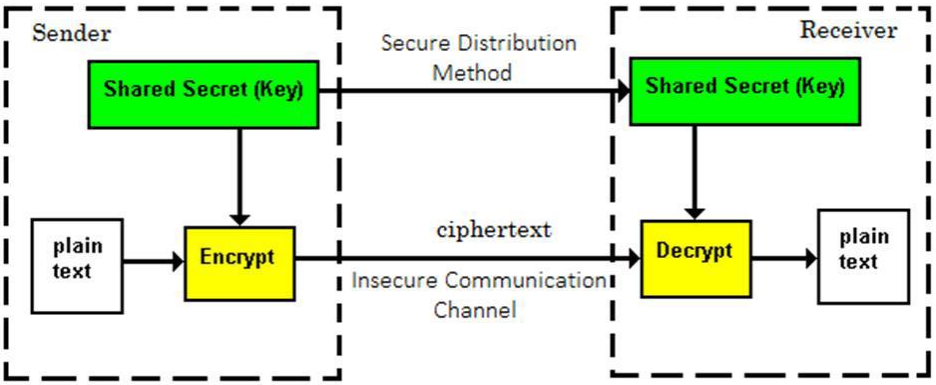
Fundamentally, there are two types of cryptosystems based on the manner in which encryption-decryption is carried out in the system:

- Symmetric Key Encryption
- Asymmetric Key Encryption

The main difference between these cryptosystems is the relationship between the encryption and the decryption key. Logically, in any cryptosystem, both the keys are closely associated. It is practically impossible to decrypt the ciphertext with the key that is unrelated to the encryption key.

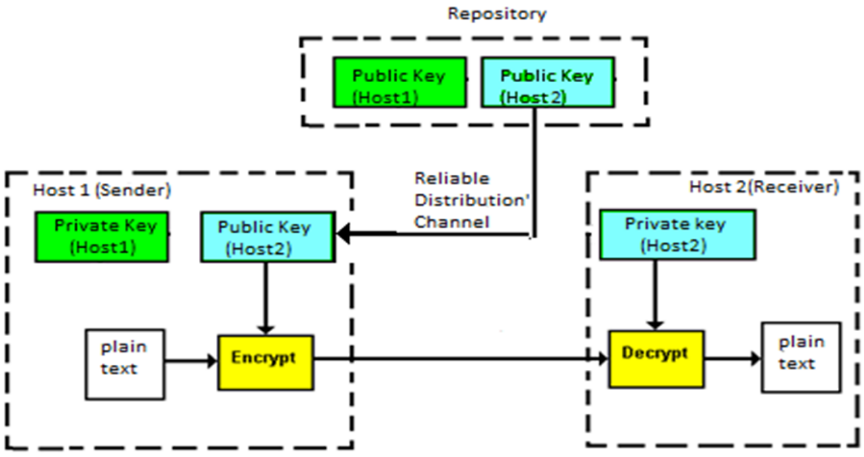
Symmetric Key Encryption

The encryption process where **same keys are used for encrypting and decrypting** the information is known as Symmetric Key Encryption. The study of symmetric cryptosystems is referred to as **symmetric cryptography**. Symmetric cryptosystems are also sometimes referred to as **secret key cryptosystems**.



Asymmetric Key Encryption

The encryption process where **different keys are used for encrypting and decrypting the information** is known as Asymmetric Key Encryption. Though the keys are different, they are mathematically related and hence, retrieving the plaintext by decrypting ciphertext is feasible. The process is depicted in the following illustration:



Generally, symmetric algorithms are much faster to execute on a computer than asymmetric ones. In practice they are often used together, so that a public-algorithm is used to encrypt a randomly generated encryption key, and the random key is used to encrypt the actual message using a symmetric algorithm.

Chapter Two
CLASSICAL ENCRYPTION TECHNIQUES

2.1 Cryptography Classification

The old Encryption and Decryption techniques before the implementation of computer systems are called Classical techniques, while those invented and implemented for the computer systems are called modern techniques. However, cryptography system (whether Classical or Modern) are generally classified along three independent dimensions:

1- The **type of operations** used for transforming plaintext to cipher text. All encryption algorithm are based on general principle:

(a) Substitution,

(b) Transposition,

(c) Bit manipulation.

2- The **number of keys** used.

(a) **Symmetric:** If the same key is used by both, the sender and the receiver for encryption and decryption. It might be also called **Single key, Secret key, or Conventional encryption.**

(b) **Asymmetric:** If the sender and receiver, each were using different keys, usually two sets of keys, one for encryption and the other for decryption.

3- The **way,** in which the plantext is processed.

- **Block cipher**: The input message is divided in blocks of elements and each block is processes at a time, producing an output block for each input block.
- **Stream cipher**: The input elements are processed individually, producing an output as one element at a time, too.

2.2 Symmetric Cipher Model:

All traditional schemes are symmetric / single key / private-key encryption algorithms, with a single key, used for both encryption and decryption, since both sender and receiver are equivalent, either can encrypt or decrypt messages using that common key.

However, there are hundreds of traditional methods for information security which all employ (1) **Substitution** or (2) **Transposition** techniques (or both), however, they can be categorized into only two techniques, symmetric and asymmetric systems, which are well suited and implemented for computer system applications, which will be studied during the course.

The basic terminology used:

- **plaintext** - the original message
- **ciphertext** - the coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering ciphertext from plaintext
- **cryptography** - study of encryption principles/methods
- **cryptanalysis (code breaking)** - the study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - the field of both cryptography and cryptanalysis

A simplified model of conventional encryption/decryption system is shown in figure 2-2.

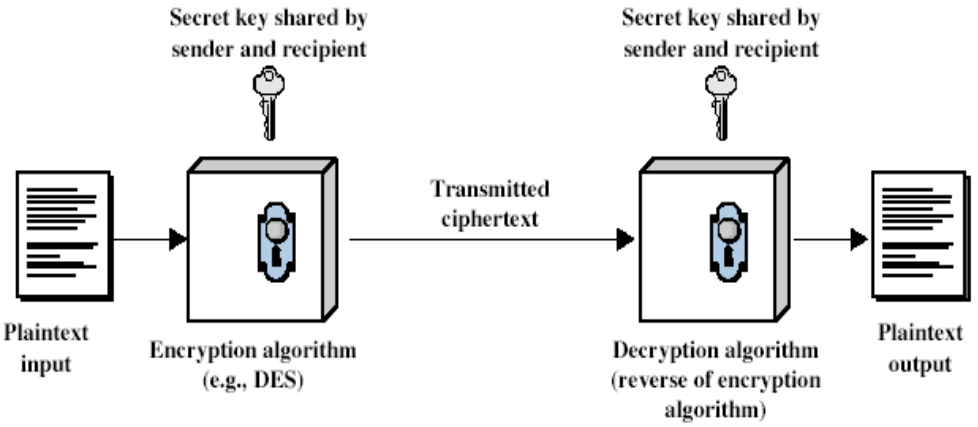


Fig. 2-2 simplified cipher model

The five ingredients of the symmetric cipher model of figure 2-1 are:

- **plaintext**
- **encryption algorithm** – performs substitutions/transformations on plaintext
- **secret key** – control exact substitutions/transformations used in encryption algorithm
- **ciphertext**
- **decryption algorithm** – inverse of encryption algorithm

Requirements:

Two requirements for secure use of symmetric encryption:

1. a strong encryption **algorithm**
2. a secret **key** known only to sender / receiver

Generally one assumes that the algorithm is known. This allows easy distribution of s/w and h/w implementations and hence assume just keeping **key secret** is sufficient to secure encrypted messages.

Having plaintext X, ciphertext Y, secret key k, encryption algorithm E_k and decryption algorithm D_k , the calculation involve

$C = E_k(Y)$ and $X = D_K(Y)$

This implies the need for secure channel to distribute key.

2.2.1 Substitution Techniques:

A substitution technique is one in which the letters of plaintext are replaced by other letters or numbers or symbols. But, if plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.

Few security techniques will be considered here as examples for substitution cipher.

1. Caesar Cipher:

Substitution ciphers form the first of the fundamental building blocks. The core idea is to replace one basic unit (letter/byte) with another. Whilst the early Greeks described several substitution ciphers, the first attested use in military affairs of one was by Julius Caesar, described by him in *Gallic Wars* (cf. Kahn pp83-84). Still any cipher using a simple letter shift is called **Caesar cipher**, not just those with shift **3**.

Caesar cipher involves replacing each letter of the alphabet with a letter standing 3 places further down the alphabet. Therefore the alphabet transformation sets for plain and cipher are:

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

Example 1.
Encipher the message: plaintext = "COME HERE" by Caesar cipher.
Solution: Ciphertext = "FRPH KHUH"

Example 2..
Plaintext = "meet me after the toga party"
Ciphertext = "PHHW PH DIWHU WKH WRJD SDUWB"

This mathematical description uses **modulo arithmetic** (i.e. clock arithmetic). Here, when you reach **Z** you go back to **A** and start again. Mod **26** implies that when you reach **26**, you use **0** instead (i.e. the letter after **Z**, or **25 + 1** goes to A or 0).

Mathematically, if we assign a numerical equivalent to each letter (a=1, b=2, etc.), i.e.

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

Then the algorithm can be expressed as follow: For each plaintext letter p, substitute the cipher text letter C:

$$C = E(p) = (p + 3) \bmod (26)$$

A shift may be of any value **k**, so that the general Caesar algorithm is

$$C = E(p) = (p + k) \bmod (26)$$

Where **k** takes on a value in the range 1 to 25.

The Decryption algorithm is simply

$$p = D(C) = (C - k) \bmod (26)$$

2. Monoalphabetic Cipher

Arbitrary substitution of letters in the alphabet gives dramatic increase in the key space.

e.g. one possibility:

plain: **a b c d e f . . . x y z**

cipher: **K M Z A F R . . . D S E**

There is a significant improvement in the security of a message encoded by using a randomized version of the alphabet. There are **26** factorials (**26!**) ways to arrange the alphabet, with the inclusion of space, that number becomes (**27!**) ways. Therefore, **26!** Is a very large number which equals to about **4×10^{26}** possible keys.

***Therefore*, this technique is quite safe using *Brute-Force*,** however, another line of attack is possible.

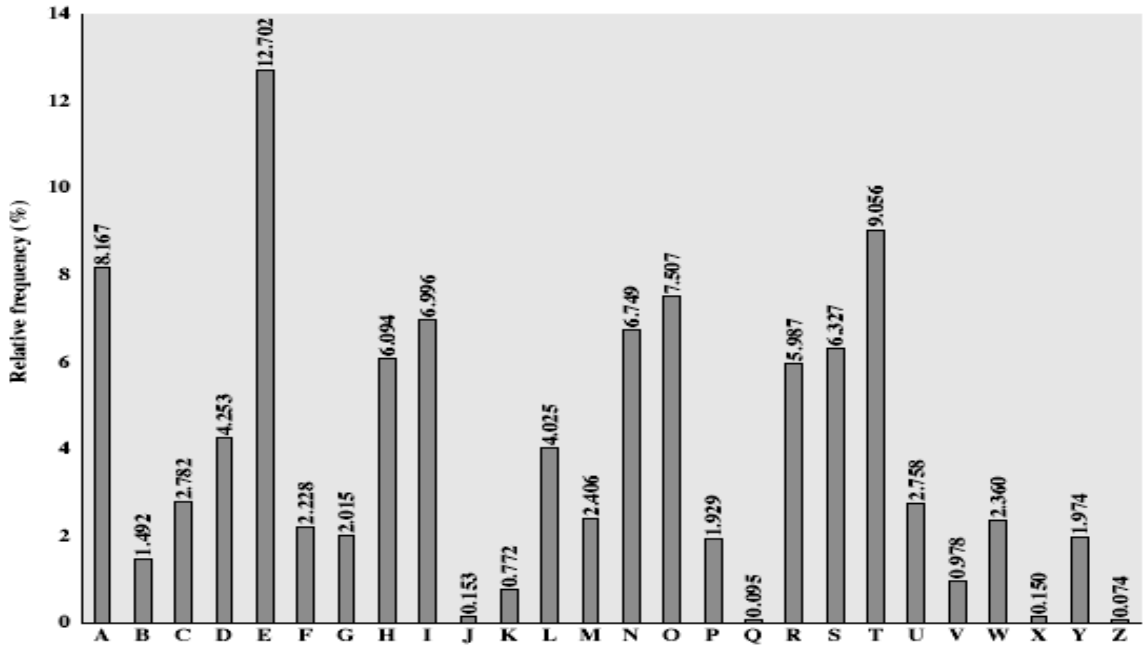


Fig. 2-3 Frequency counts for English alphabet

- **guess P & Z are e and t**
- **guess ZW is the and hence ZWP is the**
- **proceeding with trial and error finally get:**

"it was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in Moscow" .

3. **Playfair Cipher:** (Invented by Charles Wheatstone in 1854, but named after his friend Baron Playfair). This method uses multiple letter encryption cipher.
   - a 5X5 matrix of letters based on a keyword.
   - fill in letters of keyword (no duplicates)
   - fill rest of matrix with other letters
   - eg. using the keyword MONARCHY, then table 2-2 is constructed (no duplicate letters and I & J are counted as one letter).

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

Notes: Plaintext encrypted two letters at a time.

1- If a pair is a repeated letter, insert a filler like 'X',

e.g. "balloon" encrypts as "ba lx lo on".

2- If both letters fall in the same row, replace each with letter to right (wrapping back to start from end),
e.g. "ar" encrypts as "RM".

3- If both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom),
e.g. "mu" encrypts to "CM".

4- Otherwise each letter is replaced by the one in its row in the column of the other letter of the pair,
e.g. "hs" encrypts to "BP", and "ea" to "IM" or "JM" (as desired).

5- If the number of letters is odd, use a filler letter X.
6- Decryption is done in reverse direction, then remove extra X's (or fillers).

Example: encipher the message: "**HASHIMY**".

Solution: Message is arranged in two letters: "**HA SH IM YX**", Then the Cipher will be: "**BO PB EA BW**".

Remarks: In playfair Cipher,

1- There is 26X26 = 676 diagram
2- Frequency analysis is more difficult,
i.e. it has good strength against ciphertext-only attack.

4. Hill cipher

Another interesting multi-letter cipher is **Hill cipher**, developed by Lester Hill in 1929. Here, the technique works as follows:

- The encryption algorithm takes **m** successive plaintext letters, substituting for **m** cipher letters according to **m** linear equations.

- Each character is assigned a numerical value:

(a = 0, b = 1, . . . z = 25)

For **m = 3**, the system can be described as follows:

$$C1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26$$

$$C2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26$$

$$C3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26$$

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

$$\begin{pmatrix} C1 \\ C2 \\ C3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

Or $C = K\,P$

Where C and P are column vectors of length 3 and K is a 3X3 matrix.

Example :

Consider the plaintext "pay more money" then use the encryption key

$$\mathbf{K} = \begin{pmatrix} \mathbf{17} & \mathbf{17} & \mathbf{5} \\ \mathbf{21} & \mathbf{18} & \mathbf{21} \\ \mathbf{2} & \mathbf{2} & \mathbf{19} \end{pmatrix}$$

Solution:

Take m = 3, i.e. three letters at a time. If the first **three** letters of plaintext are represented by the vector (**15 0 24)** instead of **"pay", t**hen

K(15 0 24) = (375 819 486) mod 26

= (11 13 18) = LNS.

Continuing in this fashion, the ciphertext for the entire plaintext will be: "**LNSHDLEWMTRW"**

Decryption requires using the inverse of the matrix **K**. The inverse K^{-1} of the matrix K is defined by the equation $\mathbf{KK^{-1} = K^{-1}\,K = I,}$ Where I is the identity Matrix.

$$K^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

This is demonstrated as follow:

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 9 & 15 \\ 17 & 6 \\ 24 & 0 \\ 17 & \end{pmatrix} = \begin{pmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{pmatrix} \bmod 26$$

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

We have, $C = E_K(P) = K\,P \bmod 26$

and $P = D_K(C) = K^{-1}\,C \bmod 26$

$= K^{-1}\,K\,P \bmod 26 \rightarrow P$ because $K^{-1}\,K = I$

Hill cipher is also strong against a ciphertext only attack as it hides letter frequencies. There are 26 X 26 = 676 digraph frequencies for 2 letters, 26^3 for m = 3, 26^4 for m = 4, etc. **However, it is breakable using matrix analysis.**

5. Polyalphabet Cipher:

Multi-alphabet sets are used with:

1- A set of related mono-alphabetic substitution rules is used.
2- A key determines which rule is chosen.

One famous cipher is **Vigenere** Cipher

Vigenere Cipher: Vigenere Cipher consists of 25 Caesars Cipher shifts from 0 to 25. Each is denoted by a key letter. A table is then constructed as shown in table 2-3.

Table 2-3The modern Vigenere tableau

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

Example: Encrypt the message "CRYPTOGRAPHY SYSTEM" with key = .

- **Distribute the key on the message as follow**

P	C	R	Y	P	T	O	G	R	A	P	H	Y		S	Y	S	T	E	M
K																			
C																			

- **Intersect the plain with key depending on the above table:**

The rule of substitution:

Given a key letter X for the alphabet set and a plaintext letter **y** for the column, i.e. in this case it is V.

Example: Encrypt the message M, where

M: "we are discovered save yourself"

using Vigenere cipher with a

key: "deceptive".

Solution

Plaintext: "wearediscoveredsaveyoursef"

Key: "deceptivedeceptivedeceptive"

Ciphertext:"zicvtwqngrzgvtwavzhcqyglmgj"

Decryption is equally simple:

1- Key letter identifies the row.
2- Ciphertext letter in the row identifies the column.
3- Plaintext letter is at the top of the column.

► A modification to prevent ciphertext letter frequency analysis attack suggest the use of key that is built from a keyword and the message itself.

i.e. for the previous example,

Key: "deceptivewearediscoveredsav"

Plaintext: "wearediscoveredsaveyoursef"

Ciphertext: "zicvtwqngkzeiigasxstslvvwla"

2.2.2 Transposition Techniques

A very different kind of mapping is achieved by performing some sort of permutation on the plantext letters. This technique is referred to as a *transposition* cipher.

1- Rail fence Technique

It is the simplest transposition cipher, in which the **plaintext** is written down as a sequence of diagonals (columns) and then read off as a sequence of rows.

For example, to encipher the message "meet me after the toga party" with a rail fence of depth 2, we write the message **M** as follows:

Solution:

m e m a t r h t g p r y

e t e f e t e o a a t

The encrypted message **C** is "MEMATRHTGPRYETEFETEOAAT"

Example: Encrypt the message

M: "DISCONNECT THE PLUGS NOW"

Solution:

Rewritten the message first in the form:

		S				**N**				**T**				**L**				**N**		
	I		**C**		**N**		**E**		**T**		**H**		**P**		**U**		**S**		**O**	
D				**O**				**C**				**E**				**G**				**W**

Then the ciphertext is taken as:

C: "SNTLNICNETHPUSODOCEGW"

This sort of thing would be trivial to crypt.

2- Matrix transposition

A more complex Scheme is to write the message in a rectangle, row by row, then read the message off: column by column, but permute the order of the columns. The order of the columns then becomes the key to the algorithm. The blanks are filled with characters.

Example:

Encrypt the message **M** = "attack postponed until two am", using the key :
4 3 1 2 5 6 7

Solution:

Write the message **M** in a rectangle having 7 columns.

key: 4 3 1 2 5 6 7

Plaintext: a t t a c k p

o s t p o n e

d u n t I l t

w o a m x y z

The output cipher text is going to be

C: "TTNAAPTMTSUOAODNCOIXKNIPETZ"

- The transposition cipher can be made more secure by performing more than one transposition stage. The result is a more complex permutation that is not easily reconstructed.

For instant, if the output of the previous example is rewritten in a rectangle again and taking up column in the same key sequence, i.e. key: 4 3 1 2 5 6 7, then see the following:

Key: 4 3 1 2 5 6 7

Input: t t n a a p t

m t s u o a o

d w c o I x k

n l y p e t z

The output would be

C: "NSCYAUOPTTWLTMDNAOIEPAXTTOKT"

►To fool the cryptanalyst, i.e. to make his work more difficult, the letters can be replaced by their sequence numbers in the text, which will be of the form:

The message characters are numbered sequentially,

i.e. **"01 02 03 04 05 . . . . .28"**

Then, after the 1st transposition, it becomes:

"03 10 17 24 04 . . . 21 28"

And after the 2nd transposition, it becomes:

"17 09 05 27 24 . . . 06 28"

Which has no regular structure.

3- Code Book

Another example of the transposition cipher is the use of "**Code Book**", i.e. using a code book or table for enciphering, as shown in the following example.

Example

The code book is shown in the table:

Word	Code
BAKER	1701
FRETTING	5603
GUITARIST	4008
LOAFING	3790
.	.
.	.

For a message (plaintext):

***M**: "LOAFING BAKER"*

*The ciphertext **C**, when the code book is used will be :*

C: "3790 1701"

2.2.3 Bit-Manipulation ciphers

Bit manipulation ciphers are well-suited for computer use because they employ operations easily performed by the system.

- The ciphertext looks like unused or crashed files and thereby confusing any one who tries to gain access to the file.

- Bit manipulation ciphers covert plaintext into cipher text by altering the actual bit pattern of each character through the use of one or more of the character through the use of one or more of the following logical operations: AND, OR, NOT, XOR, 1's Complement.

► An improved method of bit-manipulation coding uses the XOR operator. The XOR operator has the following truth table

A	B	Y
0	0	0
1	1	0
1	0	1

For example

Plaintext: **1 1 0 1 1 0 0 1**
Key: **0 1 0 1 0 0 1 1**

Then the Ciphertext: 1 0 0 0 1 0 1 0

Chapter 3
Modern Encryption Techniques

3.1 Inroduction

The objective of this part is to illustrate the principle of modern encryption techniques.

We focus on the most widely used encryption algorithm: the

"Data encryption standard (DES)"

3.1 Simplified Data Encryption Standard (S-DES)

S-DES is an educational rather than a secure encryption algorithm. It has similar structure to **Data encryption standard** DES but with much smaller parameters. It was developed by professor Edward Schaefer of Santa Clara University.

3.2 S-DES Structure

As shown below the overall structure of DES:

The S-DES **encryption** algorithm takes an

(1) 8 bits block of text (example 10111101),

(2) 10-bit keys as input and

(3) Produces an 8-bit block of cipher.

Also the S-DES decryption algorithm takes an 8-bit block of ciphertext and the same 10-bits key used to produce the original 8-bits plaintext.

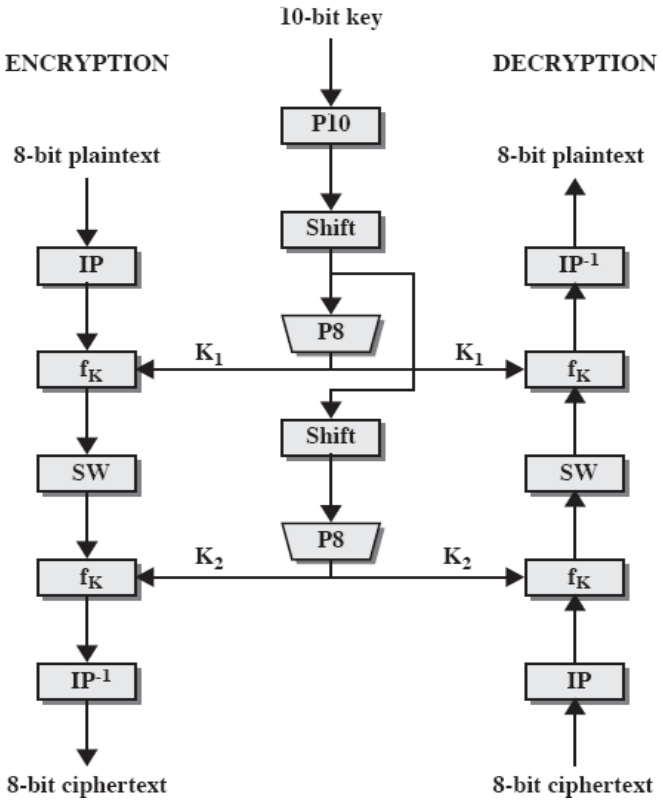


Figure 3-1. S-DES Scheme

1. Encryption:

The encryption algorithm involves **five functions**:

a. An initial permutation (IP);

b. A complex functions labeled f_k which involves both permutation and substitution operations and depends on a key input a simple permutation function

c. Switches (SW) the two halves of the data;

d. The function f_k again, and

e. Finally a permutation function that is the inverse of initial permutation (IP^{-1}).

We can express the encryption algorithm as a composition of functions:

$$IP^{-1}.F_{K2}.SW.F_{K1}.IP$$

Which can be written as:

$$\text{Ciphertext} = IP^{-1}\,(F_{K2}(SW(F_{K1}(IP(\text{Plaintext})))))$$

Where: $K_1 = P_8\,(\text{shift}(P_{10}(\text{key})))$ and $K_2 = P_8\,(\text{shift}(\,\text{shift}(P_{10}(\text{key}))))$

2- Decryption:

Decryption is also shown in the above figure and essentially it is the reverse of encryption, i.e.

$$\text{Plaintext} = IP^{-1}\,(F_{K1}(SW(F_{K2}\,(IP(\text{Ciphertext})))))$$

Block Cipher Principle:

- ***Stream Cipher***: it encrypts data as stream of characters or bytes, e.g. Vigenere Cipher.

- ***Block cipher***: a block of data is encrypted together, block size of 64 or 128 bits is used. E.g. S-DES, DES, 3DES, etc.

More International Symmetric Algorithms

This part include the most important symmetric block cipher in current use. The cipher were selected based on a number of criteria:

1- They are popular in internet applications.
2- They illustrate modern symmetric block cipher techniques that have been developed.
3- Considerable cryptographic strength.

These algorithms are:

DES, Blowfish, RC5, RC2 CAST and IDEA.

Chapter 4
Public Key Cryptography

4.1 Introduction

There were two problems associated with symmetric system;

1. Key distribution; This is only achieved by one the following methods

a- Already distributed shared key.

b- Use of key distribution center.

(Both methods are liable to be compromised)

2. Digital signature problem; Electronic documents would need the equivalent of signatures used on paper documents.

Diffie and Hellman have achieved new technique in 1976 that addresses these problems and was completely different from all previous methods of ciphering, it is called public-key cryptosystem.

4.2 Principle of Public key cryptosystems

Public key algorithms rely on **two** keys, **one** key for encryption and **another** key for decryption. These keys are related and the algorithms have the following important characteristics:

1- It is computationally infeasible to determine the decryption key given only knowledge of the algorithm and the encryption key.

2- Either of the two related keys can be used for encryption with the other used for decryption.

Example of public-key system is **RSA** (**R**ivest, **S**hamir and **A**dleman).

Figure 4-1 illustrates the public key encryption process and figure 4-2 illustrates the public key authentication. It consists of six ingredients; i.e.

1. Plaintext.
2. Encryption algorithm.
3. Public key.
4. Private key.
5. Ciphertext.
6. Decryption algorithm.

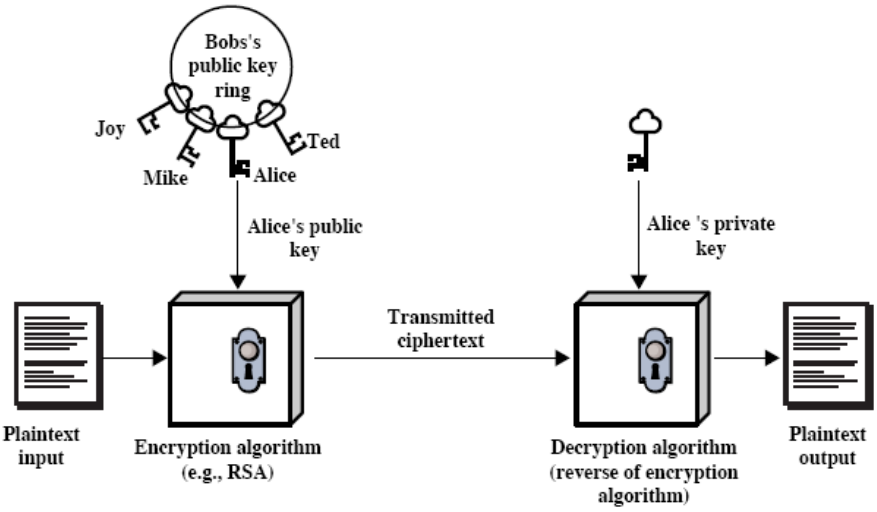


Figure 4-1. Public key encryption

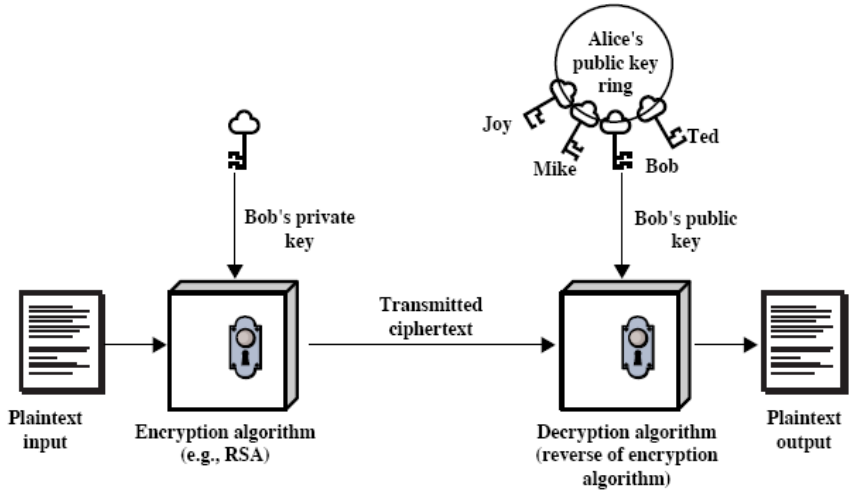


Figure 4-2. Public key Authentication

The essential steps for secrecy shown in figure 4-1 are the following:

1- ***Each end system (user) in the network generates two keys, one for encryption of message at the sender end and the other for decryption at the receiver.***
2- ***Each system (user) publishes its encryption key by placing it in a public register or file. This is the public key and the companion key is kept private. The user also keeps the private keys of all other users.***
3- ***If A (Bob) wishes to send a message to B (Alice), he encrypts the message using B's public key.***
4- ***When B (Alice) receives the message, B decrypts it using her owns private key. No other recipient can decrypt the message because only B knows B's private key***

(Note: No private key distribution, but only public key).

4.3 Symmetric versus public –key Encryption

The following table summarizes some important aspects of Conventional (symmetric) and Public Key (asymmetric) encryption systems.

Conventional Encryption	**Public- key Encryption**
Needed work: • The same **algorithm** with the same **key** is use for encryption and decryption. • The sender and receiver must **share** the algorithm and key.	**Needed work:** • **1. One algorithm** is used for encryption and decryption with a **pair of keys**, one for encryption and one for decryption. • 2- The sender and receiver must each have one of the matched pair of keys (**not the same one**).
Need for Security: ○The key must be kept **secret**. ○It must be **impossible** or at	**Need for Security:** • **One** of the two keys must be kept **secret**. • 2-It must be **impossible** or at least

least impractical to decipher a message if **no** other information is available. ○ Knowledge f the algorithm plus samples of ciphertext must be **insufficient** to determine the key.	impractical to decipher a message if **no** other information is available. • 3-Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be **insufficient** to determine the other key.

To discriminate between the two cryptosystems, we will generally refer to the key used in **symmetric encryption as a secret key**. The two keys used for asymmetric key encryption referred to as **public key and private key.**

4-4. Essential Elements of public-key encryption

To understand how the system works, consider figure 4-3 below, which is suitable for confidentiality or secrecy.

Let the plaintext message which consists of letters in some finite alphabet at the source **A** be $X = \{X_1, X_2, \ldots, X_m\}$

Where **m** is the number of elements of **X**, (e.g. English language alphabet).

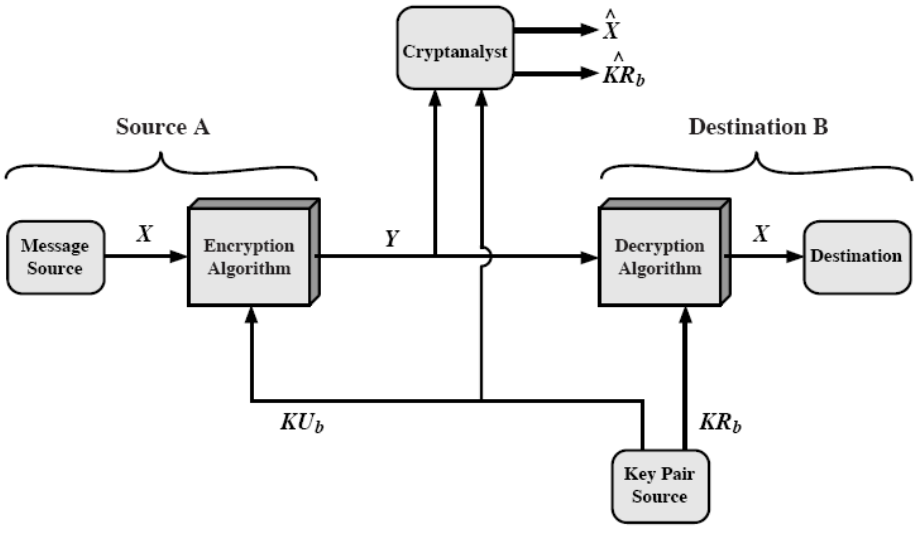


Figure 4-3. Public-key cryptosystem: secrecy

The message is intended to be received by the destination **B**. The intended receiver **B** generates related pair of keys; a public key, $\mathbf{KU_b}$, and private key $\mathbf{KR_b}$. Therefore, key $\mathbf{KR_b}$ is known only to the receiver **B**, whereas key $\mathbf{KU_b}$ is publicly available and therefore accessible by the source **A**.

1- With the message **X** and the encryption key $\mathbf{KU_b}$ as input, the ender **A** forms the ciphertext: $Y = \{Y_1, Y_2, \ldots, Y_m\}$ by $Y = E_{KUb}(X)$
2- The receiver **B**, in possession of the matching private key $\mathbf{KR_b}$, is able to invert the transformation: $X = D_{KRb}(Y)$

We mentioned earlier that either of the two related key can be used for encryption, with the other being used for decryption.

NOTES: This model for secrecy may be attacked by opponent who either;

1- **Interested in the current message and tries to recover a plaintext X^, or**
2- **Interested in future messages, and tries to recover KR_b by generating an estimate KR_b^.**

Figure 4-3, which resembles the action of figure 4-1, has shown the use of public-key cryptosystem for secrecy of message. However, it does not provide for authentication. For authentication purpose, illustrated in figure 4-2, the arrangement of figure 4-4 can be used.

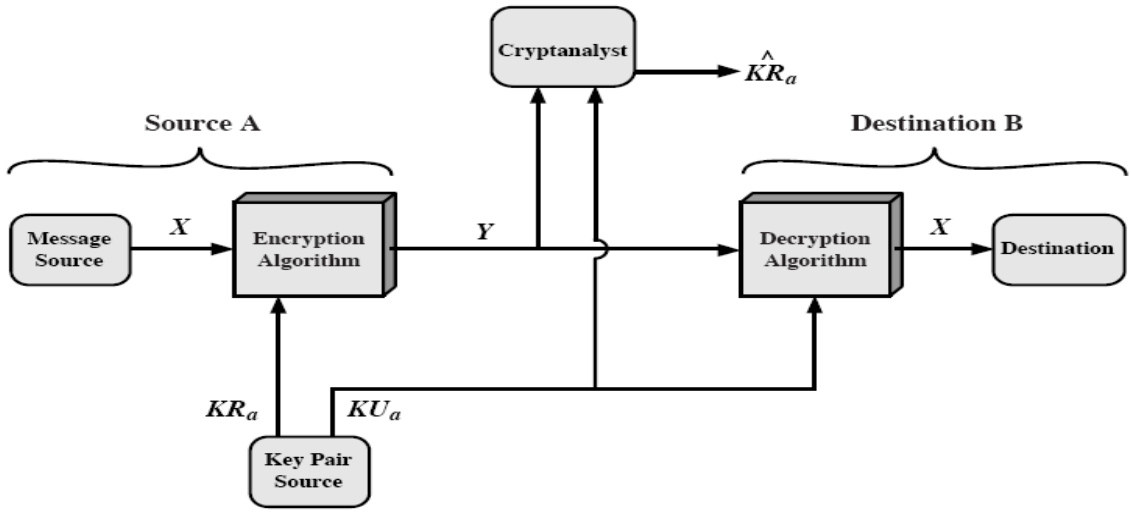


Figure 4-4. Public-key cryptosystem: authentication

In this case, **A** prepares a message to **B**, but he/she encrypts it using his own private-key before transmitting it to **B**, i.e.

$$Y = E_{KRa}(X)$$

Then **B** can decrypt it using **A**'s public key.

$$X = D_{KUa}(Y)$$

Because the message was decrypted with A's public key, therefore, only **A** could have encrypted it. This means the entire encrypted message serves as a "**digital signature**".

It is obvious that the encryption process used when **digital signature** is implemented means that there will be no confidentiality because the public key of the sender is available and can be used by anybody to encrypt the message.

However, if **authenticity of sender** and **secrecy** (**confidentiality) of a message** are required, this can be achieved by double use of public-key scheme, see figure 4-5.

Here, the message **X** is first encrypted using the sender's private key (**signing**), and then the resulted cryptogram **Y** is encrypted using the intended recipient public key. The signed and encrypted message **Z** is then sent over the unsecure channel to the intended destination.

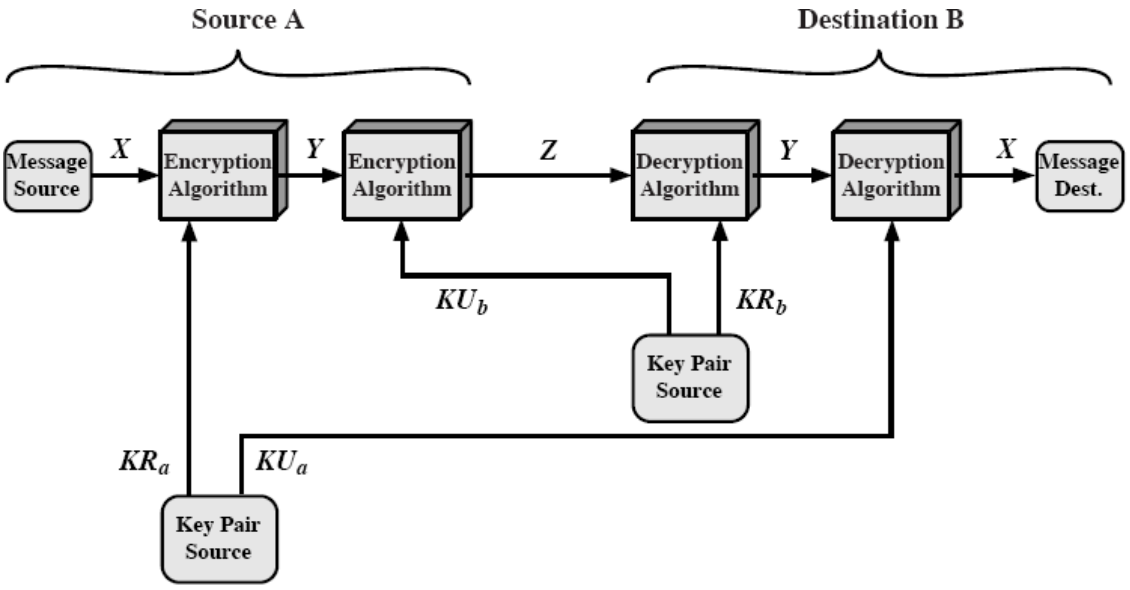


Figure 4-5 Public-key cryptosystem: secrecy and authentication

$$Z = E_{KUb} [E_{KRa} (X)]$$

Upon reception at the destination, the recipient uses his private key to decrypt the received message **Z**, and then use the sender's public key to check the authenticity of the message, i.e.

$$X = D_{KUa} [D_{KRb} (Z)]$$

This process provides digital signature of A and encryption with B's key for secrecy at the sender A side and then decryption at the B's side first then checking the senders signature.

4.5 Applications for public- key cryptosystems

In broad terms, we can classify the use of public-key cryptosystems into three categories:

- **Encryption/decryption:** The sender encrypts a message with the recipient's public key.

- **Digital signature:** the sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.

- **Key exchange:** two sides cooperate to exchange a session key.

Some algorithms are suitable for all three applications, where others can be used only for one or two of these applications, as show in the following table.

Algorithm	**Encryption/ Decryption**	**Digital signature**	**Key exchange**
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

4.6 The RSA Algorithm

The most widely used public-key cryptosystem scheme since year 1978. It is named after three scientists; Ron Rivest, Adi Shamir and Len Adleman (RSA).

- The RSA system is a block cipher in which integer representation for plantext and ciphertext are between **0** and **n-1** for some **n.**

- This system rest upon the computational difficulty involved in factoring very large prim composite integer **n. [**large mean between 100 and 200 bits at the beginning. However, now **n = 1024** bits (or 309 decimal digits) may considered large but more for military applications].

- RSA has proved popular in email communication.

Description of RSA algorithm:

- It uses exponentiation.

- Plaintext encrypted in blocks with binary value **< n.**
  In practice, block size is $\mathbf{2^k}$, where $\mathbf{2^k < n <= 2^{k+1}}$

- If M is the message then ciphertext C is calculated by:
  $\mathbf{C = M^e \bmod n}$

  And M can be recovered by:

  $\mathbf{M = C^d \bmod n}$

  $= (M^e \bmod n)^d \bmod n = (M^e)^d \bmod n$

  $= M^{ed} \bmod n$

Where **n** is known to both sender and receiver,

e known to sender and **d** is known to receiver only.

Thus, this is a public-key encryption/decryption algorithm with public key **KU = {e, n}** and private key **KR = {d, n}.**

Now, we need to find a relationship of the form $\mathbf{M^{ed} = M \bmod n}$.

From Euler's theorem,

Given two prime numbers, p & q, and two integers n & m, such that n = pq and 0 < m < n, and an arbitrary integer k, the following relationship holds:

$$\mathbf{m^{k\phi(n)+1} = m^{k(p-1)(q-1)+1} \equiv m \bmod n}$$

Where ϕ(n) is the Euler tutent function, which is the number of positive integers less than n and relatively prime to n.

It is shown that ϕ(pq) = (p-1)(q-1). Thus we can achieve the desired relationship if

$$\mathbf{e\ d = k\ \phi(n) + 1}$$

This is equivalent to saying

$$\mathbf{e\ d \equiv 1 \bmod \phi(n)}$$

$$\mathbf{d \equiv e^{-1} \bmod \phi(n)}$$

that is e and d are multiplicative inverses mod **ϕ(n)**. It must be noted that this is only true if and only if **e** and **d** are relatively prime to **ϕ(n)**. Equivalently **gcd(ϕ(n), d) =1.**

RSA Design

RSA scheme ingredients are:

- Two large positive prime integers;
  p and **q** are chosen (private, chosen)
- Their product **N=pq** is calculated (public, calculated)
- Now a positive integer **e** is chosen which is prime to **(p-1)(q-1)** or **ϕ(n),** the Euler's tutent function. (public, chosen)
- Fermat's theorem is applied to calculate another integer **d**,
  i.e. $\mathbf{d \equiv e^{-1} \bmod \phi(n)}$ (private, calculated)

[Euclid stated :

"If **e** and **ϕ(n)** satisfy $\mathbf{1 < e < \phi(n)}$ and **gcd(e, ϕ(n)) = 1**, then there is a unique integer **d**, where $\mathbf{1 < d < \phi(n)}$ such that $\mathbf{e\ d \equiv 1 \bmod \phi(n)}$"].

Hence,

The **public key** consists of **{e, n}** and the **private key** consists of {d, n}.

In conclusion, RSA technique involves three main operations; Key generation, Encryption and Decryption as summarized in figure 4-6.

Key Generation

Select p, q	p and q both prime, $p \quad q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; \ 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \bmod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

Decryption

Ciphertext:	C
Plaintext:	$M = C^d \pmod{n}$

Therefore, the message **M** can be encrypted by the sender using the equation $\mathbf{C = M^e \bmod n}$, and decrypted by the receiver using the relation $\mathbf{M = C^d \bmod n.}$

To prove this; we have $\mathbf{e\ d \equiv 1 \bmod \phi(n)}$

Since $\mathbf{C \equiv M^e \bmod n}$

Generally, to encipher a message X, it is first divided into blocks $X_1, X_2, X_3, \ldots, X_m$ at the sender end, and then each block X_i is encrypted by:

$\mathbf{C_i = X_i^e \bmod n,}$ $\mathbf{C_i}$ is ciphertext block.

At the receiver end, it is deciphered by:

$\mathbf{X_i = C_i^e \bmod n}$

4.7 Simple RSA Implementation examples:

Example 1: Select two prime integers; p = 5 and q = 7

Then calculate n = p q = 5 X 7 = 35

Now ϕ(35) = (5-1) (7-1) = 4 X 6 = 24

Then chose d = 11 , which is relative prime to 24 and < 35,

Calculate e using **e d mod 24 = 1,** *i.e.*

e X 11 mod 24 = 1 ➔ therefore e = 11

Let e & n be public key, or [11, 35], and

d & n be the private key [11, 35]

Now take any message (number) M (such that 0 <= M <=24), for example M = 3, calculate the ciphertext C;

$C = M^e \bmod n = 3^{11} \bmod 35 = 12$

C is sent to the receiver on insecure channel, who will convert it back by using the private key [11, 35] in the equation;

$C^d \bmod n = 3^{11} \bmod 35$ ➔ *the result is =* ***3***,

which is the original message.

Example 2:

1- *Select two primes;* ***p = 7*** *and* ***q = 17.***
2- *Calculate* ***n = pq*** *= 7 X 17 = 119.*
3- *Calculate ϕ(n);*

ϕ(n)= *(p-1)(q-1) = 6 X 16 = 96.*

4- *Select* ***e****, relatively prime to* $\phi(n)$ *and* $< \phi(n)$; *Chose* ***e = 5****.*

5- *Calculate* ***d****;*

$$d \equiv e^{-1} \bmod 96 = 5^{-1} \bmod 96$$

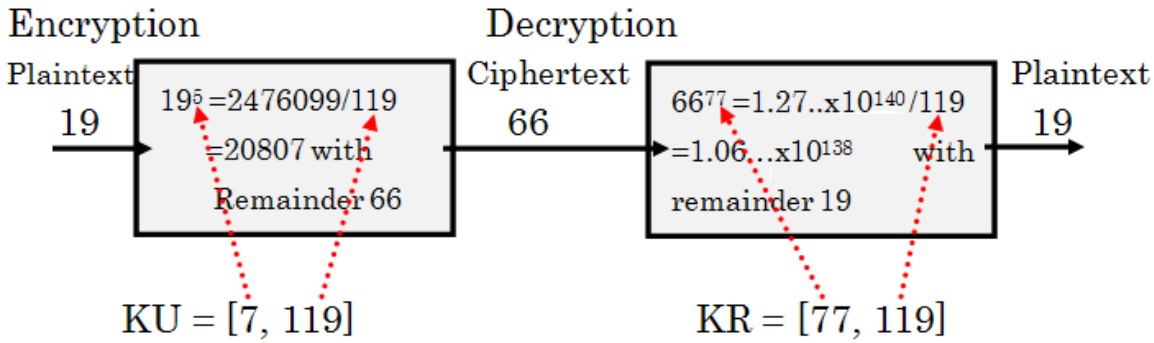
or $d \times 5 \equiv 1 \bmod 96$ ➔ $d = 77$,

(because $5 \times d = k \times 96 + 1$ *or*

$5 \times 77 = k \times 96 + 1$, *i.e.*

$385 = 4 \times 96 + 1$ *or*

$385 = 384 + 1$*).*



Now *using KU = [5, 119] as public key and KR = [77, 119] as private key*

For a message M = 19 for example, one does the following:

6- *For encryption at the sender;*

$$C = 19^5 \bmod 119 = 66 \bmod 119 = 66$$

7- *For decryption at the receiver;*

$$M = C^{77} \bmod 119 = 66^{77} \bmod 119 = 19$$

Example 3:

1- Select two primes; ***p = 17*** *and* ***q = 11.***
2- Calculate ***n = pq*** *= 17 X 11 = 187.*
3- Calculate $\phi(n)$;

$$\phi(n) = (p-1)(q-1) = 16 \text{ X } 10 = 160.$$

4- Select ***e****, relatively prime to $\phi(n)$ and < $\phi(n)$; Chose* ***e = 7****.*
5- Calculate ***d****;*

$$d \equiv e^{-1} \bmod \phi(n) = 7^{-1} \bmod 160$$

or $d \text{ X } 7 \equiv \bmod 160 \rightarrow d = 23$

Now *using {7, 187} as public key and {23, 187} as private key for a message M = 88 for example, one does the following:*

6- For encryption at the sender;

$$C = 88^7 \bmod 187 = 11$$

7- For decryption at the receiver

$$M = C^{23} \bmod 187$$

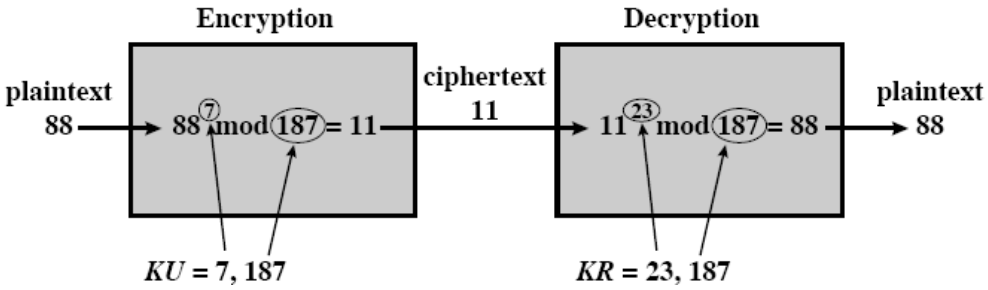
$$= 11^{23} \bmod 187$$

$$= [11^1 \bmod 187][11^2 \bmod 187][11^4 \bmod 187][11^8 \bmod 187]$$

$$= [(11)(121)(55)(33)] \bmod 187$$

$$= [79 \text{ X } 720 \text{ X } 245] \bmod 187$$

$$= 88$$


Encryption
Decryption
plaintext
88
88 7 mod 187 = 11
ciphertext
11
11 23 mod 187 = 88
plaintext
88
KU = 7, 187
KR = 23, 187

4.8 Mini RSA:

To demonstrate RSA algorithm on a mini – case, take the following example.

Let p = 59 and q = 89 [both integers]

And let d = 3403 [also integer]

Now

Calculate n = pq = 59 *89 = 5251

Calculate $\phi(n)$

= (p-1)(q-1) = 58 * 88 = 5104

Select d = 3403, calculate e by: e d = 1 mod $\phi(n)$ or

E * 3403 mod 5104 = 1

The result is e = 3.

Now if the following character set is used:

00	A	10	K	20	U	30	4	40	>
01	B	11	L	21	V	31	5	41	$
02	C	12	M	22	W	32	6	42	+
03	D	13	N	23	X	33	7	43	/
04	E	14	O	24	Y	34	8	44	%
05	F	15	P	25	Z	35	9	45	(

06	G	16	Q	26	0	36	?	47	)
07	H	17	R	27	1	37	.		
08	I	18	S	28	2	38	SPACE		
09	J	19	T	29	3	39	<		

Every two characters are mixed per block [which is simplistic].

Then for the message :

"**Quoth the raven never more**" without spaces, then the message is segmented into 11 blocks, each of two characters. And encryption is achieved using the equation:

$C_i = M_i^3 \bmod 5251$

(11 block $M_1, M_2, \ldots, M_{11}$) as shown below.

And decryption using the equation:

$M_i = C_i^{3403} \bmod 5251,$

Plaintext	Q	U	O	T	H	T	H	E	R	A	V	E	N	N	E	V	E	R	M	O	R	E
	16	20	14	19	07	19	07	04	17	00	21	04	13	13	04	21	04	17	12	14	17	04
block	1620		1419		0719		0704		1700		2104		1313		0421		0417		1214		1704	
ciphertext	3340		0676		2924		0467		1619		1853		1723		1751		0654		4612		1663	

decryption	1620		1419		0719		0704		1700		2104		1313		0421		0417		1214		1704	
	16	20	14	19	07	19	07	04	17	00	21	04	13	13	04	21	04	17	12	14	17	04
Plaintext	Q	U	O	T	H	T	H	E	R	A	V	E	N	N	E	V	E	R	M	O	R	E

4. 9 Computational Aspects

For computer application of RSA, digital representation of messages and keys is implemented. In this context, there are two important issues; i.e.

- Encryption and Decryption.
- Key generation.

Encryption/Decryption

The results of calculation are reduced to practical values because of **modulo n**. Therefore, for both of the above aspects, exponentiation is used. (Square and Multiply).

Generally, suppose we wish to find value of $\mathbf{a^m}$, with **a** and **m** are *positive integers*. If **m** is binary number $b_k, b_{k-1}, b_{k-2}, \ldots, b_0$, then

$$m = \sum 2^i \qquad , b_0 \neq 0$$

$$a^m = a^{(\Sigma 2^{\wedge} i)} = \Pi\, a^{(2^{\wedge} i)} \qquad , b_0 \neq 0$$

Therefore, $a^m \bmod n = (\Pi\, a^{(2^{\wedge} i)}) \bmod n, \qquad b_0 \neq 0$

$$= [\Pi\, a^{(2^{\wedge} i)} \bmod n] \bmod n$$

The algorithm is shown below for **a^b mod n.**

```
c ← 0; d ← 1
for i ← k downto 0
    do  c ← 2 × c
        d ← (d × d) mod n
        if  bi = 1
            then  c ← c +
```

Example: Employ the above algorithm for a = 7, b = 560 and n = 561

Solution:

We can represent b in binary as b = 560 = 1000110000, therefore k = 9.

The values of **i, b_i, c** and **d** in the algorithm would be as shown in the table below.

I	9	8	7	6	5	4	3	2	1	0

Bi	1	0	0	0	1	1	0	0	0	0
C	1	2	4	8	17	35	70	140	280	560
D	7	49	157	526	160	241	298	166	67	1

The results of the fast modular exponentiation algorithm for

$a^m \bmod n$, i.e. for

Therefore $d = 7^{560} \bmod 561 = = > 1$

(The variable c is included for explanatory purposes).

Key generation

Each user must generate a pair of keys. It involves

Select 2 prime numbers **p** and **q** = = > **n = pq.**

Selecting either **e** or **d** and then calculate the other.

The problem is mainly finding a large prime integer. It must be noted that there is no short cut, but there are few available algorithm, such as *Miller – Robin algorithm*. It is a probabilistic method that is characterized by selecting an odd number and testing its' primarily.

Information Security2

Computer Science

4th Class

Chapter ٥

MESSAGE AUTHENTICATION AND HASH FUNCTION

5.1 Message Authentication:

Message authentication is concerned with the following issues:

- protecting the integrity of a message.
- validating identity of originator.
- non- repudiation of origin (dispute resolution).

This section will consider the security requirements first. Then three generally used alternative functions, i.e.

- message encryption
- message authentication code (MAC)
- hash function

5.2 Message Authentication Requirements

- **Attacks:** In any communication across a network, the attacks that can be identified are:
  - **Disclosure:** release of message contents to any unauthorized person.
  - **Traffic analysis**: discovery of pattern of traffic between parties.
  - **Masquerade**: insertion of message into a network by a fraudulent source.
  - **Content modification**: altering the content of a message (includes deletion and transposition and changing).
  - **Sequence modification**: altering the sequence of a message.
  - **Timing modification**: delay or replay of a message.

- **Repudiation:**

1. Source repudiation: denial of transmission of a message by source.
2. Destination repudiation: denial of transmission of a message by destination.

Measures to withstand the above attacks are:

- **Message confidentiality: it includes:**
  - Disclosure
  - Traffic analysis.
- **Message Authentication: it includes:**
  - Masquerade.
  - Content modification.
  - Sequence modification.
  - Timing modification.
- **Digital signature: it includes:**
  - Repudiation.

Authentication Mechanism:

- There two fundamental levels for message authentication or digital signature:
  - Lower-level:
    A function that produce an authenticator; a value to be used to authenticate a message.
  - Higher-level:
    The lower-level function is used as primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message.

5.3 Authentication Functions:

- **Message encryption:**
  - The ciphertext of the entire message serves as its authenticator.

- **Message Authentication Code** (MAC):
  - A public function of the message and a secret key that produces a fixed length value that serves as the authenticator.
- **Hash function:**

- A public function that maps a me4sage of any length into a fixed-length hash value, which serves as the authentication.

(a) <u>Message encryption:</u> There are two techniques:

1- Conventional (Symmetric) Encryption.
2- Public-Key Encryption.

<u>Conventional Encryption.</u>

Having a sender A and a receiver B in a transmission system, confidentiality can be used for message authentication as shown in figure 5-1a. Here a secret key, K is used for both encryption and decryption.

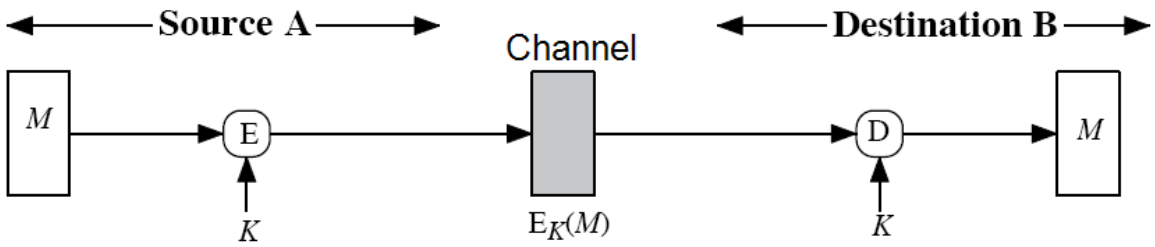


Fig. 5-1a. Symmetric encryption: confidentiality and authentication

- If **symmetric encryption** is used then: $A \rightarrow B: E_K[M]$
  - Provide **confidentiality**:
    - Receiver B knows the sender A must have created it because they share the key K.
  - Degree of **Authentication**: They know;
    - Content could only come from A.
    - Content has not been altered, if message has suitable structure, redundancy or a checksum to detect any changes.
  - Does not provide **signature**:
    - Receiver could forge message.
    - Sender can deny a message.

<u>Public-Key Encryption:</u>

For each user in this system, two keys are used, a private key KR and a public key KU, as shown in figure 5-2b-d.

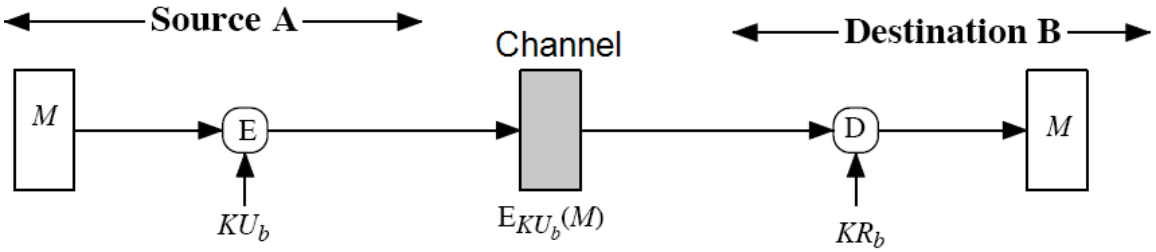


Fig. 5-2b. Public-key encryption: confidentiality.

$A \rightarrow B: E_{KU_b}[M]$

- Provides confidentiality
  - —Only B has KR_b to decrypt
- Provides no authentication
  - —Any party could use KU_b to encrypt message and claim to be A

(b) Public-key encryption: confidentiality

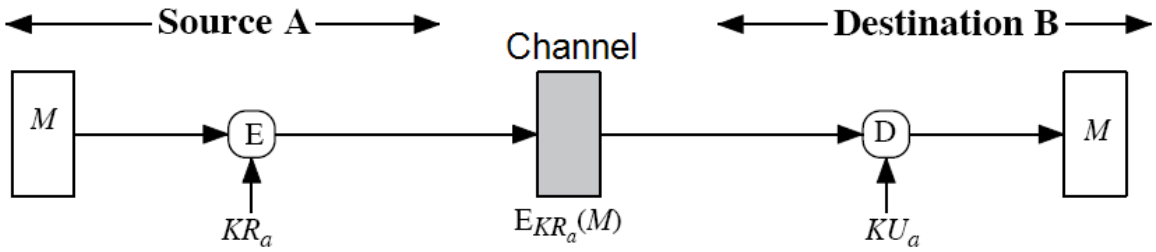


Fig. 5-2c. Public-key encryption: authentication and signature.

$A \rightarrow B: E_{KR_a}[M]$

- Provides authentication and signature
  - —Only A has KR_a to encrypt
  - —Has not been altered in transit
  - —Requires some formatting/redundancy
  - —Any party can use KU_a to verify signature

(c) Public-key encryption: authentication and signature

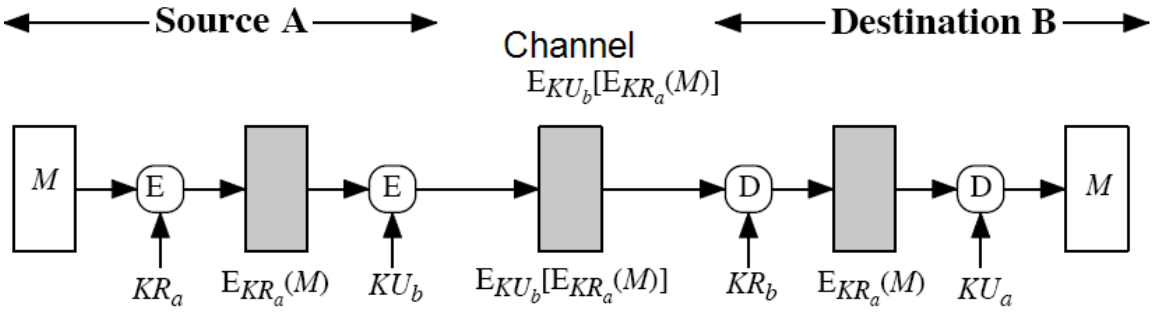


Fig. 5-2d. Public-key encryption: confidentiality, authentication and signature.

$A \rightarrow B: E_{KU_b}\left[E_{KR_a}(M)\right]$

•Provides confidentiality because of KU_b

•Provides authentication and signature because of Kr_a

(d) Public-key encryption: confidentiality, authentication, and signature

- If **public-key encryption** is used:
  - Encryption provides no **confidence** of sender.
  - Since anyone potentially knows public-key.
  - However if
    - sender **signs** message using their private-key.
    - then **encrypts** with recipients public key.
    - have both **secrecy** and **authentication**.
  - Again need to recognize corrupted messages.
  - But at cost of two public-key uses on message.

(b) Message Authentication Code (MAC):

- It involves the use of a secret key to generate a small fixed-size block of data, known as a cryptograph **checksum** or **MAC** that is appended to the message.

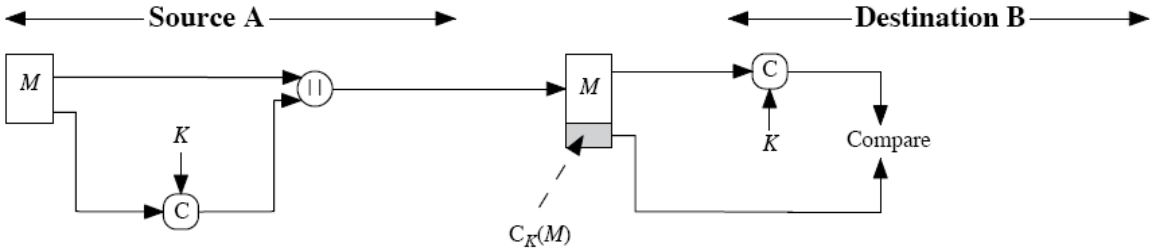
- This technique assumes that two communicating parties, say A and B, share a common secret key K.

- A **MAC** function is similar to encryption. One difference is that the MAC algorithm need not be reversible, as it must for decryption which makes it less vulnerable of being broken than encryption.

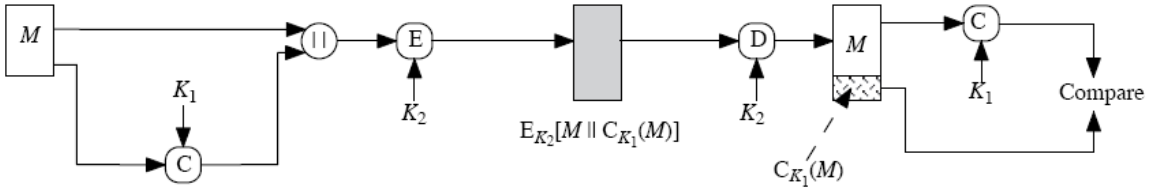
- **Authentication functions:**

  There are three levels of authentication using **MAC**, as shown in figures 5-3 a-c, i.e.

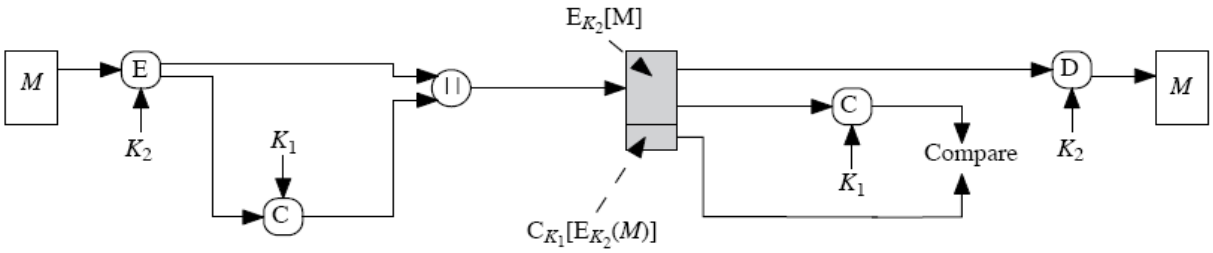
- Message authentication
- Message authentication and confidentiality; authentication tied to plaintext.
- Message authentication and confidentiality; authentication tied to ciphertext



(a) Message authentication.



(b) Message authentication and confidentiality; authentication tied to plaintext



(c) Message authentication and confidentiality; authentication tied to ciphertext

Fig 5-3. Basic Uses of Message Authentication Code (MAC)

❖ From the above figures, we **<u>can summarize</u>** as follows:

- MAC provides confidentiality
- It can also use encryption for secrecy
  - generally use separate keys for each
  - can compute MAC either before or after encryption
  - is generally regarded as better done before

- why use a MAC?
  - sometimes only authentication is needed
  - sometimes need authentication to persist longer than the encryption (e.g. archival use)

- Note that a MAC is not a digital signature.

❖ **MAC Properties**

- A MAC is a cryptographic checksum that is appended to the message, **MAC** = $C_K(M)$
  - condenses a variable-length message **M**
  - using a secret key K
  - to a fixed-sized block (authenticator)

- It is a many-to-one function
  - potentially many messages have same MAC
  - but finding these needs to be very difficult

- MAC function is similar to encryption, but it need not be reversible as it must for decryption.

❖ **Requirements for MACs**

- taking into account the types of attacks
- need the MAC to satisfy the following:
  - knowing a message and MAC, is infeasible to find another message with same MAC
  - MACs should be uniformly distributed
  - MAC should depend equally on all bits of the message

(c) Hash Functions

- The one-way hash function is a variation of message Authentication Code.

- It accepts a variable size message **M** and produces a fixed-size hash code **H(M),** message digest, as an output, i.e. condenses arbitrary message to fixed size.

- Usually assume that the hash function is public and not keyed
  - Unlike MAC which is keyed.

- Hash used to detect changes to message.

- It can use in various ways with message.

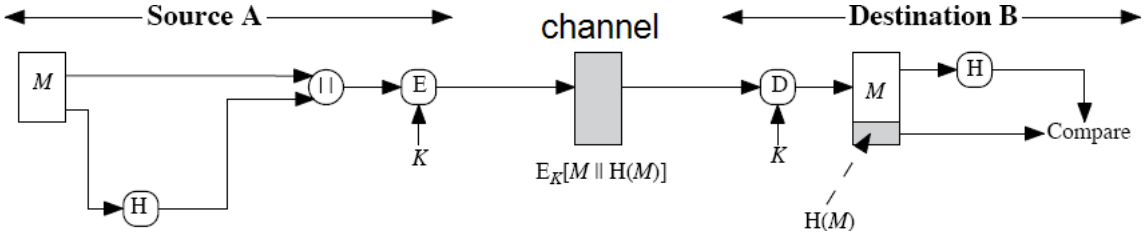
- Most often it is used to create a digital signature.

- A simple example of hash function is based on **XOR** of message blocks.

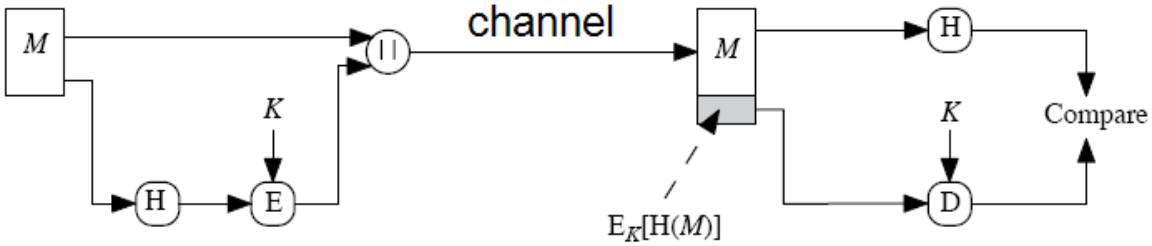
  - **<u>Hash Function application types:</u>** There are many type of usage for the hash function technique, such as:

- Message plus concatenated hash code is encrypted using conventional encryption.
- Only a hash code is encrypted using conventional encryption
- Only hash code is encrypted using public-key encryption using senders' private key.
- Message plus the public-key encrypted hash code can be encrypted using a conventional secret key.

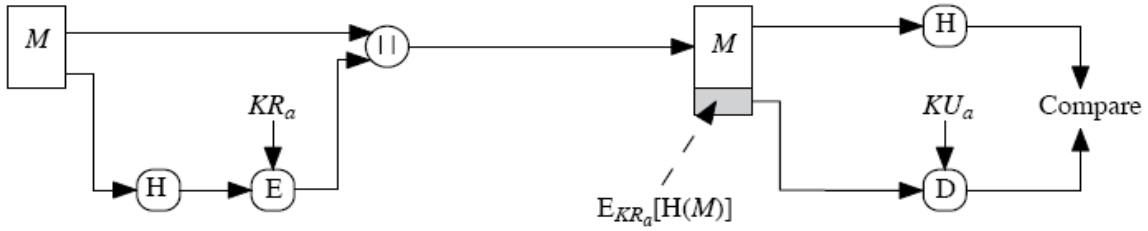
These type are illustrated in figure 5-4.



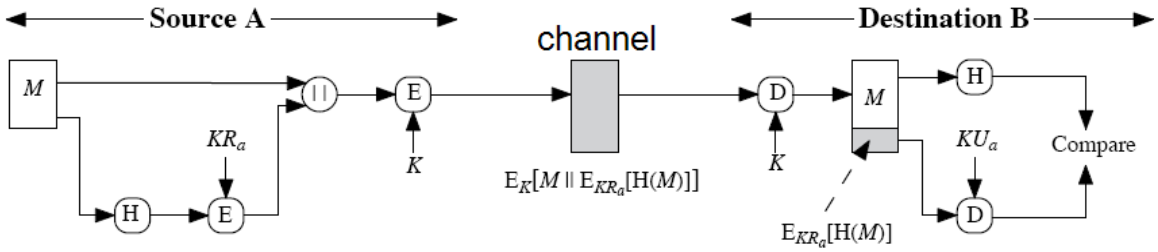
(a) Message + hash code are encrypted using symmetric encryption.



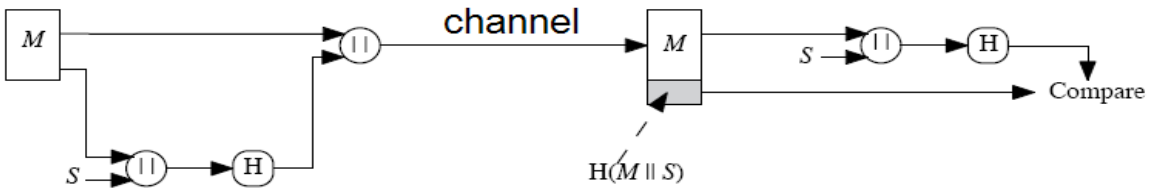
(a) Only hash code is encrypted using symmetric encryption.



(c) Only hash code is encrypted using public-key encryption (using sender's private key).

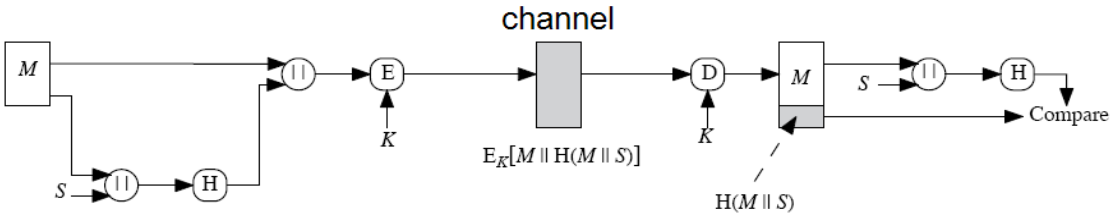


(d) Confidentiality and digital signature. Message + public-key encrypted hash code with conventional secret key.



(e) Hash (authentication);

- Sender and receiver share a common secret key **S**.
- Because **S** is not sent by itself, an opponent cannot modify an intercepted message and cannot generate a false message, also (But no confidentiality).



(f) Hash (authentication); only sender and receiver share a common secret key S. It provides confidentiality.

Chapter ٦

ACCESS CONTROL

There are two primary methods for access control, they are:

(1) System Access Controls and (2) Data Access Controls.

6.1 System Access Control

Simply it ensures that unauthorized users do not get any access to the system. It is concerned with the problem identification and authentication

Identification and Authentication:

Identification: It is the way you tell the system who you are.

Authentication: Is the way you prove to the system that you are who you say you are.

❖ Generally, there are three classic ways in which you can prove yourself. This is achieved by tell the system:

1- **Something you know**: the most familiar example is a password. The theory is that if you know the secret password for an account, you must be the owner of that account.

2- **Something you have**: examples are the keys, tokens and smart carts you must have to "unlock" your terminal or your account. The problem with this theory is that you might loose this key, it might be stolen from you or someone might borrow it and duplicate it.
With the automatic teller machines (ATMs), the people are becoming increasingly familiar with this type of authentication.

3- **Something you are**: Examples are physiological or behavioral traits, such as your fingerprints, handprints, retina pattern, voice, signature, photos or …. Biometric systems can compare your particular trait against the one stored for you and determine whether you are who you claim to be or not.

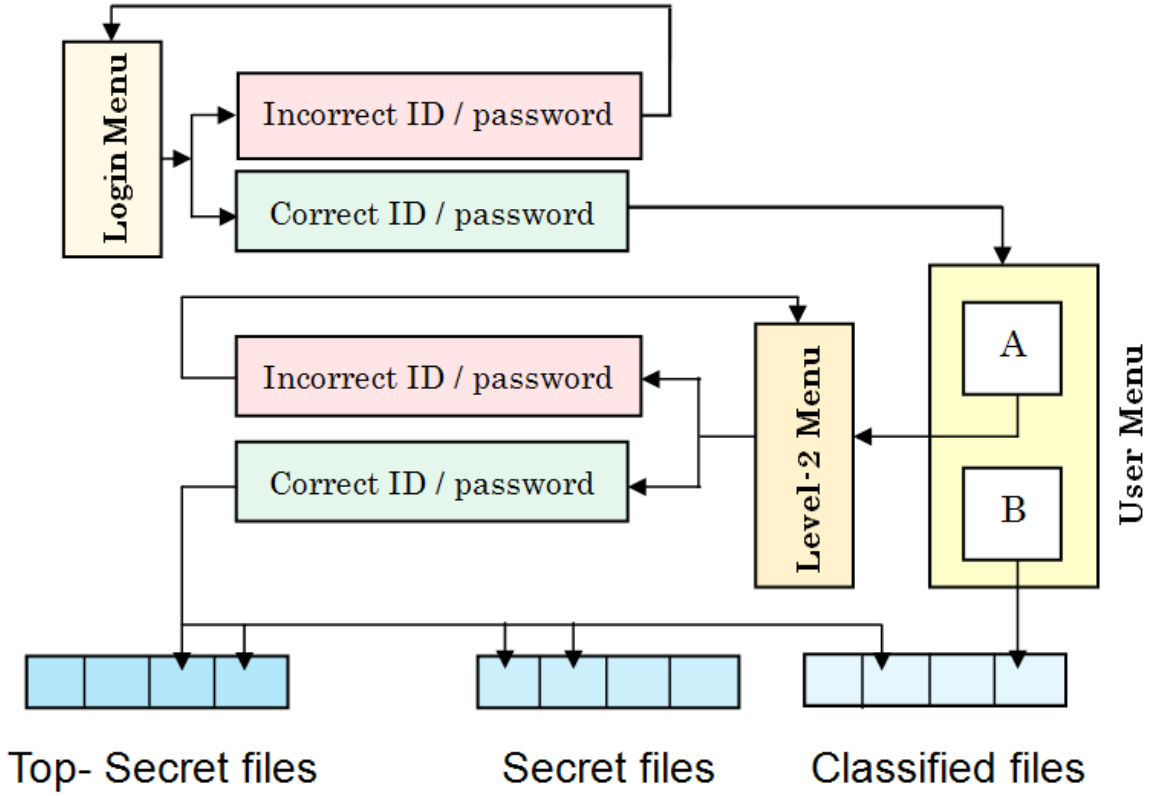
❖ The identifier (**ID**) is typically a unique *name initials*, a *login number*, or *an account number* assigned by the system administrator based on your

own name and/or group. Login identification sequence are much the same from system to another, for example: **Login:**

- The **password** is typically a string of letters and/or numbers known only to you. The typical password interaction is usually a relatively simple and user-friendly one. After you enter your login ID, the system prompts:

  Password:

- ***Usually Identification ID and passwords are used possibly with certain hierarchy. An example is shown below for a hierarchical data base structure.***



- This example shows two hypothetical user levels and three security classification levels (classified, secret and top-secret files).
  - A *correct login*: Each authorized user can view classified files, [user B].
  - But a *higher authority user A*, may also work with secret and top-secret files by passing successfully another menu, the ID may be

particular to the security level sought and the password should differ from the login password.

- Read / write privileges would be also sub-divided to be account for. Privileges tend to be related to positions in private sectors; however, they are more related to responsibilities rather than ranks in military and intelligence communities.
  This is called **"NEED TO KNOW PRINCIPLE".**

6.2 ***Hints for protecting passwords:***

Both system administrator and users share responsibility for enforcing password security. Remember password security is every one's responsibility. In addition to damaging your own files, some one who uses your password to break a system can also compromise all of the files in your system or network.
Remember,

"A password should be like toothbrush;
Use it everyday, change it regularly and
Do not share it with friends"

Notes:

1- Don't allow any login without password. If you are the system administrator, make sure every account has a password.

2- Do not keep passwords that may have come with your system, change all test or guest passwords.

3- Do not ever let any one use your password.

4- Do not write your password down, particularly on your terminal, computer or anywhere around your desk.

5- Do not type a password while anyone is watching.

6- Do not make a bad situation worse. If you do share your password, change it immediately (or ask your administrator to change it).

7- Do not record your password online or send it anywhere via email. The intruder scans email messages for references to the word "password".

8- Don't keep the same password indefinitely. Even if your password hasn't been compromised, change it on a regular basis.

6.2.1 Protecting passwords:

Access decisions are the heart of system security and access decisions are based on passwords. So, it is vital that your system protects its passwords and other login information. Most systems protect passwords in two important ways:

1- They make passwords hard to guess and login controls hard to crack (protecting your login and password on entry).

2- They protect the file in which passwords are stored.

6.2.2 Making passwords and login hard to guess and crack:

Most venders offer a login controls and password management features that the system administrator can mix and match to provide optimal protection of a particular system. Because these security features are commercially attractive and easy to implement, most systems tend to have a lot of them.
Examples of such features are:

1- System messages: Most systems display welcome announcement messages before or after you successfully login.

2- Limited attempts: After a certain number of unsuccessful tries at logging into the system (the number can be specified by the system administrator), the system locks you out and prevents you from attempting to login from that terminal.

3- Limited time periods: Certain users or terminals may be limited to

logging in during business hours or there is a specified time.

4- Last login message: When you login, the system may display the date and time of your last login. Many systems also display the number of unsuccessful login. This may give you a chance to discover that your account was accessed by some one else.

5- User – changeable password: In many systems, you are allowed to change your own password at any time after its initial assignment by the system administrator.

6- System – generated password: Some systems require to use passwords generated randomly by the system, rather than relying on your own selection of difficult – guess passwords.

7- Password aging and expiration: When a specified time is reached - for example, the end of the month – all passwords in the system expire. The new passwords usually must not be identical to the old passwords.

8- Minimum length: Because short passwords are easier to guess than long ones, some systems require that password be of a certain length, usually six to eight characters.

9- Password locks: Locks allow the system administrator to restrict certain users from logging in or to lock login accounts that have not been used for an extended period of time.

10- System passwords: System passwords control access to particular terminals that may be target for unauthorized use. Usually a system password must be entered before you enter your individual password.

11- Primary and secondary passwords: Some systems require that two users, each with a valid password, be present to login successfully to certain extremely sensitive accounts.

6.2.3 Protecting you password in storage:

Every system needs to maintain its authentication data. Typically, valid passwords are stored in a password file. This file typically is accessed only under certain limited circumstances:

1- When a new user is registered.
2- When you change your password.
3- When you login and need to be authenticated.

Systems commonly use both (a) Encryption and (b) Access controls, to protect password data.

Encryption:

Most systems perform one – way encryption of password. One – way encryption means that the password is never decrypted. When the system administrator supplies you with your initial password, it is encrypted before it is stored in the password file. Each time you login and enter your password, the system encrypts the password and compares the encrypted version with the stored encrypted in the password file in order to make sure you have entered a valid password. Remember too that the password is never displayed on the terminal screen.

6.2.4 Hint for picking passwords

If you are allowed to choose your own password, pick passwords that are hard to guess. Here are some suggestions:

1- Pick passwords that are not words or names.
2- Pick a mix of alphabetic and number characters.
3- Pick long passwords.
4- Pick different passwords for different machine or network nodes you access.
5- Be careful about including special characters in the passwords, some special characters (e.g. # and @ may have special meanings to emulate software).

- The best passwords contain mixed uppercase and lowercase letters, as well as at least one number and/or special characters.

6- Combine several short words with numbers or special characters, for example, I;did3it.
7- Add a number or special character for some security, for example: Onif;tdi or On5iftdi.
8- Pick a nonsence word that is still pronounceable, for example: 8Bektaq or Shamoaz12.

6.3 Access control:

Even encrypted passwords might be liable to be cracked by determined foe. Many systems store encrypted password data in files known as shadow-password files, which have the most restrictive protection available in the system. In most systems, access is limited to the system administrator, usually by specifying only the administrator ID in an access control list (ACL) on the file.

6.4 Data Access:

Protecting your data

There are three basic types of access controls that provide different levels of protection to the files in your system:

1- Discretionary Access Control (DAC).
2- Mandatory Access Control (MAC).
3- Role Based Access Control (RBAC)

- With **DAC** you decide how you want to protect your files and whether to share your data or not.

- With **MAC**, the system protects your files, in MAC system, everything has a label. Using the security policy relationships established for your organization, the system decides whether a user can access a file by comparing the label of the user with the label of the file or not.

- With **RBAC**, the access control framework should provide security administrators with the ability to determine who can perform what actions, when, from where, in what order, and in some cases under what relational circumstances.

Chapter ٧

VIRUSES and OTHER MALICIOUS CONTENT

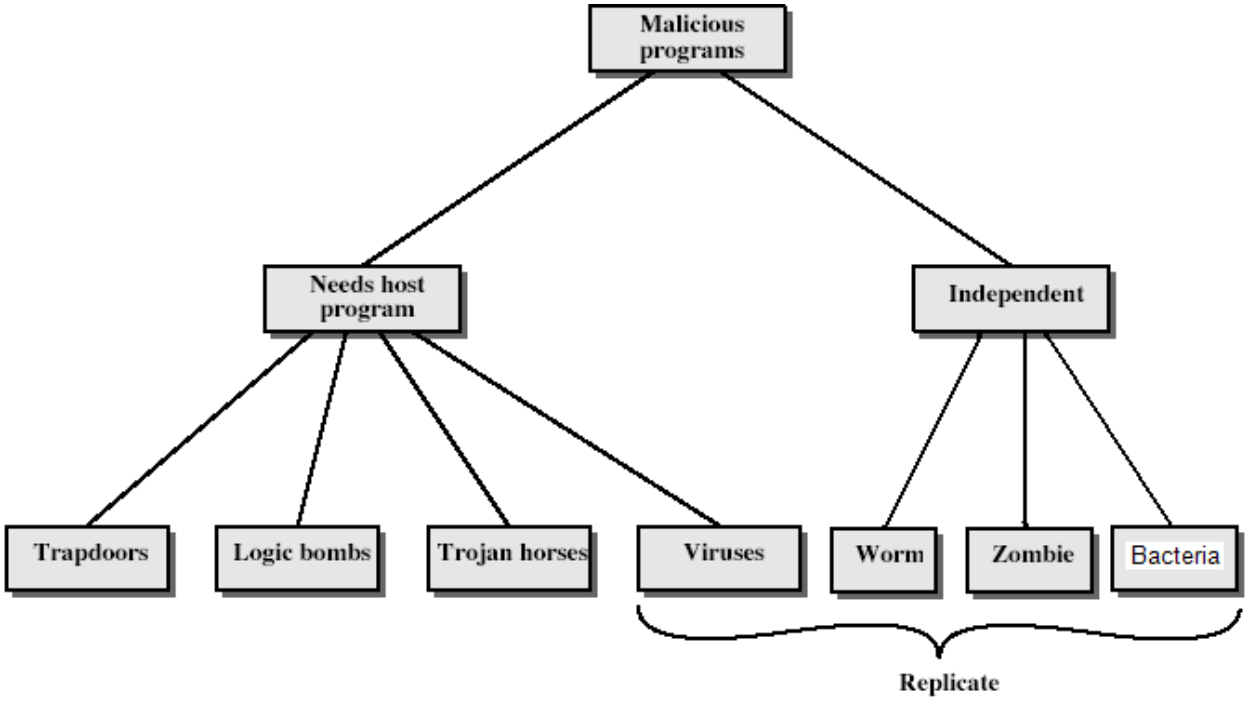
7.1 Introduction

- computer viruses have got a lot of publicity
- one of a family of **malicious software**
- effects usually obvious
- have figured in news reports, fiction, movies (often exaggerated)
- getting more attention than deserve
- are a concern though

The figure below provides an overall software threat for malicious programs. Theses threats can be divided in to two categories:

1- Those that need a host program.
2- Those, that are independent.

7.2 Malicious Software



- **The former are essentially fragments of programs that can not exist independently of some actual application program, utility or system programs.**
- **The latter are self-contained programs that can be scheduled and run by the operating system.**

- **We can also differentiate between those software threats that do not replicate and those that do.**

- **Replicate: programs (worm and bacterium) that, when executed, may produce one or more copies of itself to be activated later on the same system or some other system.**

7.3 Trapdoor

- Secret entry point into a program that allows those who know access bypassing usual security procedures without going through the usual security access.
- It has been commonly used by developers

- a threat when left in production programs allowing to be exploited by attackers
- It is very hard to be blocked in O/S
- It requires good s/w development & update

7.4 Logic Bomb

- It is one of oldest types of malicious software threats.
- It is code embedded in legitimate program that is set to explode when certain conditions are met. i.e.
- activated when specified conditions met, e.g.
  - presence / absence of some file
  - particular date / time
  - particular user
- when triggered, they damage the system;
  typical damage:- modify
  - delete files / disks

7.5 Trojan Horse

- It is a useful program or commercial procedure containing hidden side-effects.
- which is usually superficially attractive, such as:
  - games, s/w upgrades etc.
- when run, it performs some additional tasks
  - allows attacker to indirectly gain access they do not have directly
- often used to propagate a virus / worm or install a backdoor
- or simply to destroy data.

7.6 Zombie

- It is a program which secretly takes over another networked computer.
- Then it uses it to indirectly launch attacks
- Often it is used to launch distributed denial of service (DDoS) attacks.

- It exploits known flaws in network systems.

7.7 Bacteria

- Programs that do not explicitly damage any files.
- Their sole purpose is to replicate themselves.
- A typical bacteria program may do nothing more than execute two copies of itself simultaneously.
- Or perhaps create two new files each of which is a copy of original source file of the bacteria program.
- It may take up all the processing capacity, memory or disk space.
- denying users access to those resources.

7.8 Viruses

- A program that can a piece of self-replicating code attached to some other code and can infect other programs;
  - cf biological virus.
- Both it propagates itself & carries a payload, i.e.
  - carries code to make copies of itself.
  - as well as code to perform some covert task.

Virus Operation:

Viruses can do any thing that other programs do, the only difference is that it attaches itself to another programs and executes secretly when the host program is run. Once a virus is executing, it can perform any function, such as erasing files and programs. They have the following phases:

- Dormant phase – waiting on trigger event; the virus is idle. It will be activated by some event, such as date, presence of another program or file, capacity of disk exceeding certain limit, etc.

- Propagation phase – replicating to programs/disks; the virus places an identical copy of itself into another program or into certain system area on the disk.

- Triggering phase – by event to execute payload; the virus is activated to perform the function for which it was intended.

- Execution phase – of payload; 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.

Note: Viruses are usually machine / OS specific

– exploiting features and weaknesses of the systems.

Virus Structure

```
program V :=
    {goto main;
    1234567;
    subroutine infect-executable := {loop:
                    file := get-random-executable-file;
                    if (first-line-of-file = 1234567) then goto loop
                    else prepend V to file; }
    subroutine do-damage :=         {whatever damage is to be done}
    subroutine trigger-pulled :=    {return true if some condition holds}
    main: main-program := {infect-executable;
                                    if trigger-pulled then do-damage;
                                    goto next;}
    next:
}
```

Note:

1- The first line of the code is a jump to main virus program,
2- The second line is a special marker that is used by the virus to determine whether or not a victim program has already been infected with this virus.
3- When the program is invoked, control is immediately transferred to the main virus program.
4- The virus program first seeks out uninfected executable file to

infect them.

5- The virus may perform some action, usually to the system. The action could be performed every time the program is invoked or it could be a logic bomb that triggers only under certain conditions.

Types of Viruses

- can classify on basis of how they attack
- parasitic virus
- memory-resident virus
- boot sector virus
- stealth
- polymorphic virus
- macro virus

7.9 Macro Virus

1- It is a platform independent, all of the macro viruses infect software word document. Any how, platform and OS that supports word can be infected.

2- A macro virus infects documents, not executable portions of codes.

3- Macro viruses are easily spread. A very common method is by electronic mail.

In summery**:**

- macro code attached to some data file
- interpreted by program using file
  - eg Word/Excel macros
  - esp. using auto command & command macros
- code is now platform independent
- is a major source of new viral infections
- blurs distinction between data and program files making task of detection much harder
- classic trade-off: "ease of use" vs "security".

In recent years, according to the National Computer Security Agency, macro viruses now make up two-third of computer viruses.

7.10 Email Virus

- spread using email with attachment containing a macro virus
  - cf Melissa
- triggered when user opens attachment
- or worse even when mail viewed by using scripting features in mail agent
- usually targeted at Microsoft Outlook mail agent & Word/Excel documents

7.11 Worms

Network worm programs use network connections to spread from system to system. Once active with a system, a network worm can behave as a computer virus or bacteria or it could implant Trojan horse programs or perform any number of destructive actions.

- replicating but not infecting program
- typically spreads over a network
  - cf Morris Internet Worm in 1988
  - led to creation of CERTs
- using users distributed privileges or by exploiting system vulnerabilities
- widely used by hackers to create **zombie PC's**, subsequently used for further attacks, esp DoS
- major issue is lack of security of permanently connected systems, esp PC's

Worm Operation

- worm phases like those of viruses:
  - dormant
  - propagation
    - search for other systems to infect
    - establish connection to target remote system
    - replicate self onto remote system
  - triggering
  - execution

Morris Worm

- best known classic worm
- released by Robert Morris in 1988
- targeted Unix systems
- using several propagation techniques
  - simple password cracking of local pw file
  - exploit bug in finger daemon
  - exploit debug trapdoor in sendmail daemon
- if any attack succeeds then replicated self

Recent Worm Attacks

- new spate of attacks from mid-2001
- **Code Red**
  - exploited bug in MS IIS to penetrate & spread
  - probes random IPs for systems running IIS
  - had trigger time for denial-of-service attack
  - 2nd wave infected 360000 servers in 14 hours
- **Code Red 2**
  - had backdoor installed to allow remote control
- **Nimda**
  - used multiple infection mechanisms
    - email, shares, web client, IIS, Code Red 2 backdoor

7.12 Anti-Virus Software

- **first-generation**
  - scanner uses virus signature to identify virus
  - or change in length of programs
- **second-generation**
  - uses heuristic rules to spot viral infection
  - or uses program checksums to spot changes
- **third-generation**
  - memory-resident programs identify virus by actions

- **fourth-generation**
  - packages with a variety of antivirus techniques
  - eg scanning & activity traps, access-controls

Advanced Anti-Virus Techniques

- generic decryption
  - use CPU simulator to check program signature & behavior before actually running it
- digital immune system (IBM)
  - general purpose emulation & virus detection
  - any virus entering org is captured, analyzed, detection/shielding created for it, removed

Behavior-Blocking Software

- integrated with host O/S
- monitors program behavior in real-time
  - eg file access, disk format, executable mods, system settings changes, network access
- for possibly malicious actions
  - if detected can block, terminate, or seek ok
- has advantage over scanners
- but malicious code runs before detection

Security (2):4th Class

WHAT IS CYBER SECURITY?

Cybersecurity is the practice of protecting internet-connected systems of hardware, software, and data, from threats. These threats range from ransomware and data theft to phishing scams. Cybersecurity encompasses everything from keeping sensitive information safe to making sure IT systems work properly.

Effective cybersecurity protection involves a combination of physical security measures, software tools like firewalls and antivirus programs, and organizational policies that protect data privacy and prevent data loss or theft.

The Difference Between Cybersecurity and Information Security

Considering definitions provided by these basic sources, it can be concluded that Information Security fully includes Cybersecurity as one of its components. Cyber Security, on the other hand, is responsible to ensure the security of information against cyber threats and cyber-attacks while it is processed, stored, or transported. Access Controls, Procedural Controls, Compliance Controls, and Technical Controls are examples of Information Security, whereas Application Security, Network Security, Cloud Security, and Critical Infrastructure are examples of Cybersecurity. An example to compare Cybersecurity and Information Security is when sensitive information is left on the desk of an employee and copied by a customer aiming to sell it to an unauthorized party. This is a case of an Information Security breach since Cyberspace is not involved in the process. However, if this sensitive information was shared on social media by the employee hurting the reputation of the company, it was considered a breach in Cybersecurity as well as Information Security. Thus, Cybersecurity incidents can be generalized to information security leading to breaches in confidentiality, integrity, or availability of information and exposing an organization to the threat of information loss. The difference between Cybersecurity and Information Security is represented in Figure 1.

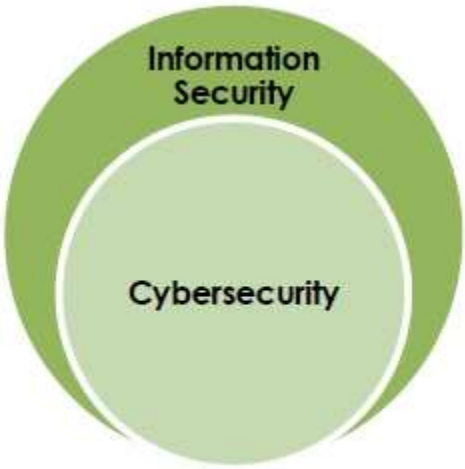


Fig 1. Difference between Cybersecurity and Information Security

Considering the differences between Cybersecurity and Information Security from different aspects, Cybersecurity protects cyberspace from cyber-attacks while Information Security considers protecting information from any form of threat regardless of being digital or physical. Thus, the scope of Cybersecurity is limited to cyberspace and Information Security deals with data protection in a wider realm. In terms of threats, Cybersecurity provides protection against dangers in the digital environment while Information Security deals with threats that endanger information regardless of their type. Attacks that endanger information in cyberspace include cyber frauds, cybercrime, and law enforcement; however, any type of unauthorized access to information, disruption, or information disclosure is considered as an attack that should be addressed through Information Security. Besides, professional standards are established to protect information from threats in cyber reals such as personal information on social media; however, Information security professional standards consider the security of information assets to ensure information confidentiality, availability, and integrity.

Importance of Cyber Security

Today we live in a digital era where all aspects of our lives depend on the network, computer and other electronic devices, and software applications. All critical infrastructure such as the banking system, healthcare, financial institutions, governments, and manufacturing industries use **devices connected to the Internet** as a core part of their operations. Some of their information, such as intellectual property, financial data, and personal data, can be sensitive for unauthorized access or exposure that could

have **negative consequences**. This information gives intruders and threat actors to infiltrate them for financial gain, extortion, political or social motives, or just vandalism.

Cyber-attack is now an international concern that hacks the system, and other security attacks could endanger the global economy. Therefore, it is essential to have an excellent cyber security strategy to protect sensitive information from high-profile security breaches. Furthermore, as the volume of cyber-attacks grows, companies and organizations, especially those that deal with information related to national security, health, or financial records, need to use strong cyber security measures and processes to protect their sensitive business and personal information.

Who are Cyber Criminals?

Cybercriminals are those who conduct such acts. They can be categorized into three groups that reflect their motivation.

Type 1: Cybercriminals – hungry for recognition:
- Hobby hackers;
- IT professionals (social engineering is one of the biggest threats);
- Politically motivated hackers;
- Terrorist organizations.

Type 2: Cybercriminals – not interested in recognition:
- Psychological prevents;
- Financially motivated hackers (corporate espionage);
- State – sponsored hacking (national espionage, sabotage);
- Organized criminals.

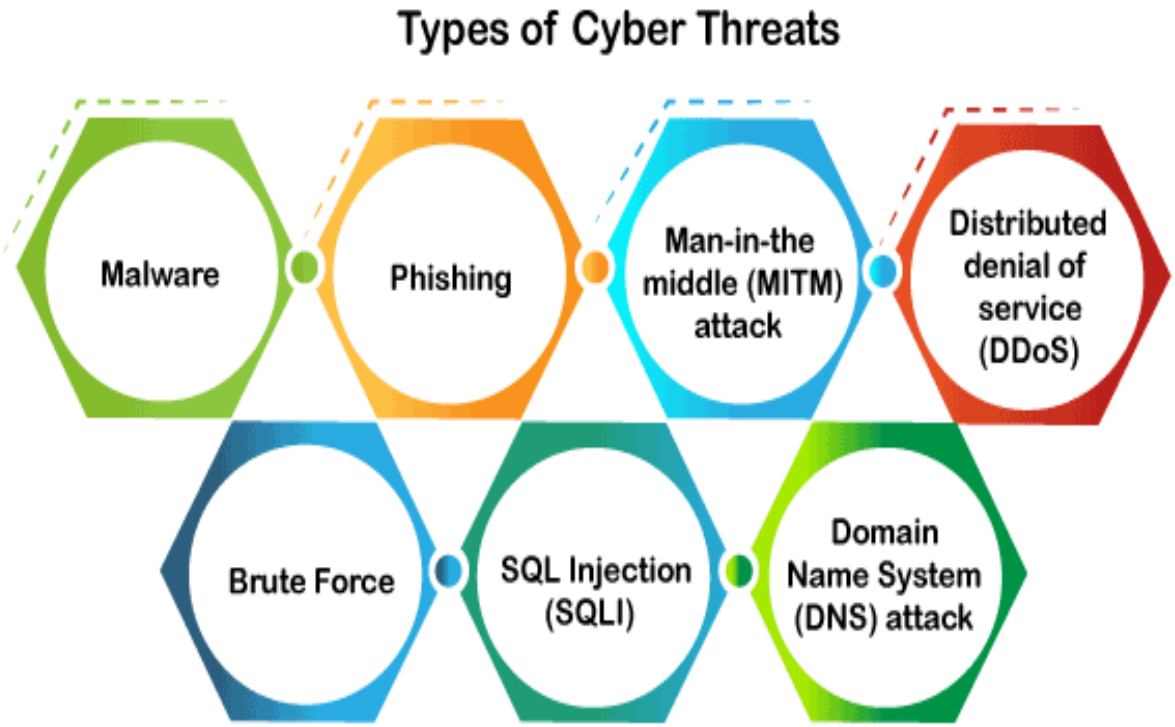
Type 3: Cybercriminals – the insiders:
- former employees seeking revenge;
- Competing companies using employees to gain economic advantage through damage and/or theft.

Who are the victims of cyber-attacks?

Companies or organizations, which have a lot of important information, and customer data, especially news media, banks, and e-commerce companies are most susceptible to cyber-attacks. Also, if a person's social media account is hacked, it becomes a cyber-attack. Hackers mainly target popular people for cyber-attacks on social media accounts.

Types of Cyber Security Threats

A threat in cyber security is a malicious activity by an individual or organization to corrupt or steal data, gain access to a network, or disrupts digital life in general. The cyber community defines the following threats available today:



Malware

Malware means malicious software, which is the most common cyber attacking tool. It is used by the cybercriminal or hacker to disrupt or damage a legitimate user's system.

Phishing

Phishing is a type of cybercrime in which **a sender seems to come from a genuine organization** like PayPal, eBay, financial institutions, or friends and co-workers. They contact a target or targets via email, phone, or text message with a link to persuade them to click on that links. This link will redirect them to fraudulent websites to provide sensitive data such as personal information, banking and credit card information, social security numbers, usernames, and passwords. Clicking on the link will **also install malware** on the target devices that allow hackers to control devices remotely.

Man-in-the-middle (MITM) attack

A man-in-the-middle attack is a type of cyber threat (a form of eavesdropping attack) in which a cybercriminal **intercepts a conversation or data transfer between two individuals**. Once the cybercriminal places themselves in the middle of a two-party communication, they seem like genuine participants and can get sensitive information and return different responses. The main objective of this type of attack is to gain access to our business or customer data. **For example**, a cybercriminal could intercept data passing between the target device and the network on an unprotected Wi-Fi network.

Distributed denial of service (DDoS)

It is a type of cyber threat or malicious attempt where cybercriminals disrupt targeted servers, services, or network's regular traffic by fulfilling legitimate requests to the target or its surrounding infrastructure with Internet traffic. Here the requests come from several IP addresses that can make the system unusable, overload their servers, slowing down significantly or temporarily taking them offline, or preventing an organization from carrying out its vital functions.

Brute Force

A brute force attack is a **cryptographic hack that uses a trial-and-error method** to guess all possible combinations until the correct information is discovered. Cybercriminals usually use this attack to obtain personal information about targeted passwords, login info, encryption keys, and Personal Identification Numbers (PINS).

SQL Injection (SQLI)

SQL injection is a common attack that occurs when cybercriminals use malicious SQL scripts for backend database manipulation to access sensitive information. Once the attack is successful, the malicious actor can view, change, or delete sensitive company data, user lists, or private customer details stored in the SQL database.

Domain Name System (DNS) attack

A DNS attack is a type of cyber-attack in which cyber criminals take advantage of flaws in the Domain Name System to redirect site users to malicious websites (DNS hijacking) and steal data from affected computers. It is a severe cyber security risk because the DNS system is an essential element of the internet infrastructure.

The following are the system that can be affected by security breaches and attacks:

- **Communication**: Cyber attackers can use phone calls, emails, text messages, and messaging apps for cyber-attacks.
- **Finance**: This system deals with the risk of financial information like bank and credit card detail. This information is naturally a primary target for cyber attackers.
- **Governments**: The cybercriminal generally targets the government institutions to get confidential public data or private citizen information.
- **Transportation**: In this system, cybercriminals generally target connected cars, traffic control systems, and smart road infrastructure.
- **Healthcare**: A cybercriminal targets the healthcare system to get the information stored at a local clinic to critical care systems at a national hospital.
- **Education**: A cybercriminals target educational institutions to get their confidential research data and information of students and employees**.**

Cyber Safety Tips

Let us see how to protect ourselves when any cyber-attacks happen. The following are the popular cyber safety tips:

Conduct cyber security training and awareness: Every organization must train their staffs on cyber security, company policies, and incident reporting for a strong cyber security policy to be successful. If the staff does unintentional or intentional malicious activities, it may fail the best technical safeguards that result in an expensive security breach. Therefore, it is useful to conduct security training and awareness for staff through seminars, classes, and online courses that reduce security violations.

Update software and operating system: The most popular safety measure is to update the software and O.S. to get the benefit of the latest security patches.

Use anti-virus software: It is also useful to use the anti-virus software that will detect and removes unwanted threats from your device. This software is always updated to get the best level of protection.

Perform periodic security reviews: Every organization ensures periodic security inspections of all software and networks to identify security risks early in a secure environment. Some popular examples of security reviews are application and network penetration testing, source code reviews, architecture design reviews, and red team assessments. In addition, organizations should prioritize and mitigate security vulnerabilities as quickly as possible after they are discovered.

Use strong passwords: It is recommended to always use long and various combinations of characters and symbols in the password. It makes the passwords are not easily guessable.

Do not open email attachments from unknown senders: The cyber expert always advises not to open or click the email attachment getting from unverified senders or unfamiliar websites because it could be infected with malware.

Avoid using unsecured Wi-Fi networks in public places: It should also be advised not to use insecure networks because they can leave you vulnerable to man-in-the-middle attacks.

Backup data: Every organization must periodically take backup of their data to ensure all sensitive data is not lost or recovered after a security breach. In addition, backups can help maintain data integrity in cyber-attack such as SQL injections, phishing, and ransom ware.