OSI SECURITY ARCHITECTURE

The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined briefly as follows:

Security attack – Any action that compromises the security of information owned by an organization
Security mechanism – A mechanism that is designed to detect, prevent or recover from a security attack
Security service – A service that enhances the security of the data processing systems and the information transfers of an organization.

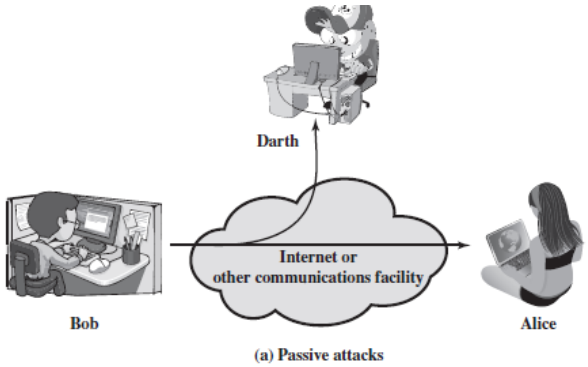
SECURITY ATTACK

There are two types of attacks

- Passive attacks
- Active attacks

Passive attack

Passive attacks attempt to learn or make use of information from the system but do not affect system resources. The goal of the opponent is to obtain information that is being transmitted.



(a) Passive attacks

Passive attacks are of two types

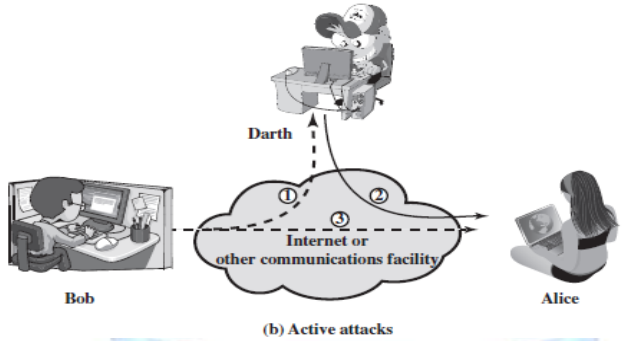
- **Release of message contents**
- **Traffic analysis**

Release of message contents: The opponent would learn the contents of the transmission. A telephone conversation, an e-mail message and a transferred file may contain sensitive or confidential information. We would like to prevent the opponent from learning the contents of these transmissions.

Traffic analysis: The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place. Passive attacks are very difficult to detect, because they do not involve any alteration of the data. However, it is feasible to prevent the success of these attacks.

Active attacks

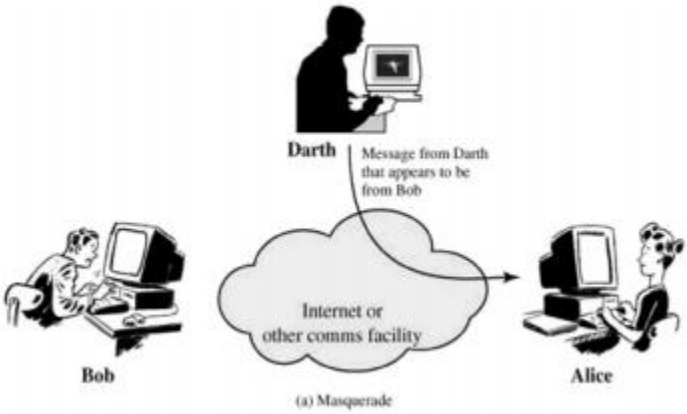
These attacks involve some modification of the data stream or the creation of a false stream.



(b) Active attacks

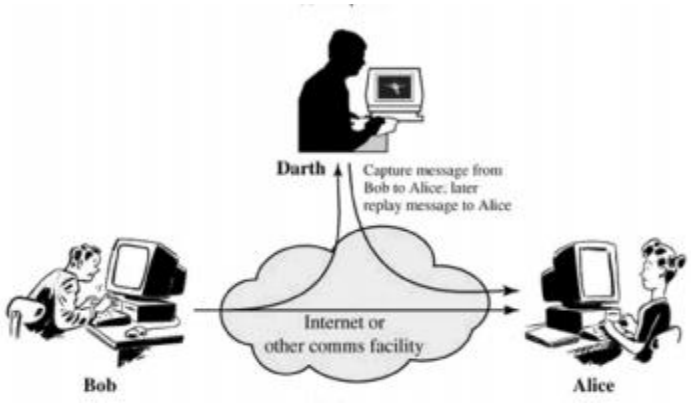
Active attacks can be classified in to four categories:

Masquerade – One entity pretends to be a different entity. Here, the attacker capturers the authentication and impersonifies the sender.

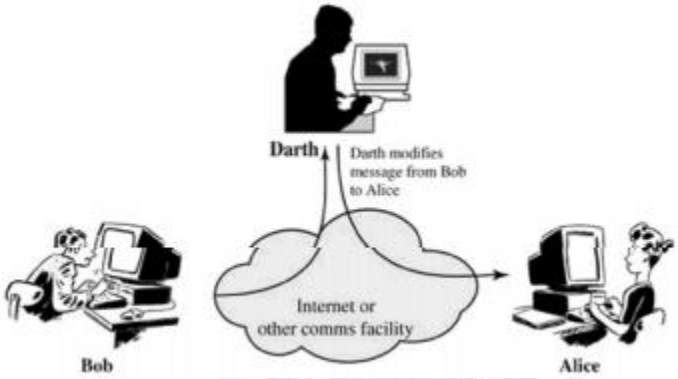


(a) Masquerade

Replay – The attacker captures the message and retransmits the message without modification to produce unauthorized effect.



Modification of messages – The attacker captures the message and retransmits the message with modification to produce unauthorized effect.



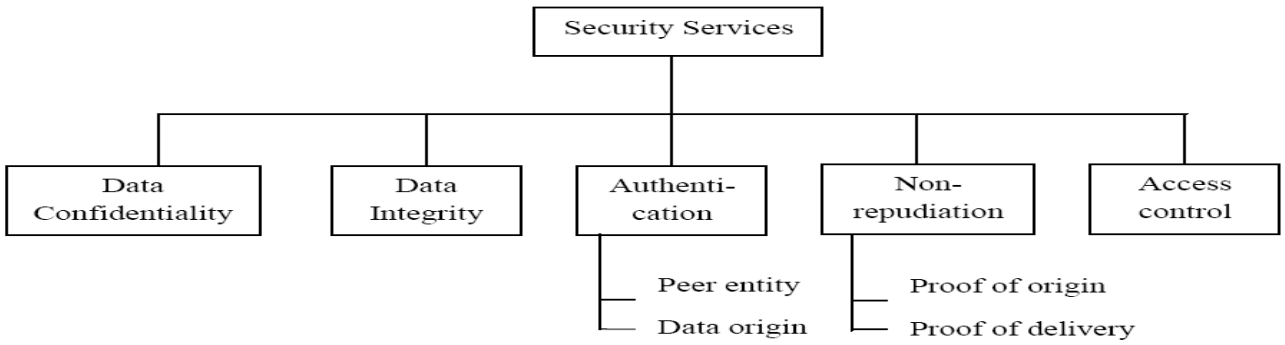
Denial of service – The attacker may suppress all messages directed to a particular destination. Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

It is quite difficult to prevent active attacks absolutely, because to do so would require physical protection of all communication facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them.

SECURITY SERVICES

X.800 defines a security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers.

The classification of security services are as follows:



(i) Authentication: The authentication service is concerned with assuring that a communication is authentic.

Two specific authentication services are defined in X.800:

- **Peer entity authentication:** Provide confidence in the identity of entities connected.
- **Data origin authentication:** Provide assurance that the source of received data is as claimed.

(ii) Access control: Access control is the ability to limit and control the access to host systems and applications.

(iii) Data Confidentiality: Confidentiality is the protection of transmitted data from passive attacks.

- **Connection Confidentiality**
  The protection of all user data on a connection
- **Connectionless Confidentiality**
  The protection of all user data in a single data block
- **Selective-Field Confidentiality**
  The confidentiality of selected fields within the user data on a connection or in a single data block
- **Traffic-Flow Confidentiality**
  The protection of the information that might be derived from observation of traffic flows

(iv)Data Integrity: The assurance that data received are exactly as sent by an authorized entity.

- **Connection Integrity with Recovery**
  Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.
- **Connection Integrity without Recovery**
  As above, but provides only detection without recovery.
- **Selective-Field Connection Integrity**
  Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.
- **Connectionless Integrity**
  Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.
- **Selective-Field Connectionless Integrity**
  Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.

(v)Non repudiation: Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.

- **Nonrepudiation, Origin**
  Proof that the message was sent by the specified party
- **Nonrepudiation, Destination**
  Proof that the message was received by the specified party

SECURITY MECHANISMS

- **Encipherment:**

  It uses mathematical algorithm to transform data into a form that is not readily intelligible. It depends upon encryption algorithm and key

- **Digital signature:**

  Data appended to or a cryptographic transformation of a data unit that is to prove integrity of data unit and prevents from forgery

- **Access control**

A variety of mechanisms that enforce access rights to resources.

- **Data integrity**

A variety of mechanism are used to ensure integrity of data unit

- **Traffic padding**

The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.

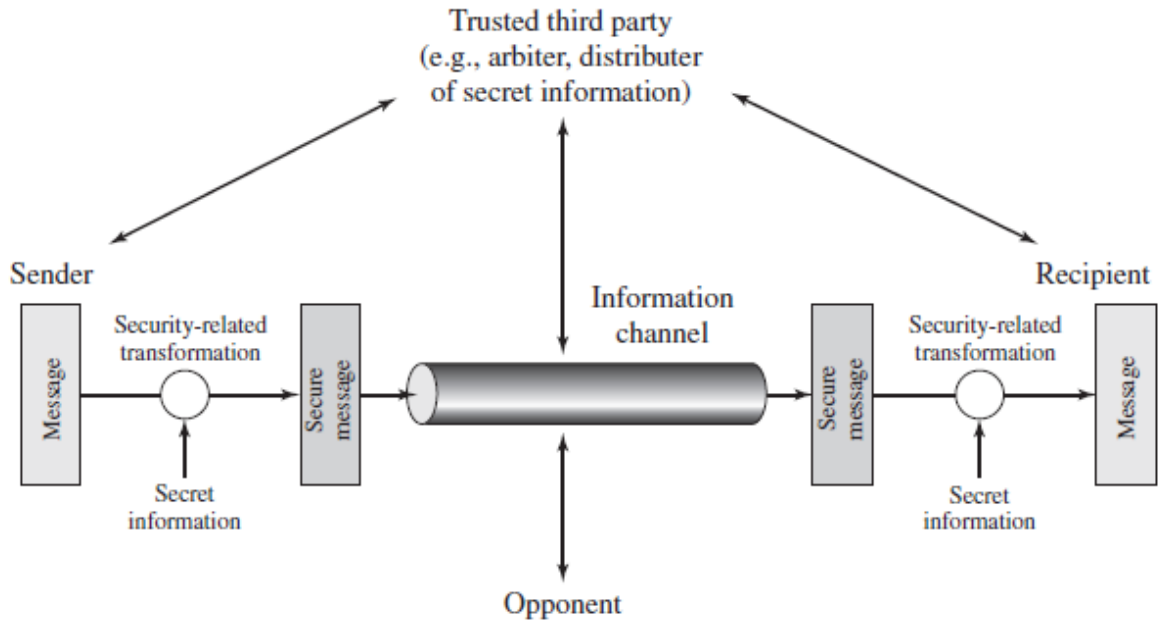
- **Notarization**
The use of a trusted third party to assure certain properties of a data exchange

A MODEL FOR NETWORK SECURITY

Encryption/Decryption methods fall into two categories.

- Symmetric key
- Public key

In symmetric key algorithms, the encryption and decryption keys are known both to sender and receiver. The encryption key is shared and the decryption key is easily calculated from it. In many cases, the encryption and decryption keys are the same. In public key cryptography, encryption key is made public, but it is computationally infeasible to find the decryption key without the information known to the receiver.



A message is to be transferred from one party to another across some sort of internet. The two parties, who are the principals in this transaction, must cooperate for the exchange to take place. A logical information channel is established by

defining a route through the internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals.

All the techniques for providing security have two components:

- A security-related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent.
- Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission

A trusted third party may be needed to achieve secure transmission. For example, a third party may be responsible for distributing the secret information to the two principals while keeping it from any opponent.

This general model shows that there are four basic tasks in designing a particular security service:

1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.
2. Generate the secret information to be used with the algorithm.
3. Develop methods for the distribution and sharing of the secret information.
4. Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

SUBSTITUTION TECHNIQUES

- A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.
- Substitution ciphers can be categorized as either

  i) Monoalphabetic ciphers or ii) polyalphabetic ciphers.
- In monoalphabetic substitution, the relationship between a symbol in the plaintext to a symbol in the ciphertext is always one-to-one.
- In polyalphabetic substitution, each occurrence of a character may have a different substitute. The relationship between a character in the plaintext to a character in the ciphertext is one-to-many.

Various substitution ciphers are

(i) Caesar Cipher

(ii) Mono alphabetic cipher

(iii) Playfair cipher

(iv) Hill cipher

(v) Poly alphabetic cipher

(vi) Vignere cipher

(i)CAESAR CIPHER (OR) SHIFT CIPHER

Caeser cipher was proposed by Julius Caesar. The Caesar cipher involves replacing each letter of the alphabet with the letter standing 3 places further down the alphabet.

```
plain:  meet me after the toga party
cipher: PHHW PH DIWHU WKH WRJD SDUWB
```

```
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
```

Let us assign a numerical equivalent to each letter:

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

Note that the alphabet is wrapped around, so that letter following ‘z’ is ‘a’.

For each plaintext letter p, substitute the cipher text letter c such that

c = E(3, p) = (p+3) mod 26

Decryption is

p=D(3,c)=(c-3) mod 26

The general Caesar algorithm is

C = E(k, p) = (p + k) mod 26

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

The decryption algorithm is simply

p = D(k, c) = (C - k) mod 26

If it is known that a given cipher text is a Caesar cipher, then a brute-force cryptanalysis is easily performed: simply try all the 25 possible keys.

Cryptanalysis of Caesar Cipher

1. The encryption and decryption algorithms are known
2. There are only 25 possible keys. Hence brute force attack takes place
3. The language of the plaintext is known and easily recognizable

---

(ii) MONOALPHABETIC CIPHER

- Each plaintext letter maps to a different random cipher text letter
- Here, 26! Possible keys are used to eliminate brute force attack

There is, however, another line of attack. If the cryptanalyst knows the nature of the plaintext (e.g., non-compressed English text), then the analyst can exploit the regularities of the language.

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
```

As a first step, the relative frequency of the letters can be determined and compared to a standard frequency distribution for English

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	T 2.50	I 0.83	N 0.00
O 7.50	X 4.17	A 1.67	J 0.83	R 0.00
M 6.67				

Continued analysis of frequencies plus trial and error should easily yield a solution.

(iii) PLAYFAIR CIPHER

The best known multiple letter encryption cipher is the playfair, which treats digrams in the plaintext as single units and translates these units into cipher text digrams. The playfair algorithm is based on the use of 5x5 matrix of letters constructed using a keyword.

Let the keyword be "monarchy".

The matrix is constructed by

- Filling in the letters of the keyword from left to right and from top to bottom
- Duplicates are removed
- Remaining unfilled cells of the matrix is filled with remaining alphabets in alphabetical order.

The matrix is 5x5. It can accommodate 25 alphabets. To accommodate the 26th alphabet I and J are counted as one character.

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

Rules for encryption

- Repeating plaintext letters that would fall in the same pair are separated with a filler letter such as 'x'.
- Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ar is encrypted as RM.

- Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, mu is encrypted as CM.
- Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, hs becomes BP and ea becomes IM (or JM, as the encipherer wishes).

Example

Plain text: Balloon

Ba ll oo n

Ba lx lo on

Ba→I/JB

lx→SU

lo→PM

on→NA

Strength of playfair cipher

- Playfair cipher is a great advance over simple mono alphabetic ciphers.
- Since there are 26 letters, 26x26 = 676 diagrams are possible, so identification of individual digram is more difficult.
- Frequency analysis is much more difficult.

Disadvantage

Easy to break because it has the structure and the resemblance of the plain text language

(iv) HILL CIPHER

It is a multi-letter cipher. It is developed by Lester Hill. The encryption algorithm takes m successive plaintext letters and substitutes for them m cipher text letters. The substitution is determined by m linear equations in which each character is assigned numerical value (a=0,b=1…z=25). For m =3 the system can be described as follows:

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} K_{11}K_{12}K_{13} \\ K_{21}K_{22}K_{23} \\ K_{31}K_{32}K_{33} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} \bmod 26$$

C=KP mod 26

C and P are column vectors of length 3 representing the cipher and plain text respectively.

Consider the message 'ACT', and

$$\begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix}$$

The key below (or GYBNQKURP in letters)

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}$$

Thus the enciphered vector is given by:

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix} = \begin{pmatrix} 67 \\ 222 \\ 319 \end{pmatrix} \equiv \begin{pmatrix} 15 \\ 14 \\ 7 \end{pmatrix} \pmod{26}$$

which corresponds to a ciphertext of 'POH'

Decryption

Decryption algorithm is done as **$P=K^{-1}C \bmod 26$**

In order to decrypt, we turn the ciphertext back into a vector, then simply multiply by the inverse matrix of the key matrix (IFKVIVVMI in letters).

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}^{-1} \equiv \begin{pmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{pmatrix}$$

Cipher text of 'POH'

$$\begin{pmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{pmatrix} \begin{pmatrix} 15 \\ 14 \\ 7 \end{pmatrix} \equiv \begin{pmatrix} 260 \\ 574 \\ 539 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix} \pmod{26}$$

Now gets us back the plain text 'ACT'

Merits and Demerits

- Completely hides single letter and 2 letter frequency information.
- Easily attacked with known plain text attack

(v)POLYALPHABETIC CIPHERS

Poly alphabetic cipher is a simple technique to improve mono-alphabetic technique.

The features are

- A set of related mono-alphabetic substitution rules are used
- A key determines which particular rule is chosen for a given transformation.

Example: **Vigenere Cipher**

Each of the 26 ciphers is laid out horizontally, with the key letter for each cipher to its left. A normal alphabet for the plaintext runs across the top. The process of encryption is simple: Given a key letter x and a plaintext letter y, the cipher text is at the intersection of the row labelled x and the column labelled y; in this case, the cipher text is V. To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating keyword.

Key=deceptive

Plain text= we are discovered save yourself

e.g., key = d e c e p t i v e d e c e p t i v e d e c e p t i v e

PT = w e a r e d i s c o v e r e d s a v e y o u r s e l f

CT = ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Decryption is equally simple. The key letter again identifies the row. The position of the cipher text letter in that row determines the column, and the plaintext letter is at the top of that column.

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

Strength of Vigenere cipher

o There are multiple ciphertext letters for each plaintext letter.

o Letter frequency information is obscured

(vi) VERNAM CIPHER or ONE-TIME PAD

It is an unbreakable cryptosystem. It represents the message as a sequence of 0s and 1s. This can be accomplished by writing all numbers in binary, for example, or by using ASCII. The key is a random sequence of 0"s and 1"s of same length as the message. Once a key is used, it is discarded and never used again.

The system can be expressed as follows:

$$C_i = P_i \oplus K_i$$

C_i - ith binary digit of cipher text P_i - i^{th} binary digit of plaintext K_i - ith binary digit of key

$\oplus$ – exclusive OR operation

Thus the cipher text is generated by performing the bitwise XOR of the plaintext and the key. Decryption uses the same key. Because of the properties of XOR, decryption simply involves the same bitwise operation:

$$P_i = C_i \oplus K_i$$

Advantages

- It is unbreakable since cipher text bears no statistical relationship to the plaintext
- Not easy to break

Drawbacks

- Practically impossible to generate a random key as to the length of the message
- The second problem is that of key distribution and key protection.

Due to the above two drawbacks, one time pad is of limited use and is used for low band width channel which needs high security.

TRANSPOSITION TECHNIQUES

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

RAIL FENCE CIPHER

It is simplest of such cipher, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.

Plaintext = meet at the school house

To encipher this message with a rail fence of depth 2,

We write the message as follows:

m e a t e c o l o s

e t t h s h o h u e

The encrypted message Cipher text MEATECOLOSETTHSHOHUE

ROW TRANSPOSITION CIPHERS-

A more complex scheme is to write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of columns then becomes the key of the algorithm.

e.g., plaintext = meet at the school house

Key = 4 3 1 2 5 6 7

PT = m e e t a t t

h e s c h o o

l h o u s e

CT = ESOTCUEEHMHLAHSTOETO

Demerits

- Easily recognized because the frequency is same in both plain text and cipher text.
- Can be made secure by performing more number of transpositions.

STEGANOGRAPHY

In Steganography, the plaintext is hidden. The existence of the message is concealed. For example, the sequence of first letters of each word of the overall message spells out the hidden message.

Various other techniques have been used historically; some examples are the following:

• **Character marking:** Selected letters of printed or typewritten text are overwritten in pencil. The marks are ordinarily not visible unless the paper is held at an angle to bright light.

• **Invisible ink:** A number of substances can be used for writing but leave no visible trace until heat or some chemical is applied to the paper.

• **Pin punctures:** Small pin punctures on selected letters are ordinarily not visible unless the paper is held up in front of a light.

• **Typewriter correction ribbon:** Used between lines typed with a black ribbon, the results of typing with the correction tape are visible only under a strong light.

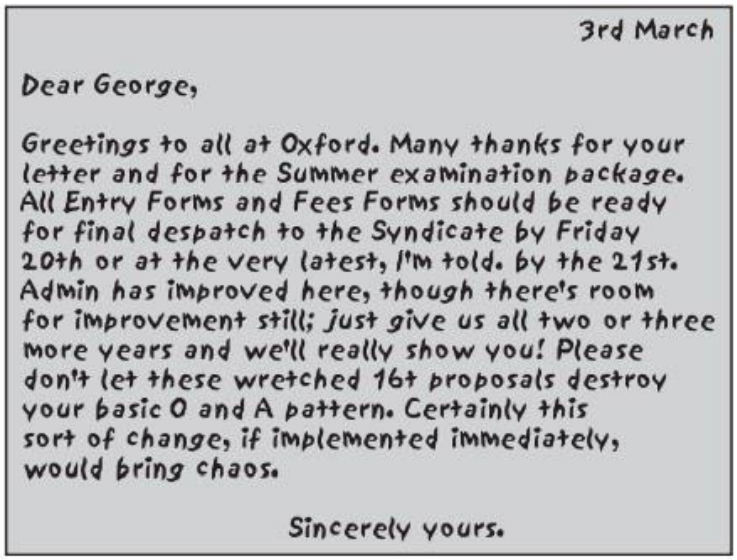
Drawback

- It requires a lot of overhead to hide a relatively few bits of information.
- Once the system is discovered, it becomes virtually worthless

STEGANOGRAPHY

A plaintext message may be hidden in one of two ways. The methods of **steganography** conceal the existence of the message, whereas the methods of cryp-tography render the message unintelligible to outsiders by various transformations of the text.

A simple form of steganography, but one that is time-consuming to construct, is one in which an arrangement of words or letters within an apparently innocuous text spells out the real message. For example, the sequence of first letters of each word of the overall message spells out the hidden message. Figure 2.9 shows an example in which a subset of the words of the overall message is used to convey the hidden message. See if you can decipher this; it's not too hard.



3rd March

Dear George,

Greetings to all at Oxford. Many thanks for your letter and for the Summer examination package. All Entry Forms and Fees Forms should be ready for final despatch to the Syndicate by Friday 20th or at the very latest, I'm told. by the 21st. Admin has improved here, though there's room for improvement still; just give us all two or three more years and we'll really show you! Please don't let these wretched 16t proposals destroy your basic O and A pattern. Certainly this sort of change, if implemented immediately, would bring chaos.

Sincerely yours.

Figure 2.9 A Puzzle for Inspector Morse
(From The Silent World of Nicholas Quinn, *by Colin Dexter)*

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

Various other techniques have been used historically; some examples are the following:

• **Character marking:** Selected letters of printed or typewritten text are over-written in pencil. The marks are ordinarily not visible unless the paper is held at an angle to bright light.

• **Invisible ink:** A number of substances can be used for writing but leave novisible trace until heat or some chemical is applied to the paper.

• **Pin punctures:** Small pin punctures on selected letters are ordinarily not visible unless the paper is held up in front of a light.

• **Typewriter correction ribbon:** Used between lines typed with a black ribbon, theresults of typing with the correction tape are visible only under a strong light.

Although these techniques may seem archaic, they have contemporary equivalents. Hiding a message by using the least significant bits of frames on a CD. For example, the Kodak Photo CD format's maximum resolution is 2048 _ 3072 pixels, with each pixel containing 24 bits of RGB color information.

The least significant bit of each 24-bit pixel can be changed without greatly affecting the quality of the image. The result is that you can hide a 2.3-megabyte message in a single digital snapshot. There are now a number of software packages available that take this type of approach to steganography.

Steganography has a number of drawbacks when compared to encryption. It requires a lot of overhead to hide a relatively few bits of information, although using a scheme like that proposed in the preceding paragraph may make it more effective.Also, once the system is discovered, it becomes virtually worthless. This problem, too, can be overcome if the insertion method depends on some sort of key. Alternatively, a message can be first encrypted and then hidden using steganography.

The advantage of steganography is that it can be employed by parties who have something to lose should the fact of their secret communication (not necessarily the content) be discovered. Encryption flags traffic as important or secret or may identify the sender or receiver as someone with something to hide.

Foundations of modern cryptography

Modern cryptography is the cornerstone of computer and communications security. Its foundation is based on various concepts of mathematics such as number theory, computational-complexity theory, and probability theory.

Characteristics of Modern Cryptography

There are three major characteristics that separate modern cryptography from the classical approach.

Classic Cryptography	Modern Cryptography
It manipulates traditional characters, i.e., letters and digits directly.	It operates on binary bit sequences.
It is mainly based on 'security through obscurity'. The techniques employed for coding were kept secret and only the parties involved in communication knew about them.	It relies on publicly known mathematical algorithms for coding the information. Secrecy is obtained through a secrete key which is used as the seed for the algorithms. The computational difficulty of algorithms, absence of secret key, etc., make it impossible for an attacker to obtain the original information even if he knows the algorithm used for coding.
It requires the entire cryptosystem for communicating confidentially.	Modern cryptography requires parties interested in secure communication to possess the secret key only.

Perfect Security

- A cipher system is said to offer perfect secrecy if, on seeing the ciphertext the interceptor gets **no extra information**about the plaintext than he had before the ciphertext was observed.
- In a cipher system with perfect secrecy the interceptor is "forced" to guess the plaintext.

- An encryption scheme satisfies perfect secrecy if for all messages m1, m2 in message space M and all ciphertexts c ∈ C, we have

where both probabilities are taken over the choice of K in K and over the coin tosses of the (possibly) probabilistic algorithm Enc.

- Intuitively, we might want to define perfect security of an encryption scheme as follows: Given a ciphertext all messages are equally likely.
- This can be formulated as: For all m(0) , m(1) ∈ M and c ∈ C we have:

  Pr[M = m (0) |C = c] = Pr[M = m (1) |C = c]

- The probability here is over the randomness used in the Gen and Enc algorithms and the probability distribution over the message space.

Definition (One: Perfect Security)

- We want the ciphertext to provide no additional information about the message
- Definition (One: Perfect Security)

  For all m ∈ M and c ∈ C, we have:

  Pr[M = m|C = c] = Pr[M = m]

- Here we are assuming that c ∈ C has Pr[C = c] > 0. Everywhere this assumption will be implicit

Definition (Two: Perfect Security)

- We want to say that the probability to generate a ciphertext given a message is independent of the message
- Definition (Two: Perfect Security)

  For all m ∈ M and c ∈ C we have:

  Pr[C = c|M = m] = Pr[C = c]

Definition (Three: Perfect Security)

- We want to say that the probability of generating a ciphertext given as message m(0) , is same as the probability of generating that ciphertext given any other different message m(1)
- Definition (Three: Perfect Security)

For any messages m(0) , m(1) ∈ M and c ∈ C we have:

Pr[C = c|M = m (0)] = Pr[C = c|M = m (1)]

Shannon's Original Definition of Secrecy

- Shannon defines perfect secrecy for secret-key systems and shows that they exist.
- A secret-key cipher obtains perfect secrecy if for all plaintexts *x* and all ciphertexts *y* it holds that $Pr(x) = Pr(x|y)$.
- In other words, a ciphertext y gives no information about the plaintext

Information theory

- **Information theory** studies the quantification, storage, and communication of information.
- A key measure in information theory is entropy. Entropy quantifies the amount of uncertainty involved in the value of a random variable or the outcome of a random process.
- For example, identifying the outcome of a fair coin flip (with two equally likely outcomes) provides less information (lower entropy) than specifying the outcome from a roll of a die (with six equally likely outcomes).
- Some other important measures in information theory are mutual information, channel capacity, error exponents, and relative entropy.

Quantities of information

- Information theory is based on probability theory and statistics.

- Information theory often concerns itself with measures of information of the distributions associated with random variables.
- Important quantities of information are entropy, a measure of information in a single random variable, and mutual information, a measure of information in common between two random variables.

A common unit of information is the bit, based on the binary logarithm

Entropy of an information source

- Based on the probability mass function of each source symbol to be communicated, the Shannon entropy *H*, in units of bits (per symbol), is given by

$$H = -\sum_i p_i \log_2(p_i)$$

- where p_i is the probability of occurrence of the *i*-th possible value of the source symbol.
- This equation gives the entropy in the units of "bits" (per symbol) because it uses a logarithm of base 2, and this base-2 measure of entropy has sometimes been called the shannon in his honor.
- If one transmits 1000 bits (0s and 1s), and the value of each of these bits is known to the receiver (has a specific value with certainty) ahead of transmission, it is clear that no information is transmitted.
- If, however, each bit is independently equally likely to be 0 or 1, 1000 shannons of information (more often called bits) have been transmitted. Between these two extremes, information can be quantified as follows.
- If $\mathbb{X}$ is the set of all messages $\{x_1, ..., x_n\}$ that *X* could be, and $p(x)$ is the probability of some $x \in X$, then the entropy, *H*, of *X* is defined:[

$$H(X) = \mathbb{E}_X[I(x)] = -\sum_{x \in \mathbb{X}} p(x) \log p(x).$$

- The special case of information entropy for a random variable with two outcomes is the binary entropy function, usually taken to the logarithmic base 2, thus having the shannon (Sh) as unit:

$$H_{\mathrm{b}}(p) = -p \log_2 p - (1-p) \log_2 (1-p).$$

Joint entropy

- The *joint entropy* of two discrete random variables *X* and *Y* is merely the entropy of their pairing: (*X*, *Y*). This implies that if *X* and *Y* are independent, then their joint entropy is the sum of their individual entropies.
- For example, if (*X*, *Y*) represents the position of a chess piece — *X* the row and *Y* the column, then the joint entropy of the row of the piece and the column of the piece will be the entropy of the position of the piece.
    - Despite similar notation, joint entropy should not be confused with *cross entropy*.

$$H(X,Y) = \mathbb{E}_{X,Y}[-\log p(x,y)] = -\sum_{x,y} p(x,y) \log p(x,y)$$

Conditional entropy (equivocation)

- The *conditional entropy* or *conditional uncertainty* of *X* given random variable *Y* (also called the *equivocation* of *X* about *Y*) is the average conditional entropy over *Y*:
- Because entropy can be conditioned on a random variable or on that random variable being a certain value, care should be taken not to confuse these two definitions of conditional entropy, the former of which is in more common use. A basic property of this

$$H(X|Y) = \mathbb{E}_Y[H(X|y)] = -\sum_{y \in Y} p(y) \sum_{x \in X} p(x|y) \log p(x|y) = -\sum_{x,y} p(x,y) \log p(x|y).$$

- The *conditional entropy* or *conditional uncertainty* of *X* given random variable *Y* (also called the *equivocation* of *X* about *Y*) is the average conditional entropy over *Y*:

- Because entropy can be conditioned on a random variable or on that random variable being a certain value, care should be taken not to confuse these two definitions of conditional entropy, the former of which is in more common use. A basic property of this form of conditional entropy is that:

$$H(X|Y) = H(X,Y) - H(Y).$$

Mutual information (Transinformation)

- *Mutual information* measures the amount of information that can be obtained about one random variable by observing another. It is important in communication where it can be used to maximize the amount of information shared between sent and received signals. The mutual information of *X* relative to *Y* is given by:

$$I(X;Y) = \mathbb{E}_{X,Y}[SI(x,y)] = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)}$$

- where SI (*S*pecific mutual *I*nformation) is the pointwise mutual information.
- A basic property of the mutual information is that

$$I(X;Y) = H(X) - H(X|Y).$$

- That is, knowing *Y*, we can save an average of *I*(*X*; *Y*) bits in encoding *X* compared to not knowing *Y*.
- Mutual information is symmetric:

$$I(X;Y) = I(Y;X) = H(X) + H(Y) - H(X,Y).$$

Kullback–Leibler Divergence (Information Gain):

- The *Kullback–Leibler divergence* (or *information divergence*, *information gain*, or *relative entropy*) is a way of comparing two distributions: a "true" probability distribution *p(X)*, and an arbitrary probability distribution *q(X)*.
- If we compress data in a manner that assumes *q(X)* is the distribution underlying some data, when, in reality, *p(X)* is the correct distribution, the Kullback–Leibler divergence is the number of average additional bits per datum necessary for compression.
- It is thus defined

$$D_{\mathrm{KL}}(p(X)\|q(X)) = \sum_{x\in X} -p(x)\log q(x) \; - \sum_{x\in X} -p(x)\log p(x) = \sum_{x\in X} p(x)\log\frac{p(x)}{q(x)}.$$

Coding theory

- Coding theory is one of the most important and direct applications of information theory.
- It can be subdivided into source coding theory and channel coding theory.
- Using a statistical description for data, information theory quantifies the number of bits needed to describe the data, which is the information entropy of the source.
- Data compression (source coding): There are two formulations for the compression problem:
  - lossless data compression: the data must be reconstructed exactly;
  - lossy data compression: allocates bits needed to reconstruct the data, within a specified fidelity level measured by a distortion function. This subset of information theory is called *rate–distortion theory*.
- Error-correcting codes (channel coding): While data compression removes as much redundancy as possible, an error correcting code adds just the right kind of redundancy (i.e., error correction) needed to transmit the data efficiently and faithfully across a noisy channel.

Product Cryptosystems

- Data encryption scheme in which the ciphertext produced by encrypting a plaintext document is subjected to further encryption.
- By combining two or more simple transposition ciphers or substitution ciphers, a more secure encryption may result.
- A cryptosystem ***S=(P,K, C,e,d)*** with the sets ofplaintexts ***P***, keys ***K*** and cryptotexts***C*** andencryption (decryption) algorithms ***e*** (***d***) is called **endomorphic** if ***P=C***.
- If $S_1=(P,K_1, P,e^{(1)},d^{(1)})$ and $S_2=(P,K_2, P,e^{(2)},d^{(2)})$ are endomorphic cryptosystems,then the**product cryptosystem** is
- $S_1 Ä S_2=(P,K_1 Ä K_2, P,e,d)$,
- where encryption is performed by the procedure
- $e_{(k1,k2)}(w) = e_{k2}(e_{k1}(w))$
- and decryption by the procedure
- $d_{(k1,k2)}(c) = d_{k1}(d_{k2}(c))$

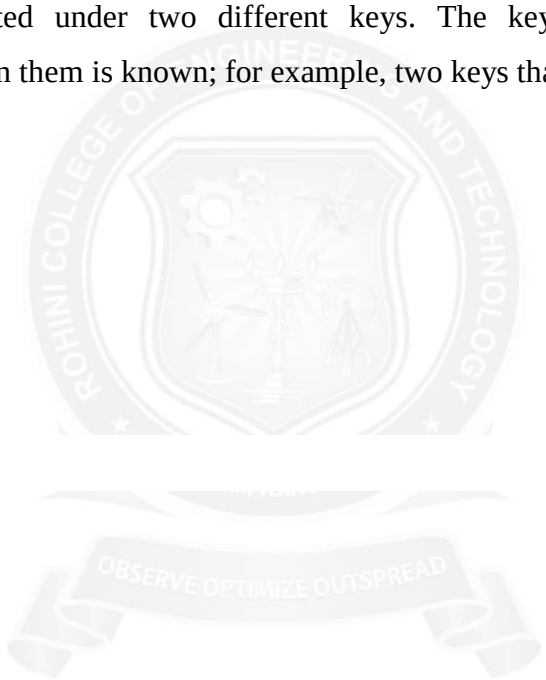
Cryptanalysis

- Cr**yptanalysis** is the study of analyzing information systems in order to study the hidden aspects of the systems.
- Cryptanalysis is used to breach cryptographic security systems and gain access to the contents of encrypted messages, even if the cryptographic key is unknown.
- In addition to mathematical analysis of cryptographic algorithms, cryptanalysis includes the study of side-channel attacks that do not target weaknesses in the cryptographic algorithms themselves, but instead exploit weaknesses in their implementation.

Methods

- *Ciphertext-only*: the cryptanalyst has access only to a collection of ciphertexts or codetexts.

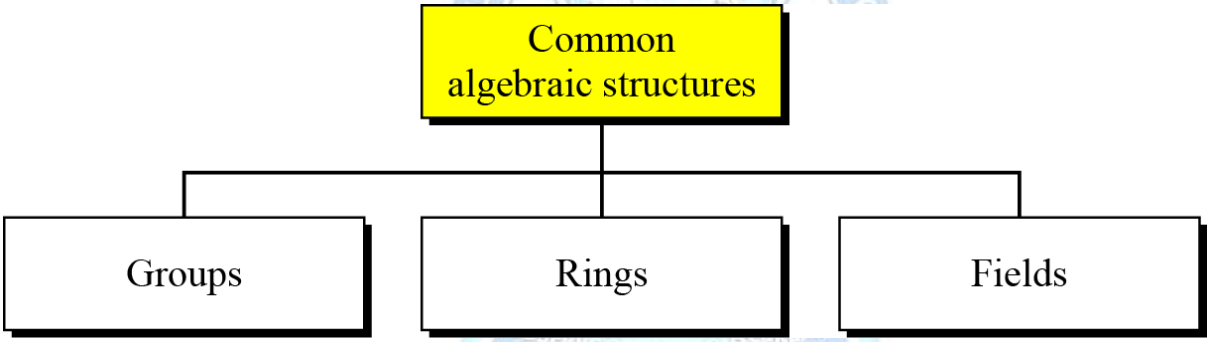
- *Known-plaintext*: the attacker has a set of ciphertexts to which he knows the corresponding plaintext.
- *Chosen-plaintext* (*chosen-ciphertext*): the attacker can obtain the ciphertexts (plaintexts) corresponding to an arbitrary set of plaintexts (ciphertexts) of his own choosing.
- *Adaptive chosen-plaintext*: like a chosen-plaintext attack, except the attacker can choose subsequent plaintexts based on information learned from previous encryptions. Similarly *Adaptive chosen ciphertext attack*.
- *Related-key attack*: Like a chosen-plaintext attack, except the attacker can obtain ciphertexts encrypted under two different keys. The keys are unknown, but the relationship between them is known; for example, two keys that differ in the one bit.



MATHEMATICS OF SYMMETRIC KEY CRYPTOGRAPHY:

Algebraic structures

Cryptography requires sets of integers and specific operations that are defined for those sets. The combination of the set and the operations that are applied to the elements of the set is called an algebraic structure. Three common algebraic structures: groups, rings, and fields.



MODULAR ARITHMETIC

If *a* is an integer and *n* is a positive integer, we define *a* mod *n* to be the remainder when *a* is divided by *n*. The integer *n* is called the **modulus**.

$$a = qn + r \qquad 0 \le r < n;$$

$$q = \lfloor a/n \rfloor$$

Congruent modulo

Two integers a and b are said to be congruent modulo n if

a (mod n)≡ b (mod n)

a ≡ b (mod n)

73 ≡ 4 mod 23

Properties of modulo operator

Congruences have the following properties:

1. a≡ b (mod n) if n|(a-b)

2. a≡ b (mod n) implies b≡ a(mod n)

3. $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ imply $a \equiv c \pmod{n}$.

Modular Arithmetic Operations

Modular arithmetic exhibits the following properties:

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$

2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$

3. $[(a \bmod n) * (b \bmod n)] \bmod n = (a * b) \bmod n$

11 mod 8 = 3; 15 mod 8 = 7

[(11 mod 8) + (15 mod 8)] mod 8 = 10 mod 8 = 2

(11 + 15) mod 8 = 26 mod 8 = 2

[(11 mod 8) - (15 mod 8)] mod 8 = -4 mod 8 = 4

(11 - 15) mod 8 = -4 mod 8 = 4

$$[(11 \bmod 8) * (15 \bmod 8)] \bmod 8 = 21 \bmod 8 = 5$$

$$(11 * 15) \bmod 8 = 165 \bmod 8 = 5$$

GROUPS, RINGS, AND FIELDS

Groups, rings, and fields are the fundamental elements of a branch of mathematics known as abstract algebra, or modern algebra.

GROUPS

A **group** G, sometimes denoted by $\{G, \bullet\}$, is a set of elements with a binary operation denoted by $\bullet$ that associates to each ordered pair (a, b) of elements in G an element $(a \bullet b)$ in G, such that the following axioms are obeyed:

(A1) Closure: If a and b belong to G, then $a \bullet b$ is also in G.

(A2) Associative: $a\bullet(b\bullet c) = (a\bullet b)\bullet c$ for all a, b, c in G.

(A3) Identity element: There is an element e in G such that $a \bullet e = e \bullet a = a$ for all a in G.

(A4) Inverse element: For each a in G, there is an element a^{-1} in G such that $a \bullet a^{-1} = a^{-1} \bullet a = e$.

If a group has a finite number of elements, it is referred to as a **finite group**, and the **order** of the group is equal to the number of elements in the group. Otherwise, the group is an **infinite group**. A group is said to be **abelian** if it satisfies the following additional condition:

(A5) Commutative: $a \bullet b = b \bullet a$ for all a, b in G.

A group G is **cyclic** if every element of G is a power a^k (k is an integer) of a fixed element $a \, \varepsilon \, G$. The element a is said to **generate** the group G or to be a **generator** of G. A cyclic group is always abelian and may be finite or infinite.

RINGS

A **ring** R, sometimes denoted by $\{R, +, *\}$, is a set of elements with two binary operations, called *addition* and *multiplication*,6 such that for all a, b, c in R the following axioms are obeyed.

(A1–A5) R is an abelian group with respect to addition; that is, R satisfies axioms A1 through A5.

(M1) Closure under multiplication: If a and b belong to R, then ab is also in R.

(M2) Associativity of multiplication: $a(bc) = (ab)c$ for all a, b, c in R.

(M3) Distributive laws: $a(b + c) = ab + ac$ for all a, b, c in R.

$(a + b)c = ac + bc$ for all a, b, c in R.

A ring is said to be **commutative** if it satisfies the following additional condition:

(M4) Commutativity of multiplication: $ab = ba$ for all a, b in R.

An **integral domain**, which is a commutative ring that obeys the following axioms.

(M5) Multiplicative identity: There is an element 1 in R such that $a1 = 1a = a$ for all a in R.

(M6) No zero divisors: If a, b in R and $ab = 0$, then either $a = 0$ or $b = 0$.

FIELDS

A **field** F, sometimes denoted by $\{F, +, *\}$, is a set of elements with two binary operations, called *addition* and *multiplication*, such that for all a, b, c in F the following axioms are obeyed.

(A1–M6) F is an integral domain; that is, F satisfies axioms A1 through A5 and M1 through M6.

(M7) Multiplicative inverse: For each a in F, except 0, there is an element a-1 in F such that

$aa^{-1} = (a^{-1})a = 1$

Relatively prime

Two integers are **relatively prime,** if their only common positive integer factor is 1.

8 and 15 are relatively prime because

Positive divisors of 8 are 1,2,4,8

Positive divisors of 15 are 1, 3, 5, 15

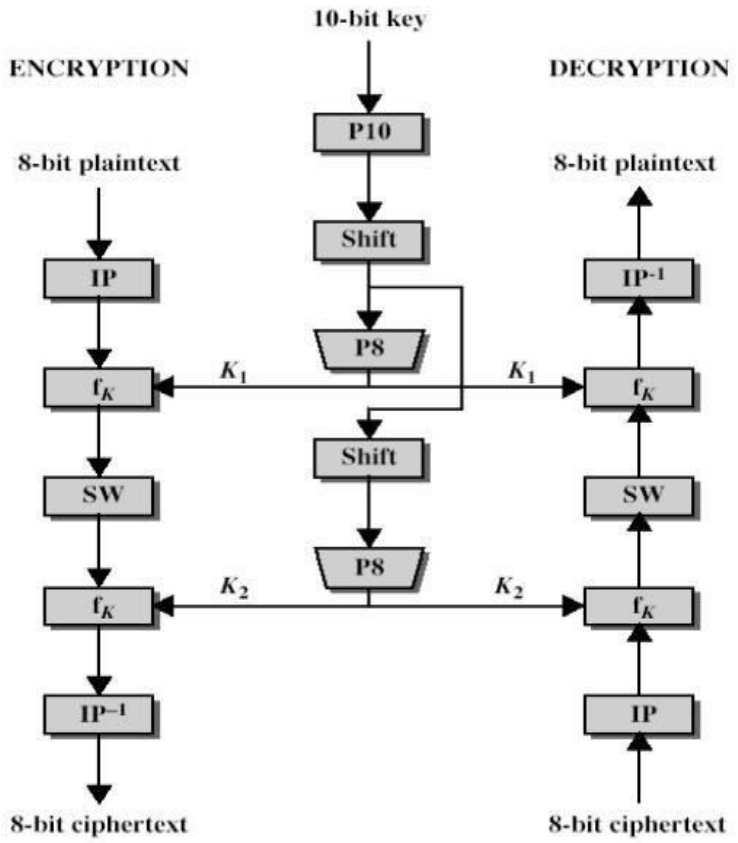
Therefore, common positive factor=1.

SIMPLIFIED DATA ENCRYPTION STANDARD (S-DES)

Simplified DES, developed by Professor Edward Schaefer of Santa Clara University, is an educational rather than a secure encryption algorithm. It has similar properties and structure to DES with much smaller parameters.

OVERVIEW

- The S-DES encryption algorithm takes an 8-bit block of plaintext (example: 10111101) and a 10-bit key as input and produces an 8-bit block of ciphertext as output.
- The S-DES decryption algorithm takes an 8-bit block of ciphertext and the same 10-bit key used to produce that ciphertext as input and produces the original 8-bit block of plaintext.
- The S-DES encryption algorithm takes an 8-bit block of plaintext (example: 10111101) and a 10-bit key as input and produces an 8-bit block of ciphertext as output. The S-DES decryption algorithm takes an 8-bit block of ciphertext and the same 10-bit key used to produce that ciphertext as input and produces the original 8-bit block of plaintext.



Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

- The encryption algorithm involves five functions: an initial permutation (IP); a complex function labeled f_K, which involves both permutation and substitution operations and depends on a key input; a simple permutation function that switches (SW) the two halves of the data; the function f_K again; and finally a permutation function that is the inverse of the initial permutation (IP^{-1}).
- The use of multiple stages of permutation and substitution results in a more complex algorithm, which increases the difficulty of cryptanalysis.
- The function f_K takes as input not only the data passing through the encryption algorithm, but also an 8-bit key. The algorithm could have been designed to work with a 16-bit key, consisting of two 8-bit subkeys, one used for each occurrence of f_K. Alternatively, a single 8-bit key could have been used, with the same key used twice in the algorithm.
- A compromise is to use a 10-bit key from which two 8-bit subkeys are generated. In this case, the key is first subjected to a permutation (P10). Then a shift operation is

performed. The output of the shift operation then passes through a permutation function that produces an 8-bit output (P8) for the first subkey (K_1). The output of the shift operation also feeds into another shift and another instance of P8 to produce the second subkey (K_2).

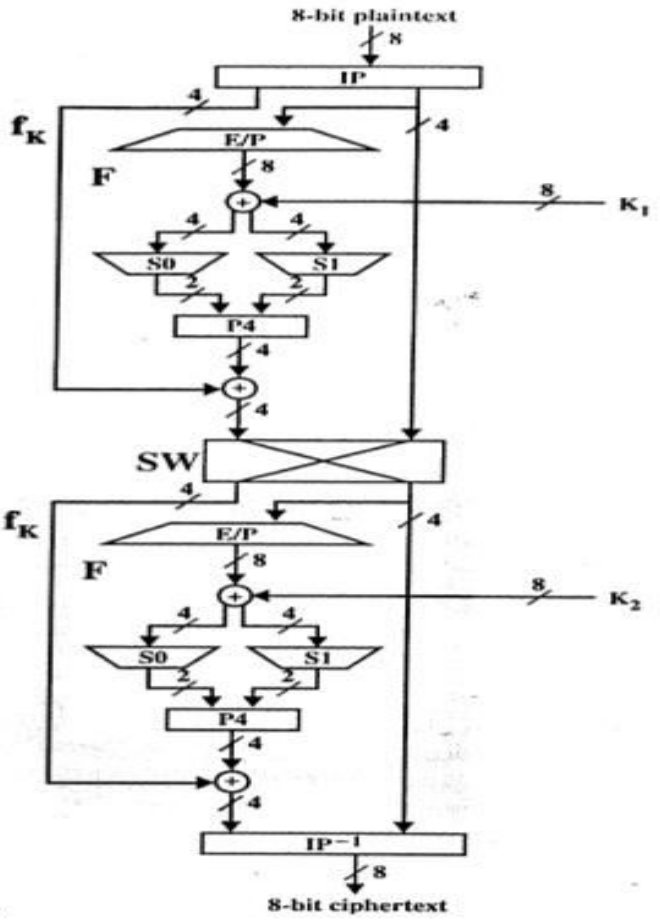


Figure 3.3 Simplified DES Scheme Encryption Detail.

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

- We can concisely express the encryption algorithm as a composition1 of functions:
  - IP-1 ○ fK2 ○ SW ○ fK1 ○ IP
- which can also be written as:
  - ciphertext = IP-1(fK2 (SW (fK1 (IP(plaintext)))))
- where K1 = P8(Shift(P10(key)))
  - K2 = P8(Shift(Shift(P10(key))))

- Decryption is essentially the reverse of encryption:
    - plaintext = IP-1(fK1 (SW (fK2 (IP(ciphertext)))))

KEY GENERATION

- S-DES depends on the use of a 10-bit key shared between sender and receiver.
- From this key, two 8-bit subkeys are produced for use in particular stages of the encryption and decryption algorithm

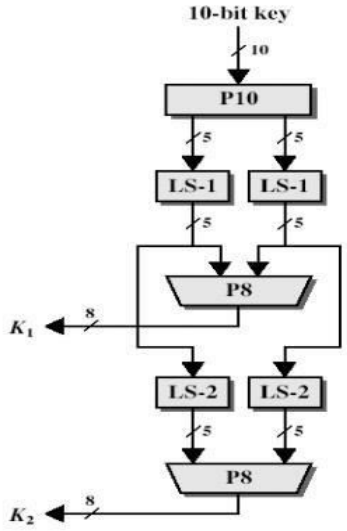


Figure: key generation for S-DES

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

- First, permute the key in the following fashion. Let the 10-bit key be designated as (k1, k2, k3, k4, k5, k6, k7, k8, k9, k10). Then the permutation P10 is defined as:
- P10(k1, k2, k3, k4, k5, k6, k7, k8, k9, k10) = (k3, k5, k2, k7, k4, k10, k1, k9, k8, k6)
- P10 can be concisely defined by the display:

P10									
3	5	2	7	4	10	1	9	8	6

- This table is read from left to right; each position in the table gives the identity of the input bit that produces the output bit in that position.
- So the first output bit is bit 3 of the input; the second output bit is bit 5 of the input, and so on. For example, the key (1010000010) is permuted to (1000001100).
- Next, perform a circular left shift (LS-1), or rotation, separately on the first five bits and the second five bits. In our example, the result is (00001 11000). Next we apply P8, which picks out and permutes 8 of the 10 bits according to the following rule:

P8							
6	3	7	4	8	5	10	9

- The result is subkey 1 (K1).
- In our example, this yields (10100100) We then go back to the pair of 5-bit strings produced by the two LS-1 functions and perform a circular left shift of 2 bit positions on each string. In our example, the value (00001 11000) becomes (00100 00011). Finally, P8 is applied again to produce K2. In our example, the result is (01000011).

S-DES ENCRYPTION

- encryption involves the sequential application of five functions.
- Initial and Final Permutations
- The input to the algorithm is an 8-bit block of plaintext, which we first permute using the IP function:
  - IP 2 6 3 1 4 8 5 7
- This retains all 8 bits of the plaintext but mixes them up.
- At the end of the algorithm, the inverse permutation is used:
  - IP–1 4 1 3 5 7 2 8 6
- It is easy to show by example that the second permutation is indeed the reverse of the first; that is, IP–1(IP(X)) = X.

The Function fK

- The most complex component of S-DES is the function fK, which consists of a combination of permutation and substitution functions.
- The functions can be expressed as follows.
- Let L and R be the leftmost 4 bits and rightmost 4 bits of the 8-bit input to fK, and let F be a mapping (not necessarily one to one) from 4-bit strings to 4-bit strings.
- Then we let
  - f K(L, R) = (L ! F(R, SK), R)
- where SK is a subkey and ! is the bit-by-bit exclusive-OR function.
- For example, suppose the output of the IP stage is (10111101) and F(1101, SK) = (1110) for some key SK.
  - Then fK(10111101) = (01011101) because (1011) ! (1110) = (0101).
- We now describe the mapping F. The input is a 4-bit number (n1n2n3n4).
- The first operation is an expansion/permutation operation:

E/P							
4	1	2	3	2	3	4	1

- For what follows, it is clearer to depict the result in this fashion:

n_4	n_1	n_2	n_3
n_2	n_3	n_4	n_1

- The 8-bit subkey K1 = (k11, k12, k13, k14, k15, k16, k17, k18) is added to this value using exclusive OR

$n_4 \oplus k_{11}$	$n_1 \oplus k_{12}$	$n_2 \oplus k_{13}$	$n_3 \oplus k_{14}$
$n_2 \oplus k_{15}$	$n_3 \oplus k_{16}$	$n_4 \oplus k_{17}$	$n_1 \oplus k_{18}$

- Let us rename these 8 bits:

$$\begin{array}{c|cc|c} p_{0,0} & p_{0,1} & p_{0,2} & p_{0,3} \\ p_{1,0} & p_{1,1} & p_{1,2} & p_{1,3} \end{array}$$

- The first 4 bits (first row of the preceding matrix) are fed into the S-box S0 to produce a 2- bit output, and the remaining 4 bits (second row) are fed into S1 to produce another 2-bit output. These two boxes are defined as follows:

$$S0 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \end{array} \qquad S1 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix} \end{array}$$

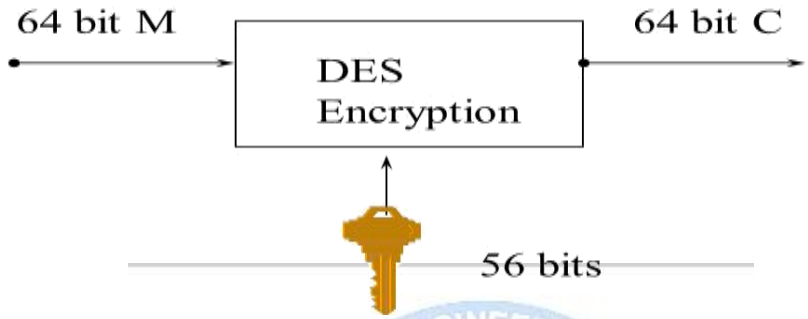
- The S-boxes operate as follows.
- The first and fourth input bits are treated as a 2-bit number that specify a row of the S-box, and the second and third input bits specify a column of the Sbox. The entry in that row and column, in base 2, is the 2-bit output. For example, if (p0,0p0,3) = (00) and (p0,1p0,2) = (10), then the output is from row 0, column 2 of S0, which is 3, or (11) in binary. Similarly, (p1,0p1,3) and (p1,1p1,2) are used to index into a row and column of S1 to produce an additional 2 bits.
- Next, the 4 bits produced by S0 and S1 undergo a further permutation as follows

P4			
2	4	3	1

- The output of P4 is the output of the function F. The Switch Function The function fK only alters the leftmost 4 bits of the input. The switch function (SW) interchanges the left and right 4 bits so that the second instance of fK operates on a different 4 bits. In this second instance, the E/P, S0, S1, and P4 functions are the same. The key input is K2.

DES - Data Encryption Standard

In DES data are encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption.



i) First, the **64-bit plaintext** passes through an initial permutation (IP) that rearranges the bits to produce the *permuted input*.

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Permutation table

i)Each entry in the permutation table indicates the position of a numbered input bit in the output

(ii) This is followed by a phase consisting of **sixteen rounds of the same function**, which involves both permutation and substitution functions.

iii) For each of the sixteen rounds, **a *subkey* (*Ki*)** is produced by the combination of **a left circular shift and a permutation**. The permutation function is the same for each round, but a different subkey is produced.

iv)The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the **pre-output**.

v)Finally, the pre-output is passed through a permutation that is the inverse of the initial permutation function, to produce the 64-bit cipher text.

The processing of plaintext proceeds in three phases

1. 64-bit plaintext passes through an initial permutation that rearranges the bits to produce the permuted input
2. 16 rounds of a function which involves both permutation and substitution functions. The output of the 16 th round consists of 64 bits that are a function of the input plaintext and the key. The left and the right halves of the output of the 16 th round is swapped to produce the preoutput.
3. The preoutput is passed through a permutation that is the inverse of the initial permutation.

The 64-bit input key is initially passed through a permutation function. Then for each of the 16 rounds a subkey is produced by combination of left circular shift and permutation. The permutation is the same for each of the

16 rounds but a different subkey is produced, because of the repeated shifts of the key bits.

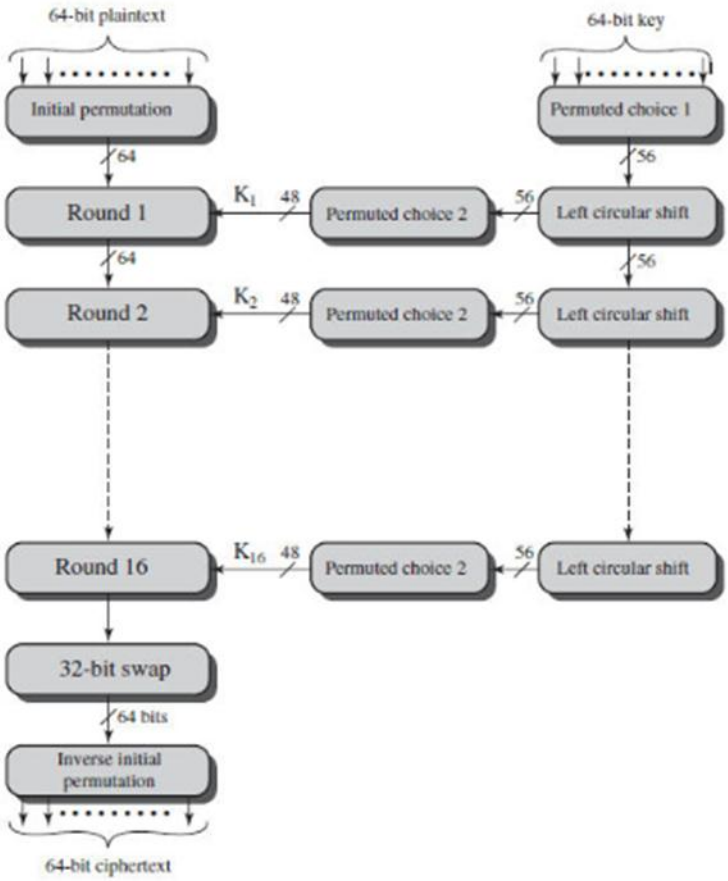
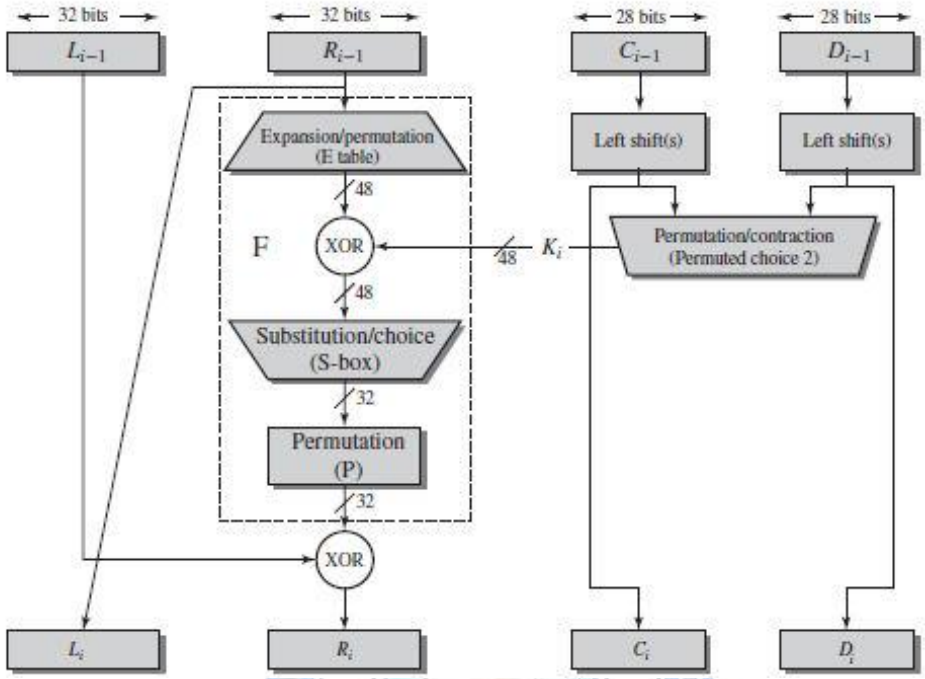


Figure 2.5: General Description of DES

Initial Permutation: It is the first step of the data computation that reorders the input data bits. The permutation. The table is to be interpreted as follows. Input to the table consists of 64 bits numbered
from 1 to 64.The 64 entries in the permutation table contain a permutation of the numbers from 1 to 64. Each entry in the permutation table indicates the position of a numbered input bit in the output, which also consists of 64 bits. It could be noted from the table that the even bits are placed in the left half and the odd bits are placed in the right half of the 64 bits of the data.

Final Permutation: It is the first step of the data computation that reorders the input data bits.



DES ROUND STRUCTURE

The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L (left) and R (right).

Each round can be summarized in the following formulae:

Li=Ri-1

Ri=Li-1 xor F(Ri-1+Ki)

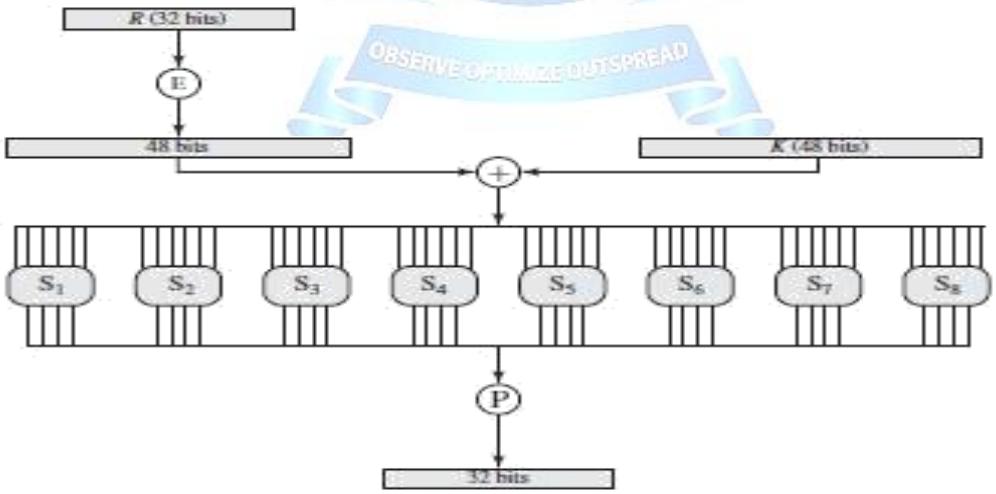
The round key is 48 bits. The input is 32 bits. This input is first expanded to 48 bits by using a table that defines a permutation plus an expansion that involves duplication of 16 of the bits.

The resulting 48 bits are XOR-ed with reduced key for the round. This 48-bit result passes through a substitution function that produces a 32-bit output, which is then permuted.

The substitution consists of a set of eight S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.

The first and last bits of the input to box form a 2-bit binary number to select one of four substitutions defined by the four rows in the table for Si.

The middle four bits select one of the sixteen columns. The decimal value in the cell selected by the row and column is then converted to its 4-bit representation to produce the output. For example, in S1, for input 011001, the row is 01 (row 1) and the column is 1100 (column 12) and assuming a value in row 1, column 12 is 9, the output is 1001.



DES - S- BOXES

Key Generation:

A 64-bit key is used as input to the algorithm. The bits of the key are numbered from 1 through 64; every eighth bit is ignored.

Bits included							Bits excluded
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	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

The key is first subjected to a permutation governed by a table labelled Permuted Choice This is nothing but permutation of numbers (referring to bit positions).

The resulting 56-bit key is then treated as two 28-bit strings, labelled *C*i, *D*i. At each round, *C*i, *D*i are separately subjected to a circular left shift or (rotation) of 1 or 2 bits, as governed by Table.

These shifted values serve as input to the next round. They also serve as input to the part labelled Permuted Choice Two , which produces a 48-bit output that serves as input to the function F(*R*i-1, *K*i *)*.

DES Decryption:

Decryption uses the same algorithm as encryption, except that the application of the sub-keys are reversed.

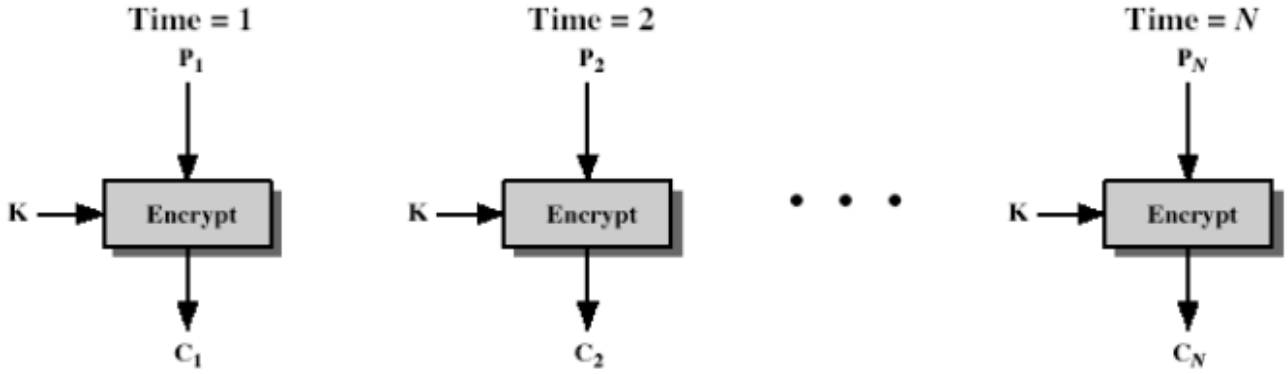
Block Cipher Modes of operations

- To apply a block cipher in a variety of applications, four "modes of operation" have been defined by NIST.

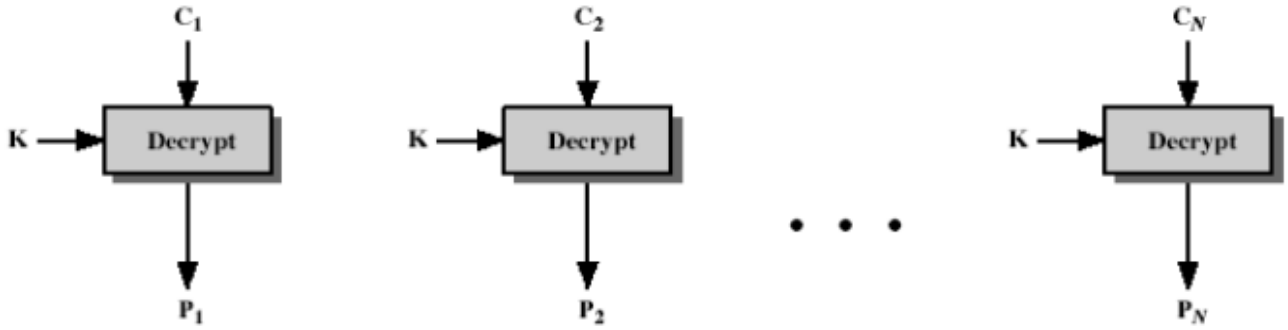
- A **mode of operation is a technique for enhancing the effect of a cryptographic algorithm** or adapting the algorithm for an application, such as applying a block cipher to a sequence of data blocks or a data stream.

Electronic code book (ECB)

The simplest mode is the electronic codebook (ECB) mode, in which plaintext is handled one block at a time and each block of plaintext is encrypted using the same key. The term *codebook* is used because, for a given key, there is a unique ciphertext for every *b*-bit block of plaintext.



(a) Encryption



(b) Decryption

For a message longer than b bits, the procedure is simply to break the message into b-bit blocks, padding the last block if necessary. Decryption is performed one block at a time, always using the same key.

Advantages :

- The ECB method is ideal for a short amount of data, such as an encryption key.
- For the same b-bit block of plaintext, if it appears more than once in the message, ECB always produces the same cipher text.

For lengthy messages, the ECB mode may not be secure.

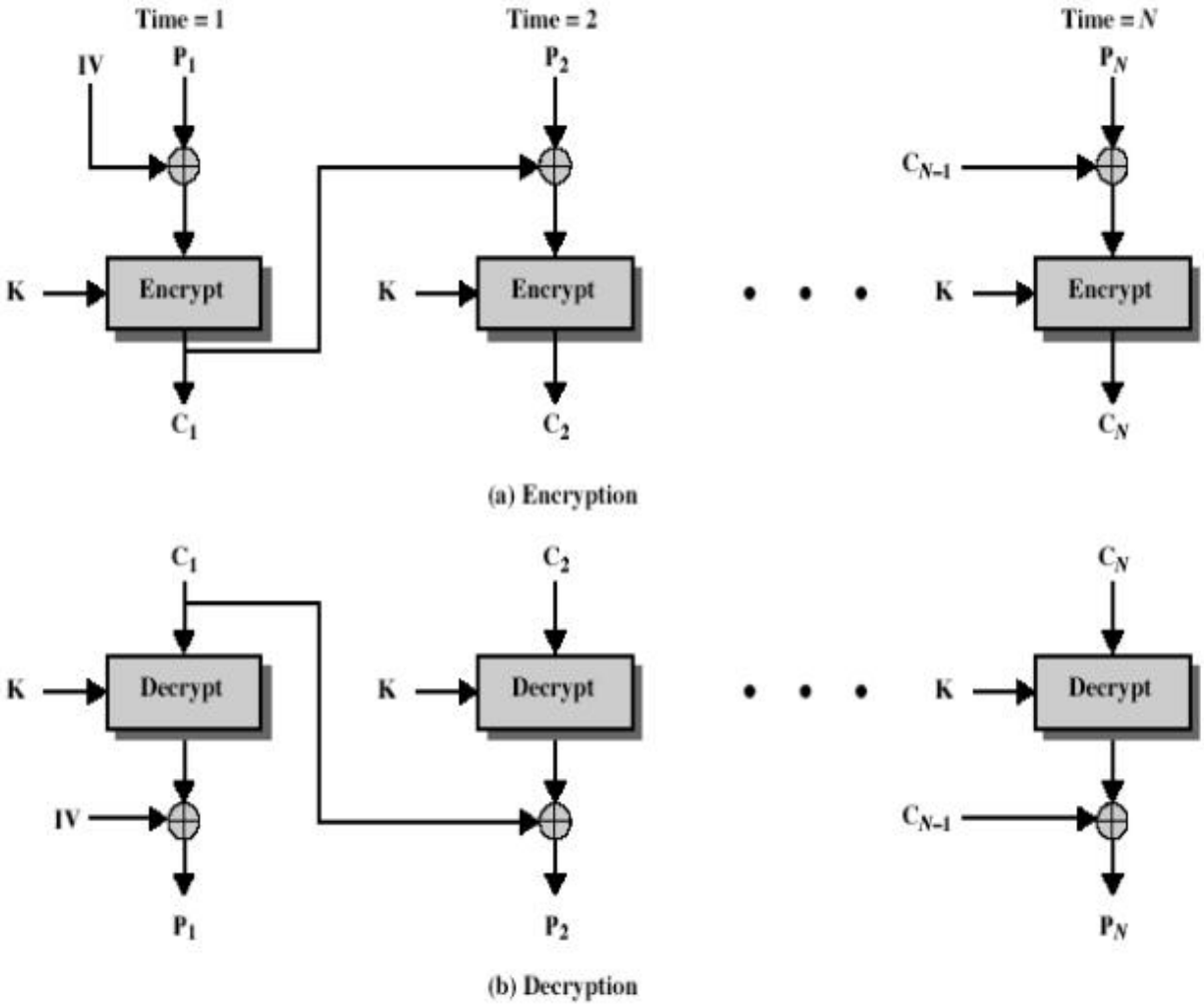
If the message is highly structured, it may be possible for a cryptanalyst to exploit these regularities.

Cipher Block Chaining Mode (CBC) –

I/P= current plaintext block XOR preceding cipher text block

In this scheme, the input to the encryption algorithm is the XOR of the current plaintext **block and the preceding cipher text block; the same key is used for each block.**

There is no relationship between plaintext block.



Advantages :

An appropriate mode for encrypting messages of length greater than b bits. In addition to its use to achieve confidentiality, the CBC mode can be used for authentication.

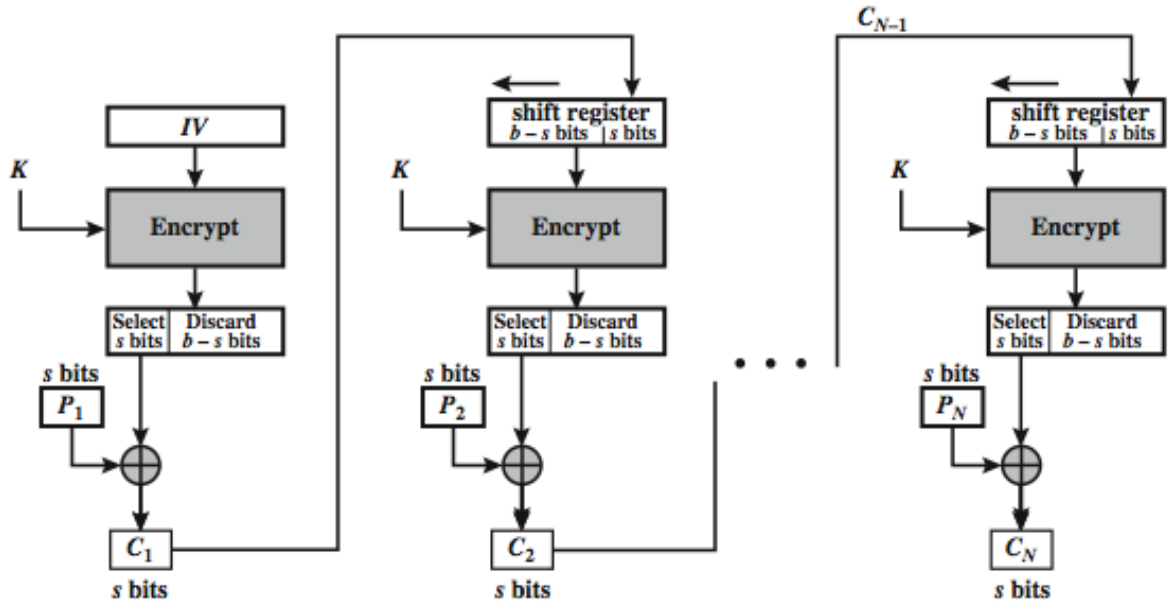
Cipher feedback mode

A stream cipher eliminates the need to pad a message to be an integral number of blocks. It also can operate in real time.

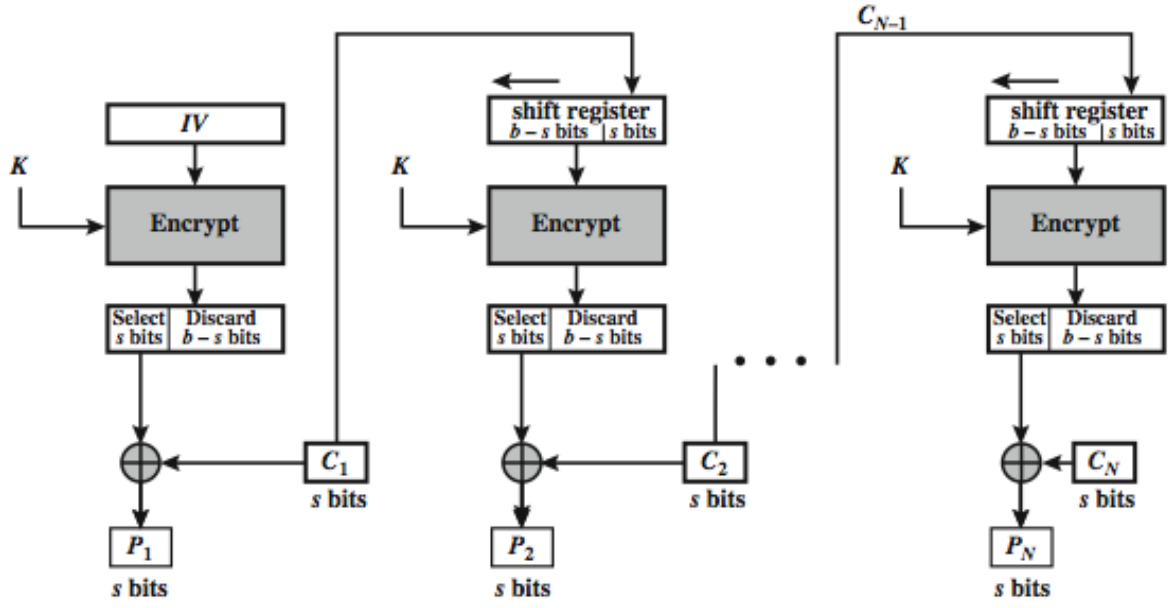
Thus, if a character stream is being transmitted, each character can be encrypted and transmitted immediately using a character-oriented stream cipher. The input to the encryption function is a *b*-bit shift register that is initially set to some initialization vector (IV).

The leftmost (most significant) *s* bits of the output of the encryption function are XORed with the first segment of plaintext *P1* to produce the first unit of ciphertext *C1*. The contents of the shift register are shifted left by s bits and C1 is placed in the rightmost.This process continues until all plaintext units have been encrypted.

For **decryption**, the same scheme is used, except that the received ciphertext unit is XORed with the output of the encryption function to produce the plaintext unit. Note that it is the **encryption function** that is used, not the decryption function. This is easily explained. Let Ss(X) be defined as the most significant s bits of X.



(a) Encryption



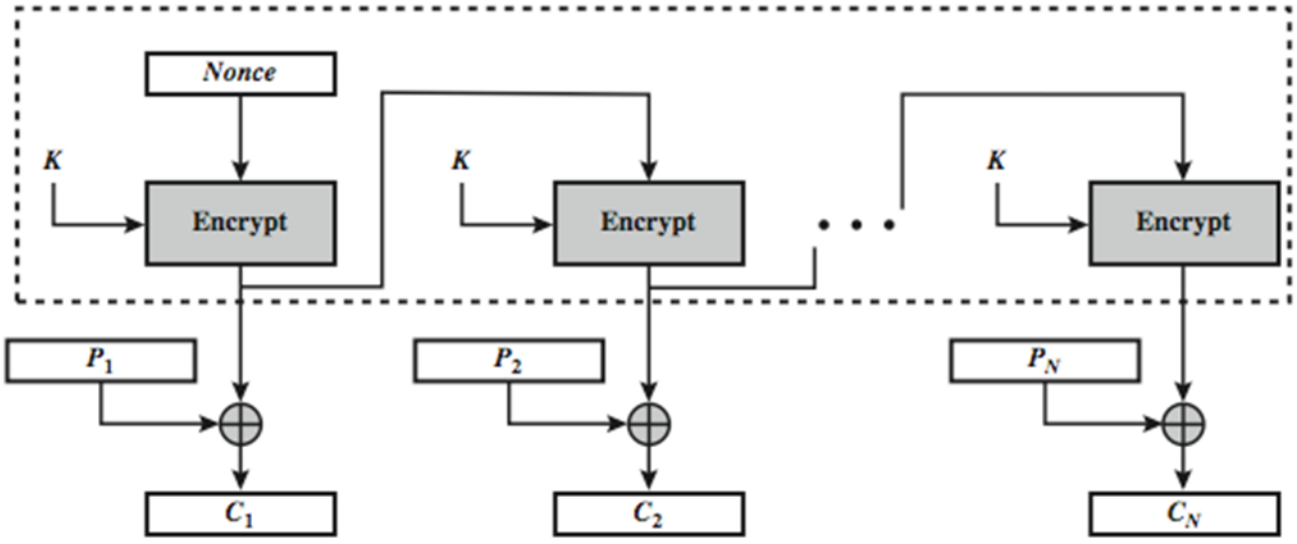
(b) Decryption

Output feedback mode

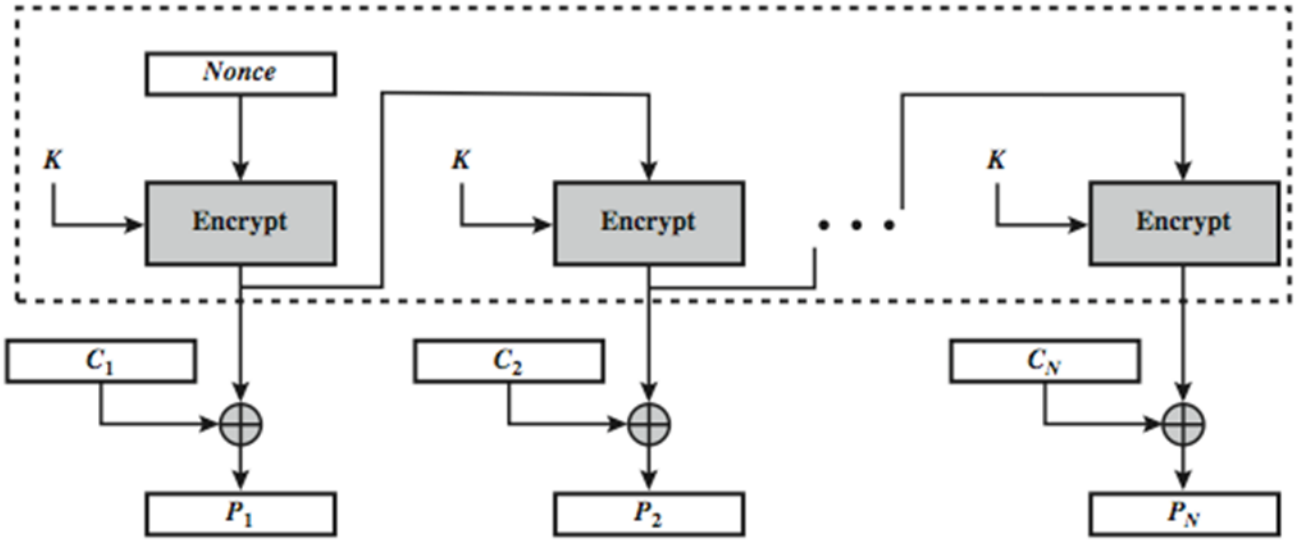
The output feedback (OFB) mode is similar in structure to that of CFB.

The output of the encryption function that is fed back to the shift

register in OFB, whereas in CFB the cipher text unit is fed back to the shift register.



(a) Encryption



(b) Decryption

Advantage :

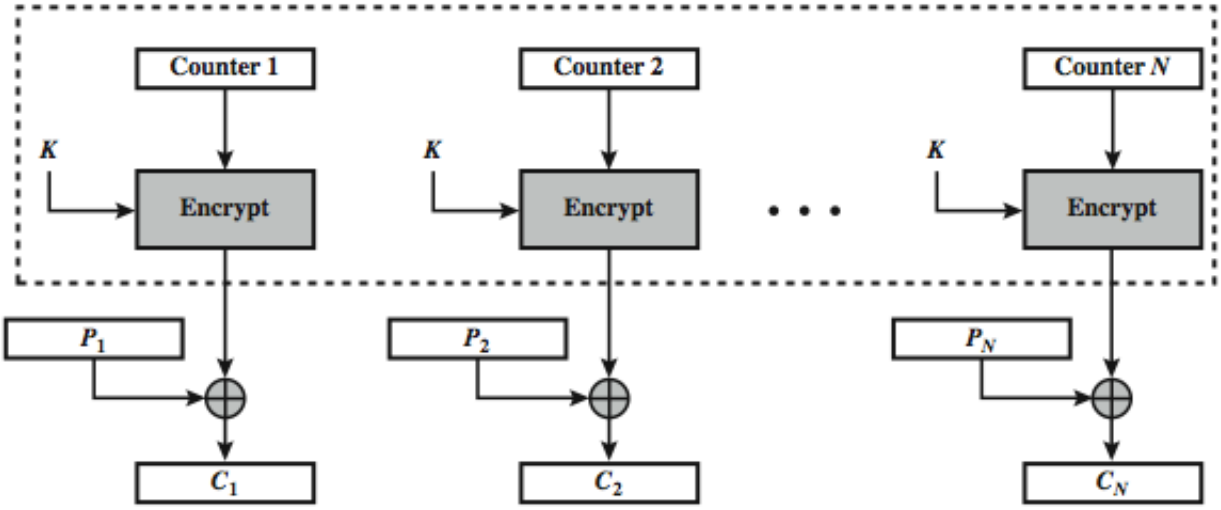
One advantage of the OFB method is that bit errors in transmission do not propagate.

Disadvantage :

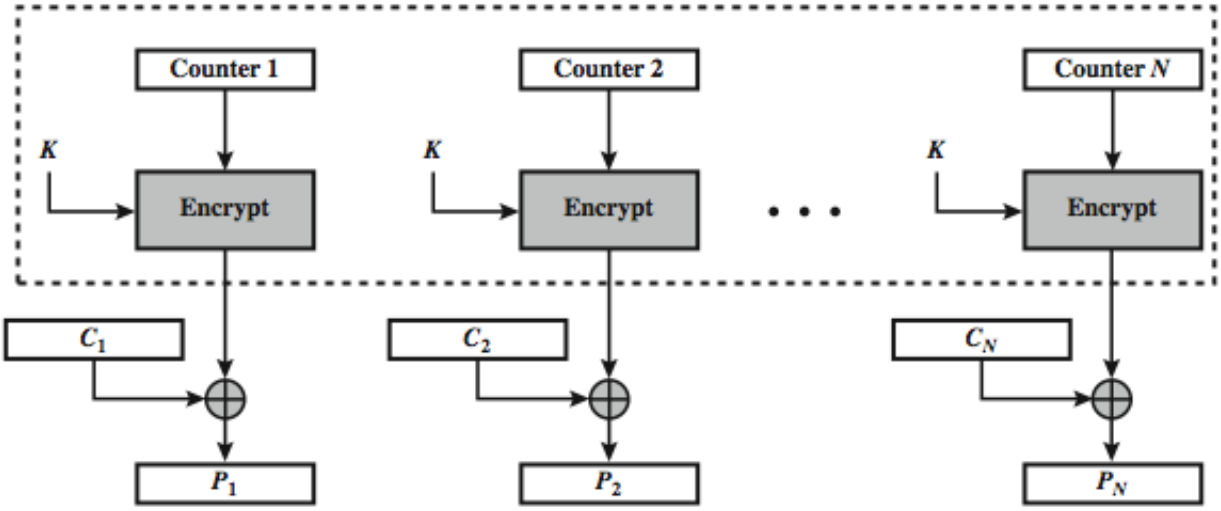
The disadvantage of OFB is that it is more vulnerable to a message stream modification attack than is CFB.

Counter Mode – (CTR)

A counter, equal to the plaintext block size is used. The counter is initialized to some value and then incremented by 1 for each subsequent block. For encryption, the counter is encrypted and then XORed with the plaintext block to produce the cipher text block; there is no chaining. For decryption, the same sequence of counter values is used, with each encrypted counter XORed with a cipher text block to recover the corresponding plaintext block.



(a) Encryption



(b) Decryption

Advantages :

Hardware efficiency: Unlike the three chaining modes, encryption (or decryption) in CTR mode can be done in parallel on multiple blocks of plaintext or cipher text.

Software efficiency: Similarly, because of the opportunities for parallel execution processors that support parallel features can be utilized

Preprocessing: The execution of the underlying encryption algorithm does not depend on input of the plaintext or cipher text.

Random access: The ith block of plaintext or ciphertext can be processed in random-access fashion.

cryptosystems

KEY MANAGEMENT

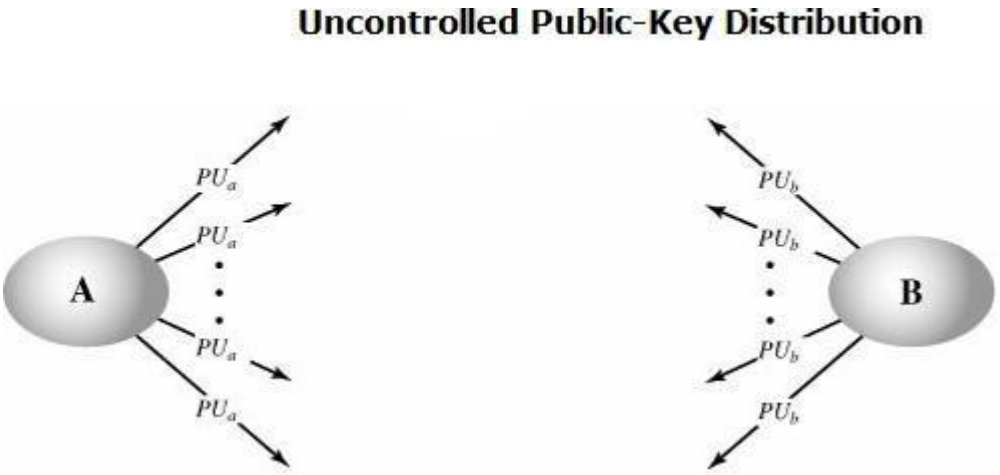
One of the major roles of public-key encryption has been to address the problem of key distribution. Two distinct aspects to use of public key encryption are present.

- The distribution of public keys.
- Use of public-key encryption to distribute secret keys.

Distribution of Public Keys The most general schemes for distribution of public keys are given below

PUBLIC ANNOUNCEMENT OF PUBLIC KEYS

Here any participant can send his or her public key to any other participant or broadcast the key to the community at large. For example, many PGP users have adopted the practice of appending their public key to messages that they send to public forums.

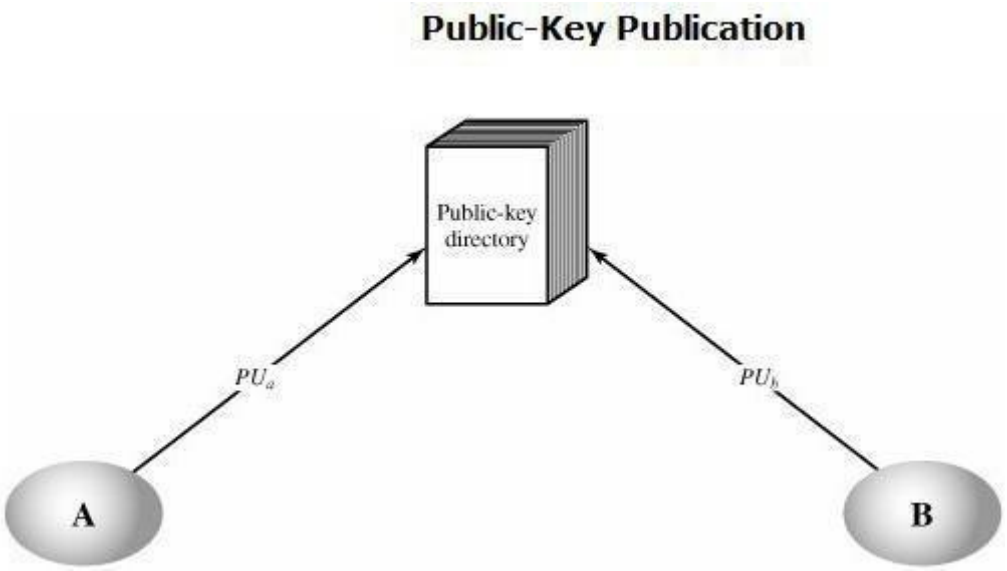


Though this approach seems convenient, it has a major drawback. Anyone can forge such a public announcement. Some user could pretend to be user A and send a public key to another participant or broadcast such a public key. Until the time when A discovers about the forgery and alerts other participants, the forger is able to read all encrypted messages intended for A and can use the forged keys for authentication.

PUBLICLY AVAILABLE DIRECTORY

A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys. Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organization. It includes the following elements:

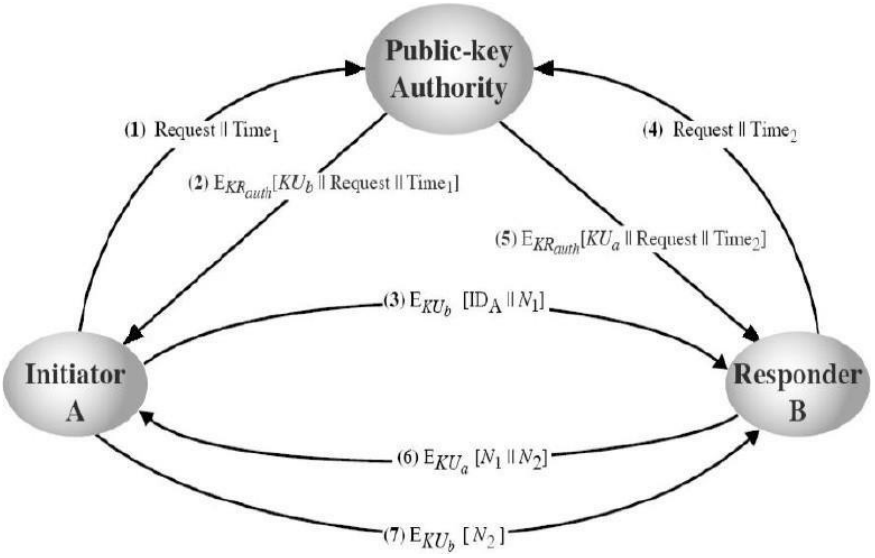
1. The authority maintains a directory with a {name, public key} entry for each participant.
2. Each participant registers a public key with the directory authority. Registration would have to be in person or by some form of secure authenticated communication.



3. A participant may replace the existing key with a new one at any time, either because of the desire to replace a public key that has already been used for a large amount of data, or because the corresponding private key has been compromised in some way.
4. Participants could also access the directory electronically. For this purpose, secure, authenticated communication from the authority to the participant is mandatory. This scheme has still got some vulnerabilities. If an adversary succeeds in obtaining or computing the private key of the directory authority, the adversary could authoritatively pass out counterfeit public keys and subsequently impersonate any participant and eavesdrop on messages sent to any participant. Or else, the adversary may tamper with the records kept by the authority.

PUBLIC-KEY AUTHORITY

Stronger security for public-key distribution can be achieved by providing tighter control over the distribution of public keys from the directory. This scenario assumes the existence of a public authority (whoever that may be) that maintains a dynamic directory of public keys of all users. The public authority has its own (private key, public key) that it is using to communicate to users. Each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key. For example, consider that Alice and Bob wish to communicate with each other and the following steps take place and are also shown in the figure below:



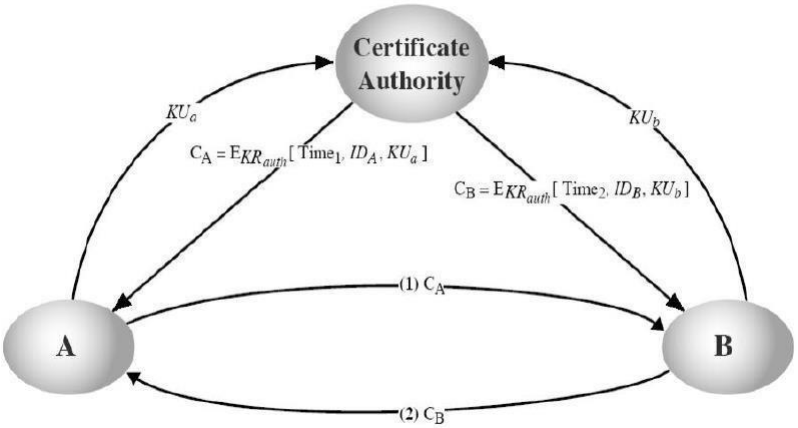
1.) Alice sends a **timestamped** message to the central authority with a request for Bob's public key (the time stamp is to mark the moment of the request)
2.) The authority sends back a message encrypted with its private key (for authentication) –message contains Bob's public key and the original message of Alice – this way Alice knows this is not a reply to an old request;
3.) Alice starts the communication to Bob by sending him an encrypted message containing her identity ID_A and a nonce N_1 (to identify uniquely this transaction)
4.) Bob requests Alice's public key in the same way (step 1)
5.) Bob acquires Alice's public key in the same way as Alice did. (Step-2)
6.) Bob replies to Alice by sending an encrypted message with N_1 plus a new generated nonce N_2 (to identify uniquely the transaction)
7.) Alice replies once more encrypting Bob's nonce N_2 to assure bob that its correspondent is Alice
Thus, a total of seven messages are required. However, the initial four messages need be used only infrequently because both A and B can save the other's public key for future use, a technique known as caching. Periodically, a user should request fresh copies of the public keys of its correspondents to ensure currency.

PUBLIC-KEY CERTIFICATES

The above technique looks attractive, but still has some drawbacks. For any communication between any two users, the central authority must be consulted by both users to get the newest public keys i.e. the central authority must be online 24 hours/day. If the central authority goes offline, all secure communications get to a halt. This clearly leads to an undesirable bottleneck. A further improvement is to use certificates, which can be used to exchange keys without contacting a public-key authority, in a way that is as reliable as if the keys were obtained directly from a public-key authority. A certificate binds an **identity** to **public key**, with all contents **signed** by a trusted Public-Key or Certificate Authority (CA). A user can present his or her public key to the authority in a secure manner, and obtain a certificate. The user can then publish the certificate. Anyone needed this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature. A participant can also convey its key information to another by transmitting its certificate. Other participants can verify that the certificate was created by the authority. This certificate issuing scheme does have the following

requirements:

1. Any participant can read a certificate to determine the name and public key of the certificate's owner.
2. Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
3. Only the certificate authority can create and update certificates.
4. Any participant can verify the currency of the certificate.



Application must be in person or by some form of secure authenticated communication. For participant A, the authority provides a certificate of the form **$C_A = E(PR_{auth}, [T||ID_A||PU_a])$** where PR_{auth} is the private key used by the authority and T is a timestamp. A may then pass this certificate on to any other participant, who reads and verifies the certificate as follows: $D(PU_{auth}, C_A) = D(PU_{auth}, E(PR_{auth}, [T||ID_A||PU_a]))$

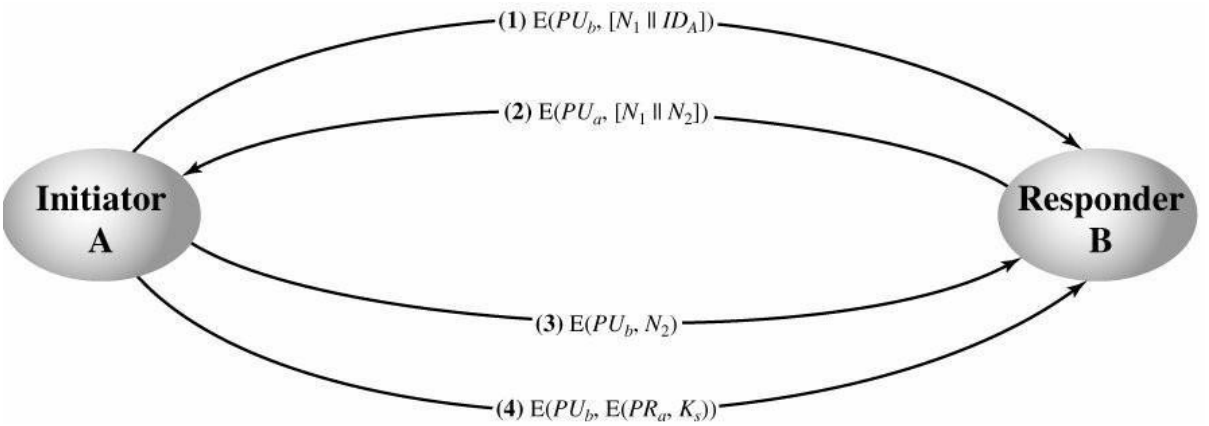
$= (T||ID_A||PU_a)$ The recipient uses the authority's public key, PU_{auth} to decrypt the certificate. Because the certificate is readable only using the authority's public key, this verifies that the certificate came from the certificate authority. The elements ID_A and PU_a provide the recipient with the name and public key of the certificate's holder. The timestamp T validates the currency of the certificate. The timestamp counters the following scenario. A's private key is learned by an adversary. A generates a new private/public key pair and applies to the certificate authority for a new certificate. Meanwhile, the adversary replays the old certificate to B. If B then encrypts messages using the compromised old public key, the adversary can read those messages. In this context, the compromise of a private key is comparable to the loss of a credit card. The owner cancels the credit card number but is at risk until all possible communicants are aware that the old credit card is obsolete. Thus, the timestamp serves as something like an expiration date. If a certificate is sufficiently old, it is assumed to be expired.

One scheme has become universally accepted for formatting public-key certificates: the

X.509 standard. X.509 certificates are used in most network security applications, including IP security, secure sockets layer (SSL), secure electronic transactions (SET), and S/MIME.

SECRET KEY DISTRIBUTION WITH CONFIDENTIALITY AND AUTHENTICATION

It is assumed that A and B have exchanged public keys by one of the schemes described earlier. Then the following steps occur:



1. A uses B's public key to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1), which is used to identify this transaction uniquely.
2. B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as a new nonce generated by B (N_2) Because only B could have decrypted message (1), the presence of N_1 in message (2) assures A that the correspondent is B.
3. A returns N_2 encrypted using B's public key, to assure B that its correspondent is A.
4. A selects a secret key K_s and sends M = E(PU_b, E(PR_a, K_s)) to B. Encryption of this message with B's public key ensures that only B can read it; encryption with A's private key ensures that only A could have sent it.
5. B computes D(PU_a, D(PR_b, M)) to recover the secret key.

The result is that this scheme ensures both confidentiality and authentication in the exchange of a secret key.

THE CHINESE REMAINDER THEOREM

The Chinese Remainder Theorem says it is possible to reconstruct integers in certain range from their residues modulo a set of pair wise relatively prime moduli.

$x \equiv a_1 \bmod n_1$, $x \equiv a_2 \bmod n_2$, $x \equiv a_k \bmod n_k$

If $n_1, n_2, .., n_k$ are positive integers that are pairwise co-prime and $a_1, a_2, \ldots, a_k$ are any integers, then CRT is used to find the values of x that solves the following congruence simultaneously.

Value of $x = (a_1 m_1 y_1 + a_2 m_2 y_2 + \ldots + a_k m_k y_k) \bmod M$

Where $M = n_1 n_2 n_3 .. n_k$

$m_i = M / n_i$

$m_i y_i = 1 \bmod n_i$

Problem 1

$x \equiv 1 \bmod 5$

$x \equiv 2 \bmod 6$

$x \equiv 3 \bmod 7$

$a_1 = 1$

$a_2 = 2$

$a_3 = 3$

$n_1 = 5$

$n_2 = 6$

$n_3=7$

$M=n_1n_2n_3$

$M=5*6*7=210$

$m_i=M/n_i$

$m1=210/5=42$

$m2=210/6=35$

$m3=210/7=30$

$m_iy_i=1 \bmod n_i$

$42y_1=1 \bmod 5$

$y_1=3 \bmod 5$

$35y_2= 1 \bmod 6$

$y_2=5 \bmod 6$

$30y_3=1 \bmod 7$

$y_3=4 \bmod 7$

$x=(a_1m_1y_1+a_2m_2y_2+ a_3m_3y_3) \bmod M$

$=((1*42*3)+(2*35*5)+(3*30*4)) \bmod 210$

$=836 \bmod 210$

$=206$

Problem 2

A bag has contained number of pens if you take out 3 pens at a time 2 pens are left. If you take out 4 pens at a time 1 pen is left and if you take out 5 pens at a time 3 pens are left in the bag. What is the number of pens in the bag.

$x \equiv 2 \bmod 3$

$x \equiv 1 \bmod 4$

$x \equiv 3 \bmod 5$

$a_1=2$

$a_2=1$

$a_3=3$

$n_1=3$

$n_2=4$

$n_3=5$

$M=n_1n_2n_3$

$M=3*4*5=60$

$m_i=M/n_i$

$m1=60/3=20$

m2=60/4=15

m3=60/5=12

$m_i y_i=1 \bmod n_i$

$20y_1=1 \bmod 3$

$y_1=2 \bmod 3$

$15y_2= 1 \bmod 4$

$y_2=3 \bmod 4$

$12y_3=1 \bmod 5$

$y_3=3 \bmod 5$

$x=(a_1m_1y_1+a_2m_2y_2+ a_3m_3y_3) \bmod M$

$=((2*20*2)+(1*15*3)+(3*12*3)) \bmod 60$

$=233 \bmod 60$

$=53$

RSA – Algorithm

Definition:

Block cipher asymmetric algorithm developed by Rivest, Shamir & Adleman . It is the best known & widely used public-key scheme and based on exponentiation in a finite (Galois) field over integers modulo a prime. Its security due to cost of factoring large numbers

Factorization takes $O(e^{\log n \log \log n})$ operations (hard)

Each user will be provided with pair of keys one of which is public key used for encryption and the other is private used for decryption. Plaintext and cipher text are integers between 0 and n – 1 for some n. (eg . 1024 bits)

Ingredients of RSA Algorithm

The ingredients are the following:

p, q, two prime numbers	(private, chosen)
n = pq	(public, calculated)
e, with gcd(ϕ (n),e) = 1; $1 < e < \phi(n)$	(public, chosen)
$d \equiv e^{-1} (\mathrm{mod}\ \phi(n))$	(private, calculated)

RSA Key Setup:

This key setup is done once (rarely) when a user establishes (or replaces) their public key.

1. Each user generates a public/private key pair by: selecting two large primes at random - p, q
2. Computing their system modulus N=p .q
3. ø (N)= (p-1) (q-1)
4. Selecting at random the encryption key e where 1< e < ø(N), gcd (e ,ø (N)) =1
5. Solve following equation to find decryption key d
   e. d=1 mod ø(N)
6. Publish their public encryption key: KU = {e, N}
7. Keep secret private decryption key: KR = {d, p, q}

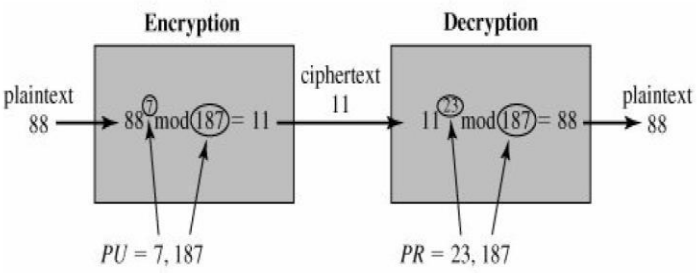
RSA Use

8. To encrypt a message M the sender:
   - obtains public key of recipient KU={e ,N}
   - computes: $\mathbf{C=M^e \bmod N}$, where $0 \leq M < N$
9. To decrypt the ciphertext C the owner:
   - uses their private key KR={d,p,q}
   - computes: $\mathbf{M=C^d \bmod N}$

Example:

1. Select primes: p=17 & q=11
2. Compute $n = pq$ =17×11=187
3. Compute ø(n)=(p–1)(q-1)=16×10=160
4. Select e *:* gcd(e,160)=1; choose e=7

5. Determine d*: de*=1 mod 160 and *d* < 160 Value is d=23 since 23×7=161
6. Publish public key KU={7,187}
7. Keep secret private key KR={23,17,11}
8. Given message M = 88 (88<187)
9. Encryption:C = 88^7 mod 187 = 11



88^7 mod 187 = [(88^4 mod 187) x (88^2 mod 187) x (88^1 mod 187)] mod 187

88^1 mod 187 = 88

88^2 mod 187 = 7744 mod 187 = 77

88^4 mod 187 = 59969536 mod 187 = 132

88^7 mod 187 = (88 x 77 x 132) mod 187

= 894432 mod 187 = 11

10. Decryption: M = 11^{23} mod 187 = 88

Note : Finding private key d (ie) multiplicative inverse of e^{-1} using extended Euclidean algorithm) ie) d ≡ e^{-1} mod Φ(n)

d* e ≡ 1 mod Φ(n) Here d * 3 ≡ 1 mod 160

According Extended Euclidean algorithm initial values

A1 = A3 = 1 A2 = 0 160

B1 = 0 B2 = 1 B3 = 7

Find Q = ⌊ A3/B3⌋ (take lowest nearest integer)

Then A1 = B1 ; A2= B2 ; A3 = B3

B1 = A1+QB1 ; B2 = A2+QB2; B3 = A3-QB3

Q	A1	A2	A3	B1	B2	B3
	1	0	160	0	1	7
22	0	1	7	1	22	6
1	1	22	6	1	23	1

Since B3 = 1 ;

Multiplicative inverse B2 = 23

d * 3 ≡ 1 mod 160

23 * 7 ≡ 1 mod 160

d = 23

Computational aspects of RSA

This includes i) Encryption / decryption ii) Key generation.

Both encryption and decryption in RSA involve raising an integer to an integer power, mod *n*. we can make use of a property of modular arithmetic: [(*a* mod *n*) x (*b* mod *n*)] mod *n* = (*a* x *b*) mod *n*

Efficient Operation Using the Public Key

To speed up the operation of the RSA algorithm using the public key, a specific choice of *e* is usually made. The most common choice is 65537 two other popular choices are 3 and 17.

Key generation :

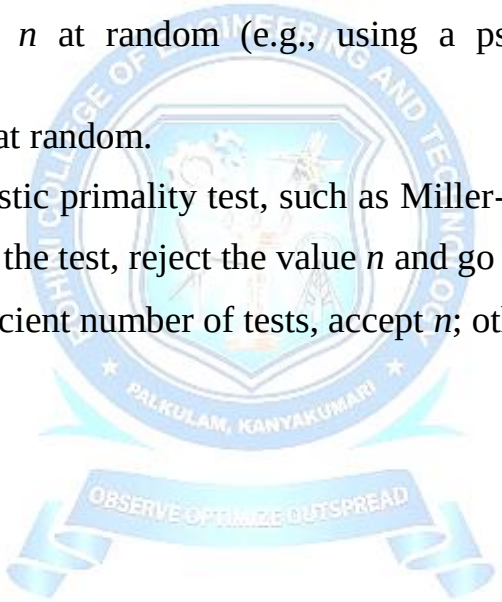
Each participant must generate a pair of keys.

This involves the following tasks:

- Determining two prime numbers, *p* and *q*
- Selecting either *e* or *d* and calculating the other

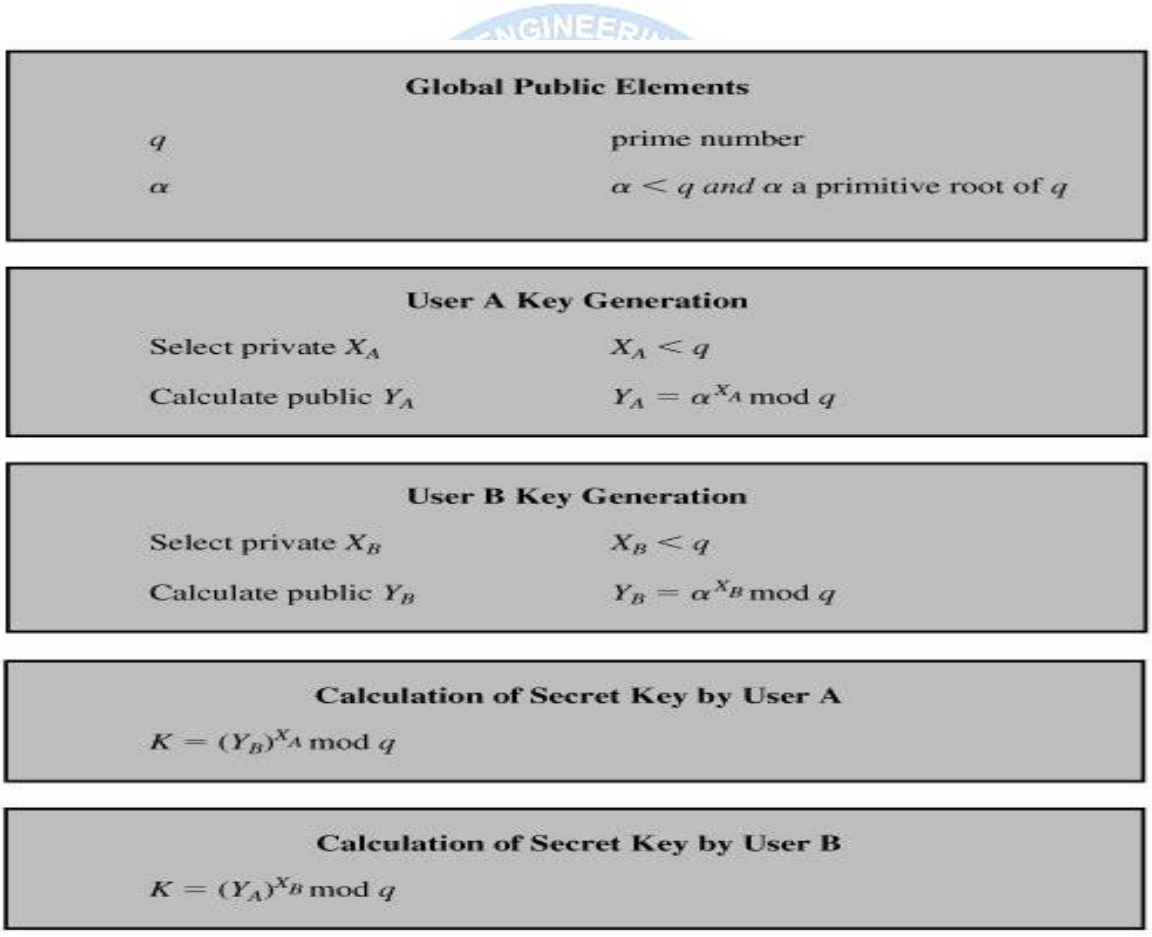
---

1. Pick an odd integer *n* at random (e.g., using a pseudorandom number generator).
2. Pick an integer $a < n$ at random.
3. Perform the probabilistic primality test, such as Miller-Rabin, with *a* as a parameter. If *n* fails the test, reject the value *n* and go to step 1.
4. If *n* has passed a sufficient number of tests, accept *n*; otherwise, go to step 2.

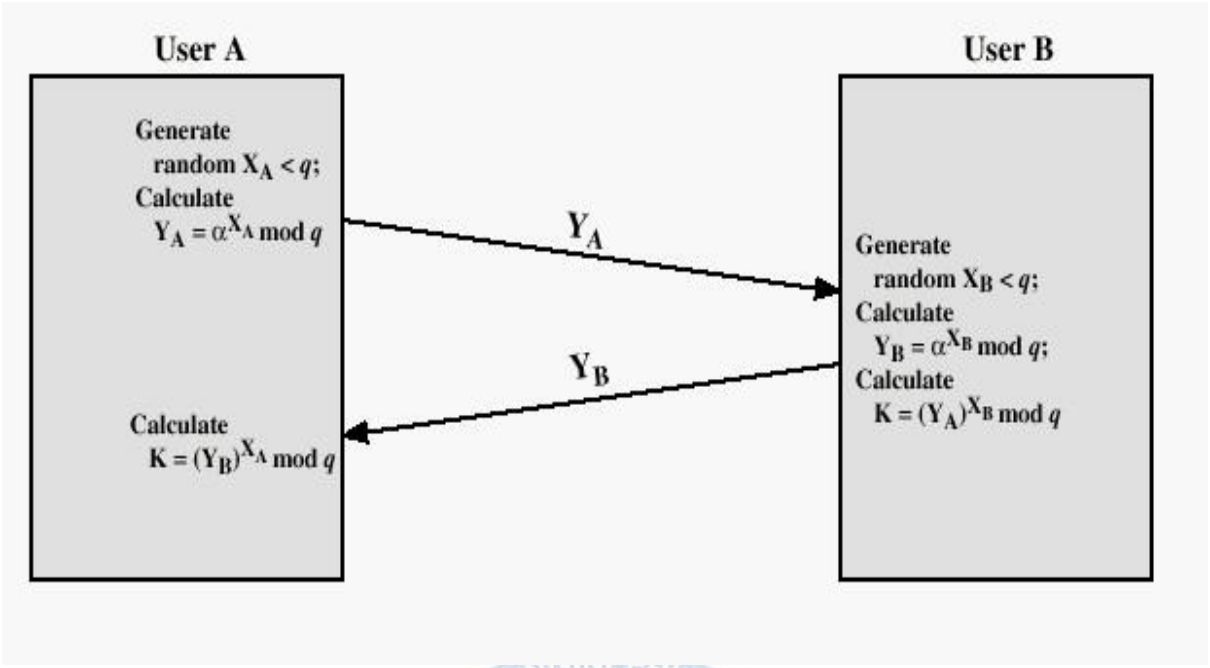


Diffie–Hellman Key Exchange

- The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages.
- The algorithm itself is limited to the exchange of secret values. A number of commercial products employ (use) this key exchange technique.
- The Diffie-Hellman algorithm uses exponentiation in a finite (Galois) field (modulo a prime or a polynomial), and depends for its effectiveness on the difficulty of computing discrete logarithms.



The result is that the two sides have exchanged a secret value.



Ex: ὰ = 3 XA= 97 and X^B=233

A computes YA = 397 mod 353 = 40.

B computes YB = 3233mod 353 = 248.

After they exchange public keys, each can compute the common secret key: A computes

K= (YB) ^{XA} mod 353 = $\mathbf{248^{97}mod}$ 353 =160.

B computes K= (YA) ^{XB} mod 353 = 40^{233} mod 353 = 160.

Man-in-the-Middle Attack

Suppose Alice and Bob wish to exchange keys, and Darth is the adversary. The attack proceeds as follows:

1. Alice sends an encrypted message M : E(K2, M).
2. Darth intercepts the encrypted message and decrypts it, to recover M.
3. Darth sends Bob E(K1,M) or E(K1,M'),where M' is any message. In the first case, Darth simply wants to eavesdrop on the communication without altering it. In the second case, Darth wants to modify the message going to Bob.

This vulnerability can be overcome with the use of digital signatures and public-key Certificates.

FERMAT'S THEOREM

Fermat's theorem states the following: if p is a prime and a is a positive integer not divisible by p, then

$$\mathbf{a^{p-1} \equiv 1 (mod\ p)}$$

Consider the set of positive integers less than p: {1,2,3..p-1}

Multiply each element by *a modulo p* to get the set

X ={ a mod p,2a mod p….(p-1)mod p}.

None of the elements of X is equal to zero because p does not divide a. No two of the integers in X are equal.(p-1) elements of X are all positive integers with no two elements are equal. Multiplying the numbers in both sets and taking the result mod p yields.

$$a * 2a *\ldots*(p-1)a \equiv [(1*2*\ldots*(p-1)](mod\ p)$$

$$\{1 * 2 *\ldots*(p-1)\}\ a^{p-1} \equiv [(1*2*\ldots*(p-1)](mod\ p)$$

$$(p-1)!\ a^{p-1} \equiv (p-1)!(mod\ p)$$

$$\mathbf{a^{p-1} \equiv 1 (mod\ p)}$$

Example

$a = 7, p = 19$

$7^2 = 49 \equiv 11\ (mod\ 19)$

$7^4 = 121 \equiv 7\ (mod\ 19)$

$7^8 \equiv 49 \equiv 11\ (mod\ 19)$

$7^{16} \equiv 121$ K $7\ (mod\ 19)$

$a^{p-1} = 7^{18} = 7^{16} * 7^2 \equiv 7 * 11 \equiv 1$ (mod 19

An alternative form of Fermat's theorem is also useful: If p is prime and a is a positive integer, then

$$a^p \equiv a \pmod{p}$$

Euler's totient function

It is represented as *ø(n).*Euler's totient function is defined as the number of positive integers less than *n* and relatively prime to *n*. *ø(1)=1*

It should be clear that for a prime number *p*

ø(p)=p-1

Suppose that we have two prime numbers *p* and *q*, with *p* not equal to *q*. Then we can show that

n=pq.

ø(n)= ø(pq)= ø(p)* ø(q)=(p-1)*(q-1)

ø(n)=(pq-1)-[(q-1)+(p-1)]

= pq-(p+q)+1

=(p-1)*(q-1)

= ø(p)* ø(q)

To determine f(35), we list all of the positive integers less than 35 that are relatively prime to it:

1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18

19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34

There are 24 numbers on the list, so f(35) = 24.

f(21) = f(3) * f(7) = (3 - 1) * (7 - 1) = 2 * 6 = 12

Elliptic curve cryptography [ECC]

- Elliptic curve cryptography [ECC] is a **public-key** cryptosystem just like RSA, Rabin, and El Gamal.
- Every user has a **public** and a **private** key.
  - Public key is used for encryption/signature verification.
  - Private key is used for decryption/signature generation.
- Elliptic curves are used as an extension to other current cryptosystems.
  - Elliptic Curve Diffie-Hellman Key Exchange
  - Elliptic Curve Digital Signature Algorithm

ECC- Algorithm

- Both parties agree to some publicly-known data items
  - The **elliptic curve equation** $y^2 = x^3 + ax + b \bmod p$
    - values of ***a*** and ***b*** such that $4a^3 + 27b^2 \neq 0$
    - prime, ***p***
  - The **elliptic group** is computed from the elliptic curve equation
  - A **base point**, G, taken from the elliptic group
- Each user generates their public/private key pair
  - Private Key = an integer, x selected from the interval [1, p-1]
  - Public Key = product of private key and base point

    (Product = x*G)

Example :

- Suppose Alice wants to send to Bob an encrypted message.

- Both agree on a base point, G.
- Alice and Bob create public/private keys.
  - Alice : Private Key = n_A

    Public Key = $P_A = n_A * G$
  - Bob : Private Key = n_B

    Public Key = $P_B = n_B * G$
- Alice takes plaintext message, M, and encodes it onto a point, P_M, from the elliptic group.

Encryption : Alice choose another random k – value from { 1,2,… p-1 }

Cipher text : $C_m = \{ KG, P_m + KP_B \}$

Decryption : by Bob

Take the first point from Cm - KG

Multiply KG and private key of Bob : Product = n_B KG

Take the second point from Cm and subtract the product from it

$P_m + KP_B - n_B$ KG

Substitute $P_B = n_B * G$ Then $P_m + K n_B * G - n_B$ KG = Pm

ECC is particularly beneficial for application where:

- computational power is limited (wireless devices, PC cards)
- integrated circuit space is limited (wireless devices, PC cards)
- High speed is required.
- Intensive use of signing, verifying or authenticating is required.

- Signed messages are required to be stored or transmitted (especially for short messages).
- Bandwidth is limited (wireless communications and some computer networks). Advantages:
- Shorter key lengths
  - Encryption, Decryption and Signature Verification speed up
  - Storage and bandwidth savings

SECURE HASH ALGORITHM (SHA)

The most widely used hash function has been the Secure Hash Algorithm (SHA). SHA was developed by the National Institute of Standards and Technology (NIST) and published as a federal information processing standard (FIPS 180) in 1993.SHA is based on the hash function MD4, and its design closely models MD4. Three new versions of SHA, with hash value lengths of 256, 384, and 512 bits, known as SHA-256, SHA-384, and SHA-512, respectively. Collectively, these hash algorithms are known as SHA-2.

SHA-512 Logic

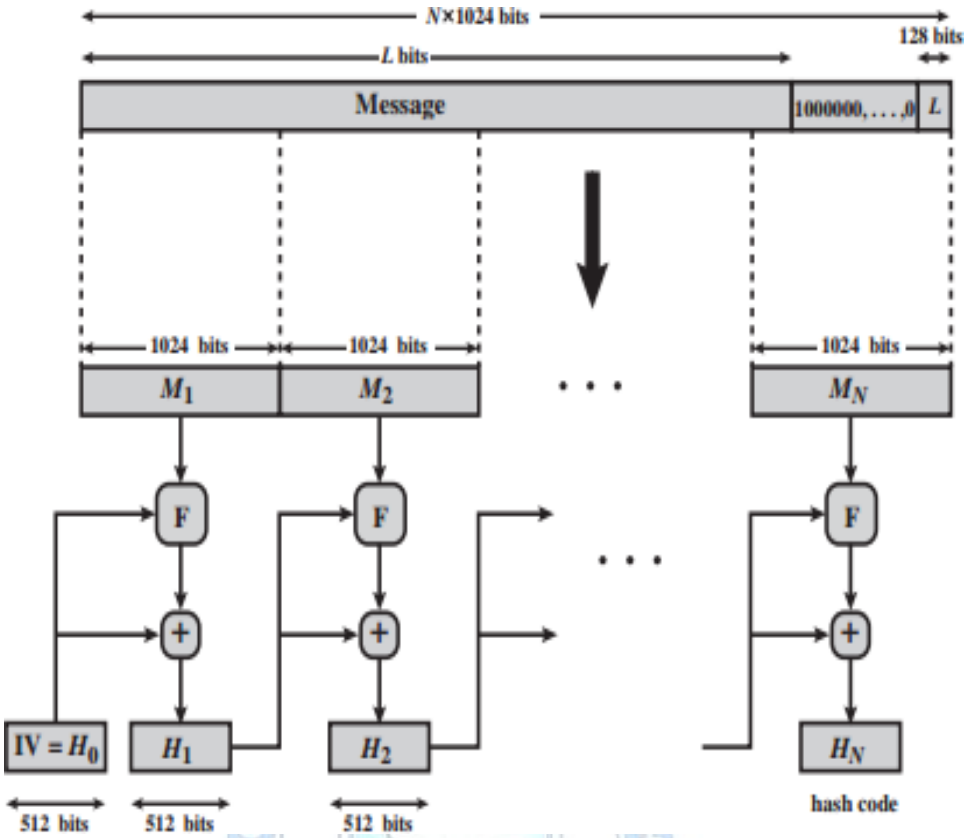
Message Digest Generation Using SHA-512

The algorithm takes as input a message with a maximum length of less than bits 2^{128} bits and produces as output a 512-bit message digest. The input is processed in 1024-bit blocks. The processing consists of the following steps.

Step 1 Append padding bits. The message is padded so that its length is congruent to 896 modulo 1024 of 1 to 1024. The padding consists of a single 1 bit followed by the necessary number of 0 bits.

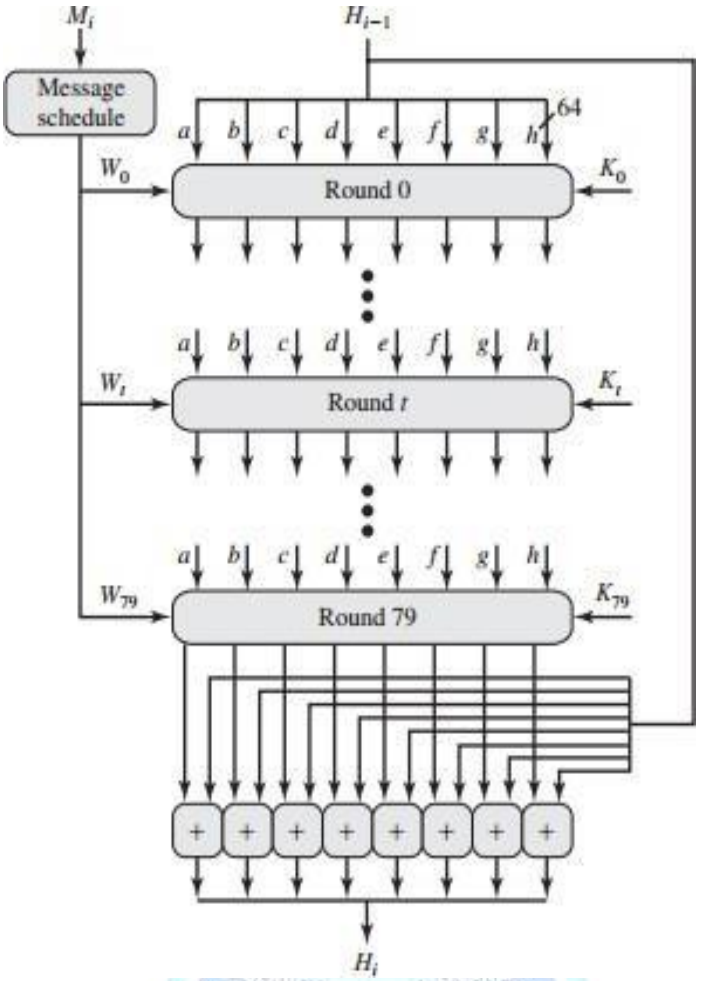
Step 2 Append length. A block of 128 bits is appended to the message. This block is treated as an unsigned 128-bit integer (most significant byte first) and contains the length of the original message (before the padding).

The outcome of the first two steps yields a message that is an integer multiple of 1024 bits in length. The expanded message is represented as the sequence of 1024-bit blocks *M*1, *M*2, Á , *MN* , so that the total length of the expanded message is *N* * 1024 bits



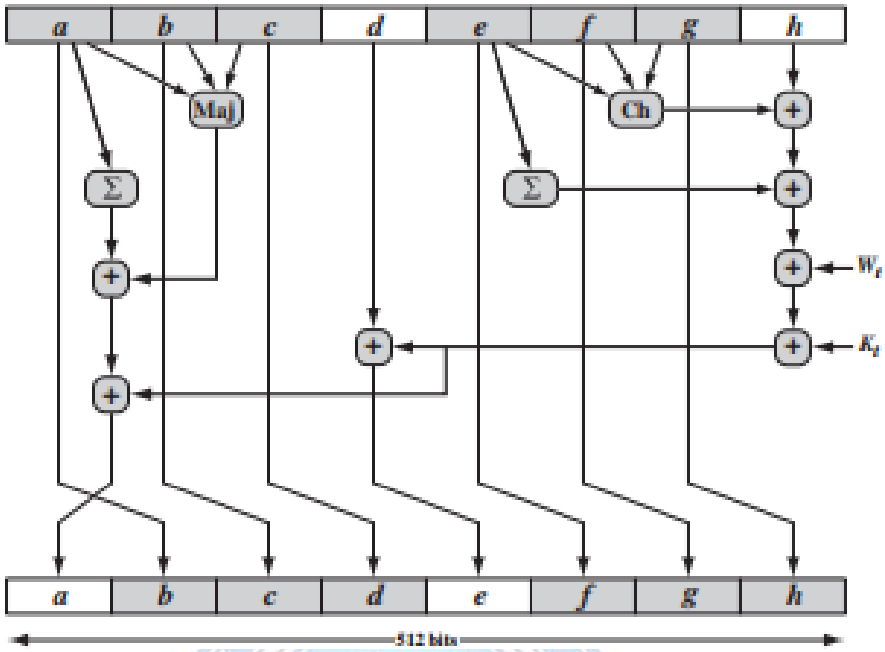
Step 3 Initialize hash buffer. A 512-bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as eight 64-bit registers (a, b, c, d, e, f, g, h).These registers are initialized to the following 64-bit integers (hexadecimal values) These values are stored in **big-endian** format, which is the most significant byte of a word in the low-address (leftmost) byte position.

Step 4 Process message in 1024-bit (128-word) blocks. The heart of the algorithm is a module that consists of 80 rounds; Each round takes as input the 512-bit buffer value, ABCDEFGH, and updates the contents of the buffer. At input to the first round, the buffer has the value of the intermediate hash value, *Hi*-1).



SHA-512 Processing of a Single 1024-Bit Block

Step 5 Output. After all 1024-bit blocks have been processed, the output from the N th stage is the 512- bit message digest. Thus, in the first 16 steps of processing, the value of W_t is equal to the corresponding word in the message block. For the remaining 64 steps, the value of W_t consists of the circular left shift by one bit of the XOR of four of the preceding values of W_t , with two of those values subjected to shift and rotate operations.

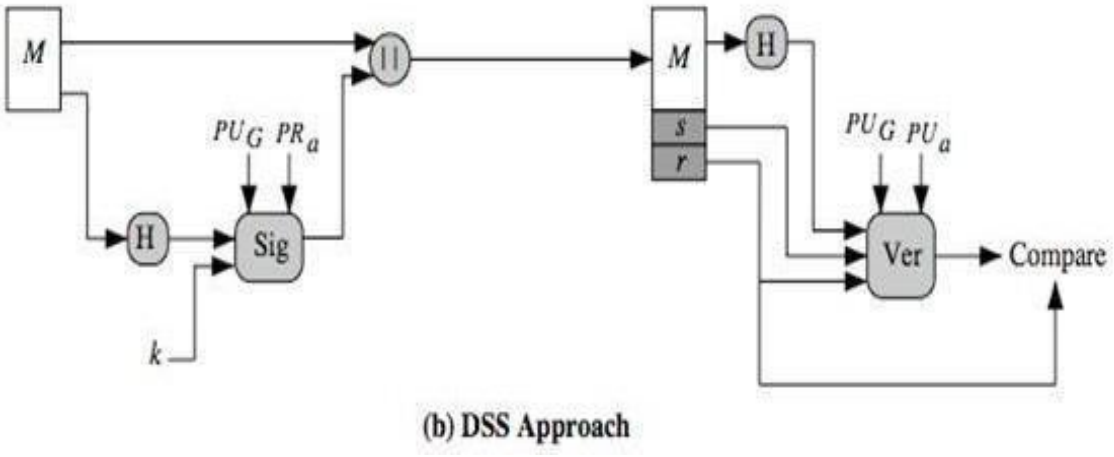


Compression Function

DSS APPROACH

DSS uses an algorithm that is designed to provide only digital signature function. Unlike RSA, it cannot be used for encryption or key exchange.

The DSS approach makes use of a hash function. The hash code is provided as input to a signature function along with a random number k generated for this particular signature. The signature function also depends on the sender's private key (PRa) and the global public key (PUG) The result is a signature consisting of two components, labeled *s* and *r*. At the receiving end, the hash code of the incoming message is generated. This plus the signature is



(b) DSS Approach

input to a verification function.

The verification function also depends on the global public key as well as the sender's public key (PUa), which is paired with the sender's private key. The output of the verification function is a value that is equal to the signature component if the signature is valid. The signature function is such that only the sender, with knowledge of the private key, could have produced the valid signature.

THE DIGITAL SIGNATURE ALGORITHM

1. Global Public key Components

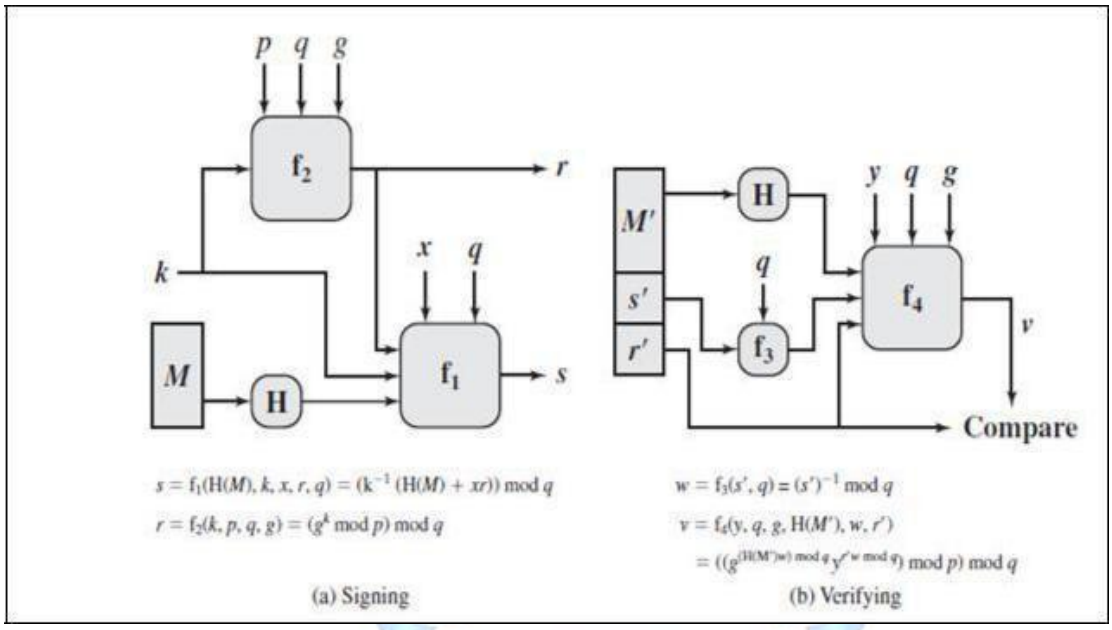
p- prime no. where $2^{L-1} < p < 2^{L}$ for

$512 \leq L \leq 1024$ q –prime divisor of (p-1

) where 2

$g = h^{(p-1)/q} \bmod p$

where h is any integer with $1 < h < (p-1)$ such that $h^{(p-1)/q} \bmod p > 12\ 160$



2. User's Private key

x - random or pseudo random integer with $0 < x < q$

3. User's

Public key

$y = g^{x} \bmod p$

4. User's Per Message Secret Number

k = random or pseudo random integer with $0 < k < q$

DSA Signature Creation

To sign a message M the sender: the sender generates a random signature key k, k<qComputes signature pair:

$r = (g^k \bmod p) \bmod q$

$s = [k^{-1}(H(M) + xr)] \bmod q$

Signature = (r,s)

DSA Signature Verification

After received M & signature (r,s)

Verify a signature, recipient computes: $w = (s')^{-1} \bmod q$

u1= [H(M')w

]mod q u2=

(r'w)mod q

$v = [(g^{u1} y^{u2}) \bmod p] \bmod q$

If v=r then signature is verified.

ELGAMAL DIGITAL SIGNATURES

Elgamal signature scheme involves the use of private key for encryption and public key for decryption

The global elements of Elgamal digital signature are prime number q and a, which is the primitive rootof q.

1. Global Public key Components

q - prime no.

a – primitive root of q

---

2. User A signs a message M to B by computing

- Generate a random integer XA, such that $1 < X_A < q-1$
- Compute $\mathbf{Y}A = a^{X_A} \bmod q$
- A's Private key is XA
- A's Public key is YA

To sign a message M, user A first computes the hash m=H(M), such that m is an integer in the range $0 \leq m \leq (q-1)$

3. User A generates the digital signature

- Choose a random integer K, such that $1 \leq K \leq (q-1)$ and gcd(K,q-1) = 1. That is, K isrelatively prime to q-1.
- Compute, S1 = a K mod q
- Compute K -1 mod q-1
- Compute, S2 = K-1(m-xAS1) mod (q-1)
- The signature consists of a pair (S1,S2)

2. User B verifies the Signature

$V1 = a^m \bmod q$

$V2 = (YA)^{S1} (S1)^{S2} \bmod q$

The signature is valid if V1 = V2.

Example I

GlobalElement

q=19 and a=10

Alice computes the private and public key

- Alice computes her key:
  - Alice chooses Private key, XA=16
  - Computes Public Key, $Y_A=10^{16} \bmod 19 = 4$
- Alice signs message with hash m=14
  - Alice chooses K=5 which is relatively prime to q-1=18
  - Compute $S1 = 10^5 \bmod 19 = 3$
  - Compute $K^{-1} \bmod (q-1) = 5^{-1} \bmod 18 = 11$
  - Compute S2 = 11(14-16*3) mod 18 = -374 mod 18=4 {-374 mod 18=18-374%18}
- B can verify the signature by computing
  - $V1 = 10^{14} \bmod 19 = 16$
  - $V2 = 4^3.3^4 = 5184 = 16 \bmod 19$
  - Since 16 = 16 signature is verified and valid.

Any other user can verify the signature as follows.

1. Compute $x' = a^y v^e \bmod p$.
2. Verify that e = H(M|| x').

To see that the verification works, observe that $x' \equiv a^y v^e \equiv a^y a^{-se} \equiv a^{y-se} \equiv a^r \equiv x \pmod p$

Hence, H(M|| x`) = H(M||x)

Authentication applications – Kerberos

Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users. Kerberos relies exclusively on conventional encryption, making no use of public-key encryption.

Requirements for Kerberos:

Secure: A network eavesdropper should not be able to obtain the necessary information to impersonate a user. More generally, Kerberos should be strong enough that a potential opponent does not find it to be the weak link.

Reliable: Kerberos should be highly reliable and should employ a distributed server architecture, with one system able to back up another.

Transparent: Ideally, the user should not be aware that authentication is taking place, beyond the requirement to enter a password.

Scalable: The system should be capable of supporting large numbers of clients and servers.

Kerberos Version-4

A simple authentication dialogue

In an unprotected network environment, any client can apply to any server for service. The security risk is that of impersonation. To counter this threat, servers must be able to confirm the identities of clients who request service. But in an open environment, this makes burden on each server.

• An alternative is to use an authentication server (AS) that knows the passwords of all users and stores these in a centralized database. In addition, AS shares a unique secret key with each server.

The simple authentication dialogue is as follows:

1. C ->AS: IDc||Pc||IDv

2. AS -> C: Ticket

3. C ->V: IDc||Ticket

Ticket= E(Kv,[IDc||ADc||IDv])

Where, C: Client, AS: Authentication Server, V: Server, IDc : ID of the client,

Pc:Password of the client, ADc: Address of client, IDv : ID of the server,

Kv: secret key shared by AS and V, ||: concatenation

AS issues the ticket only if the client is authentic.

A more secure authentication dialogue

There are two major problems associated with the previous approach:

– Plaintext transmission of the password.

– Each time a user has to enter the password.

To solve these problems, we introduce a scheme for avoiding plaintext passwords, and a new server, known as ticket granting server (TGS).

Once per user logon session:

1. C ->AS: IDc||IDtgs

2. AS -> C: E(Kc,Tickettgs)

Once per type of service:

3. C ->TGS: IDc||IDv||Tickettgs

4. TGS -> C: ticketv

Once per service session:

5. C ->V: IDc||ticketv

Tickettgs=E(Ktgs,[IDc||ADc||IDtgs||TS1||Lifetime1])

Ticketv= E(Kv,[IDc||ADc||IDv||TS2||Lifetime2])

1. The client requests a ticket-granting ticket on behalf of the user by sending its user's ID to the AS, together with the TGS ID.

2. The AS responds with a ticket that is encrypted with a key that is derived from the user's password.

Because only the correct user should know the password, only the correct user can recover the ticket. Password is not transmitted in plain text.

3. The client requests a service-granting ticket on behalf of the user. For this purpose, the client transmits a message to the TGS containing the user's ID, the ID of the desired service, and the ticket-granting ticket.

4. The TGS decrypts the incoming ticket and verifies. It checks to make sure that the lifetime has not expired. If the user is permitted to access to the server V, the TGS issues a ticket to grant access to the requested service.

If the user wants access to the same service at a later time, the client can simply use the previously acquired service-granting ticket and need not bother the user for a password.

5. The client requests access to a service on behalf of the user. For this purpose, the client transmits a message to the server containing the user's ID and the service-granting ticket. The server authenticates by using the contents of the ticket.

Kerberos V4 Authentication Dialogue Message Exchange

• Two additional problems remain in the more secure authentication dialogue:

1. Lifetime associated with the ticket granting ticket. If the lifetime is very short, then the user will be repeatedly asked for a password. If the lifetime is long, then the opponent has the greater opportunity for replay.

2. Requirement for the servers to authenticate themselves to users.

Kerberos Realms and Multiple Kerberi

A full-service Kerberos environment consisting of a Kerberos server, a number of clients, and a number of application servers requires the following:

1. The Kerberos server must have the user ID and hashed passwords of all participating users in its database. All users are registered with the Kerberos server.

2. The Kerberos server must share a secret key with each server. All servers are registered with the Kerberos server.

Such an environment is referred to as a Kerberos realm.

- A Kerberos realm is a set of managed nodes that share the same Kerberos database. The Kerberos database resides on the Kerberos master computer system, which should be kept in a physically secure room
- A read-only copy of the Kerberos database might also reside on other Kerberos computer systems.
- A related concept is that of a Kerberos principal, which is a service or user that is known to the Kerberos system.
- Each Kerberos principal is identified by its principal name.
- Principal names consist of three parts: a service or user name, an instance name, and a realm name
- Networks of clients and servers under different administrative organizations typically constitute different realms.

Kerberos provides a mechanism for supporting interrealm authentication. For two realms to support interrealm authentication, a third requirement is added:

3. The Kerberos server in each interoperating realm shares a secret key with the server in the other realm. The two Kerberos servers are registered with each other.

- The scheme requires that the Kerberos server in one realm trust the Kerberos server in the other realm to authenticate its users. Furthermore, the participating servers in the second realm must also be willing to trust the Kerberos server in the first realm.

Kerberos version 5

Version 5 of Kerberos provides a number of improvements over version 4.

provides improvements over v4

– addresses environmental shortcomings

– and technical deficiencies

Differences between version 4 and 5 Version 5 is intended to address the limitations of version 4 in two areas:

Environmental shortcomings

- Encryption system dependence
- Internet protocol dependence
- Message byte ordering
- Ticket lifetime
- Authentication forwarding
- Inter-realm authentication

Technical deficiencies

- Double encryption
- PCBC encryption
- Session keys
- Password attacks

The version 5 authentication dialogue

First, consider the authentication service exchange. Message (1) is a client request for a ticket-granting ticket. As before, it includes the ID of the user and the TGS. The following new elements are added:

Realm: Indicates realm of user

Options: Used to request that certain flags be set in the returned ticket

Times: Used by the client to request the following time settings in the ticket:

from: the desired start time for the requested ticket

till: the requested expiration time for the requested ticket

rtime: requested renew-till time

Nonce: A random value to be repeated in message (2) to assure that the response is fresh and has not been replayed by an opponent

---

The authenticator includes several new fields as follows:

Subkey: The client's choice for an encryption key to be used to protect this specific application session. If this field is omitted, the session key from the ticket (Kc,v) is used.

Sequence number: An optional field that specifies the starting sequence number to be used by the server for messages sent to the client during this session. Messages may be sequence numbered to detect replays.

X.509 Authentication services

X.509 defines a framework for the provision of authentication services by the X.500 directory to its users. The directory may serve as a repository of public-key certificates. Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority. In addition, X.509 defines alternative authentication protocols based on the use of public-key certificates.

X.509 is based on the use of public-key cryptography and digital signatures

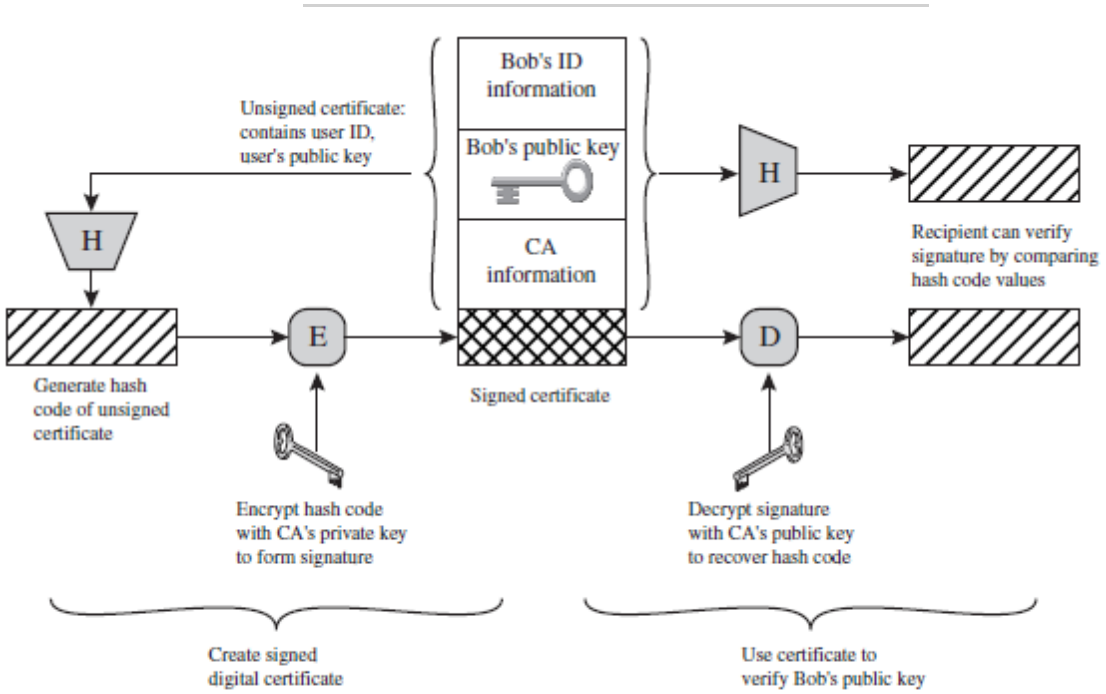


Figure 14.14 Public-Key Certificate Use

Certificates

The heart of the X.509 scheme is the public-key certificate associated with each user. These user certificates are assumed to be created by some trusted certification authority (CA) and placed in the directory by the CA or by the user.

Version: Differentiates among successive versions of the certificate format; the default is version 1. If the Issuer Unique Identifier or Subject Unique Identifier are present, the value must be version 2. If one or more extensions are present, the version must be version 3.

Serial number: An integer value, unique within the issuing CA, that is unambiguously associated with this certificate.

Signature algorithm identifier: The algorithm used to sign the certificate, together with any associated parameters. Because this information is repeated in the Signature field at the end of the certificate, this field has little, if any, utility.

Issuer name: X.500 name of the CA that created and signed this certificate.
Period of validity: Consists of two dates: the first and last on which the certificate is valid.

Subject name: The name of the user to whom this certificate refers. That is, this certificate certifies the public key of the subject who holds the corresponding private key.

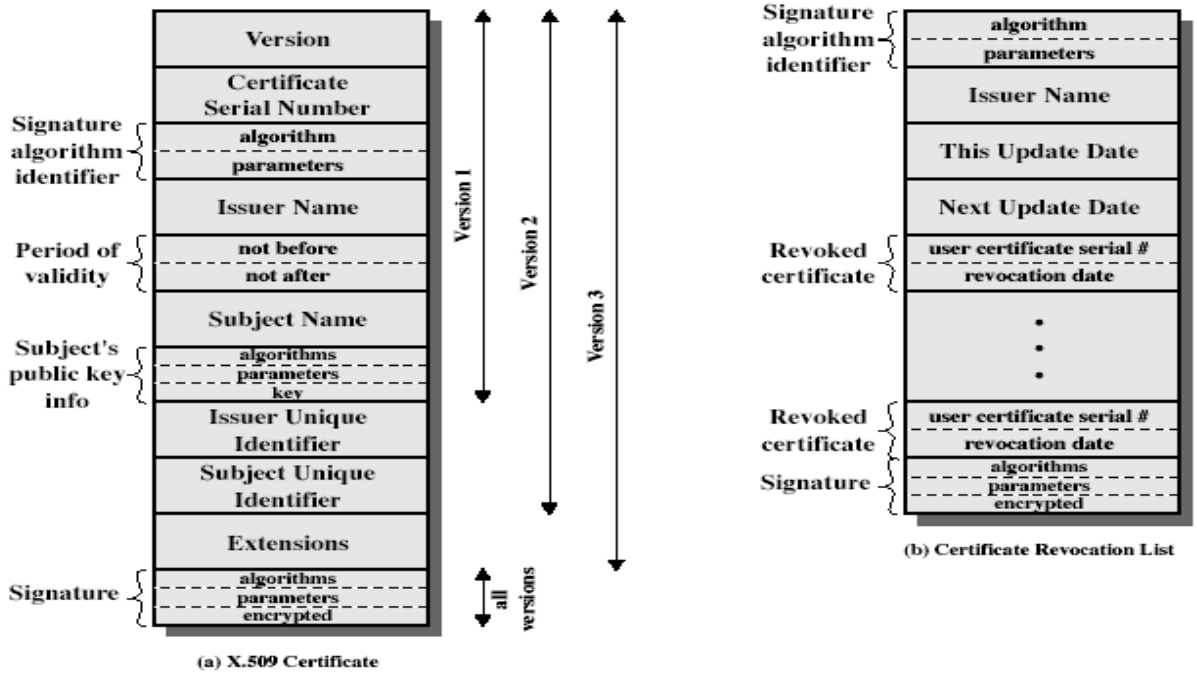
Subject's public-key information: The public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters.

Issuer unique identifier: An optional bit string field used to identify uniquely the issuing CA in the event the X.500 name has been reused for different entities.

Subject unique identifier: An optional bit string field used to identify uniquely the subject in the event the X.500 name has been reused for different entities.

Extensions: A set of one or more extension fields. Extensions were added in version 3 and are discussed later in this section.

Signature: Covers all of the other fields of the certificate; it contains the hash code of the other fields, encrypted with the CA's private key. This field includes the signature algorithm identifier.



- notation CA<<A>> denotes certificate for A signed by CA

Obtaining a Certificate

- Any user with access to CA can get any certificate from it
- Only the CA can modify a certificate
- Because cannot be forged, certificates can be placed in a public directory

CA Hierarchy

- If both users share a common CA then they are assumed to know its public key
- Otherwise CA's must form a hierarchy
- Use certificates linking members of hierarchy to validate other CA's
  - each CA has certificates for clients (forward) and parent (backward)
- Each client trusts parents certificates
- Enable verification of any certificate from one CA by users of all other CAs in hierarchy

- Forward certificates: Certificates of X generated by other CAs
- Reverse certificates: Certificates generated by X that are the certificates of other CAs

CA Hierarchy Use

In the example given below , user A can acquire the following certificates from the directory to establish a certification path to B:

X<<W>> W <<V>> V <<Y>> <<Z>> Z <<B>>

When A has obtained these certificates, it can unwrap the certification path in sequence to recover a trusted copy of B's public key.

Using this public key, A can send encrypted Messages to B. If A wishes to receive encrypted messages back from B, or to sign messages sent to B, then B will require A's public key, which can be obtained from the following certification path:

Z<<Y>> Y <<V>> V <<W>> W <<X>>X <<A>>

B can obtain this set of certificates from the directory, or A can provide them as part of its initial message to B.

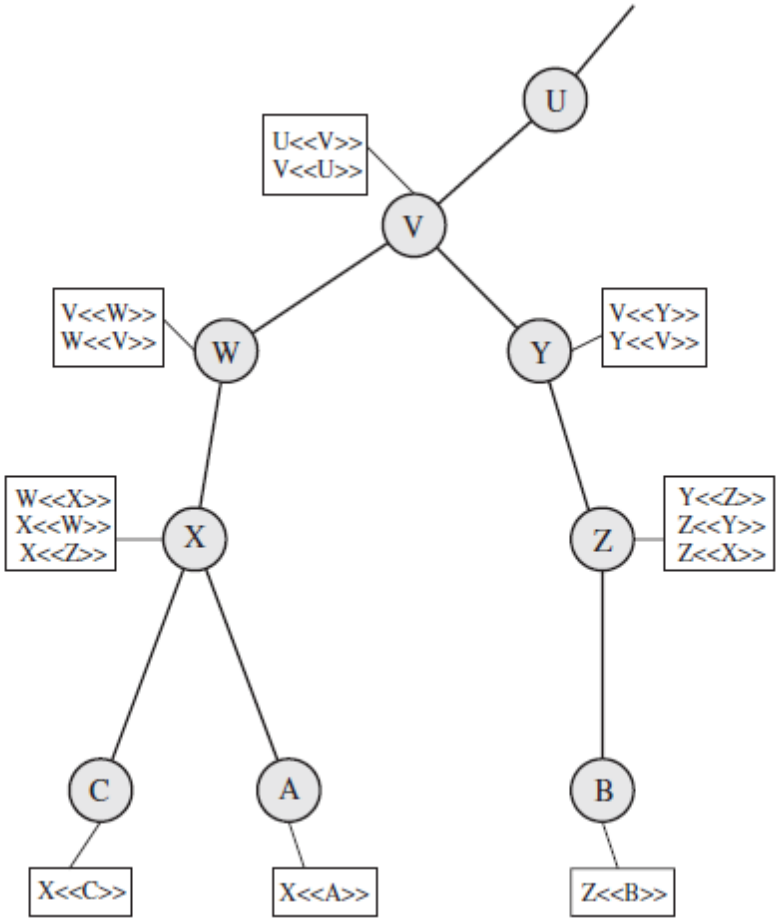


Figure 14.16 X.509 Hierarchy: A Hypothetical Example

Certificate Revocation

• Certificates have a period of validity
• may need to revoke before expiry, for the following reasons eg:

1. User's private key is compromised
2. User is no longer certified by this CA
3. CA's certificate is compromised

CA's maintain list of revoked certificates, the Certificate Revocation List (CRL).Users should - check certificates with CA's CRL.

CYBER CRIME & INFORMATION SECURITY

Cyber Crime

Meaning – Criminal activities carried out by means of computers or the internet.

Definition –

- Cybercrime is defined as a crime where a computer is the object of the crime or is used as a tool to commit an offense.
- A cybercriminal may use a device to access a user's personal information, confidential business information, government information, or disable a device.
- Cybercrime, also called computer crime, the use of a computer as an instrument to further illegal ends, such as committing fraud, trafficking in child pornography and intellectual property, stealing identities, or violating privacy.
- Cybercrime, especially through the Internet, has grown in importance as the computer has become central to commerce, entertainment, and government.
- Cyber crime or computer-oriented crime is a crime that includes a computer and a network. The computer may have been used in the execution of a crime or it may be the target.

Cyber crime encloses a wide range of activities, but these can generally be divided into two categories:

a) Crimes that aim computer networks or devices. These types of crimes involve different threats (like virus, bugs etc.) and denial-of-service attacks.

b) Crimes that use computer networks to commit other criminal activities. These types of crimes include cyber stalking, financial fraud or identity theft.

5.2 Classification of Cyber Crimes

Email spoofing

- Email spoofing is a form of cyber attack in which a hacker sends an email that has been manipulated to seem as if it originated from a trusted source.
- For example, a spoofed email may pretend to be from a well-known shopping website, asking the recipient to provide sensitive data, such as a password or credit card number.
- Alternatively, a spoofed email may include a link that installs malware on the user's device if clicked.

- An example of spoofing is when an email is sent from a false sender address, that asks the recipient to provide sensitive data.
- This email could also contain a link to a malicious website that contains malware.

Spamming

- Spamming is the use of electronic messaging systems like e-mails and other digital delivery systems and broadcast media to send unwanted bulk messages indiscriminately.
- The term spamming is also applied to other media like in internet forums, instant messaging, and mobile text messaging, social networking spam, junk fax transmissions, television advertising and sharing network spam.
- Spam is any kind of unwanted, unsolicited digital communication that gets sent out in bulk. Often spam is sent via email, but it can also be distributed via text messages, phone calls, or social media.

Cyber defamation

- The tort of cyber defamation is an act of intentionally insulting, defaming or offending another individual or a party through a virtual medium.
- It can be both written and oral.
- Defamation means giving an "injury to the reputation of a person" resulting from a statement which is false. The term defamation is used in the section 499 of Indian Penal Code, 1860.
- Cyber defamation is also known as internet defamation or online defamation in the world of internet and its users.
- Cyber defamation is also known as internet defamation or online defamation in the world of internet and its users.
- Cyber defamation is a new concept but it virtually defames a person through new medium. The medium of defaming the individual's identity is through the help of computers via internet.

Internet time theft

- It refers to the theft in a manner where the unauthorized person uses internet hours paid by another person.
- The authorized person gets access to another person's ISP user ID and password, either by hacking or by illegal means without that person's knowledge.

- Basically, Internet time theft comes under hacking. It is the use by an unauthorized person, of the Internet hours paid for by another person.

Salami Attack

- A salami attack is a small attack that can be repeated many times very efficiently. Thus the combined output of the attack is great.
- In the example above, it refers to stealing the round-off from interest in bank accounts.
- Even though it is less than 1 cent per account, when multiplied by millions of accounts over many months, the adversary can retrieve quite a large amount. It is also less likely to be noticeable since your average customer would assume that the amount was rounded down to the nearest cent.

Data Diddling

- Data diddling is a type of cybercrime in which data is altered as it is entered into a computer system, most often by a data entry clerk or a computer virus.
- Data diddling is an illegal or unauthorized data alteration. Changing data before or as it is input into a computer or output.
- Example: Account executives can change the employee time sheet information of employees before entering to the HR payroll application.

Forgery

When a perpetrator alters documents stored in computerized form, the crime committed may be forgery. In this instance, computer systems are the target of criminal activity.

- The term forgery usually describes a message related attack against a cryptographic digital signature scheme. That is an attack trying to fabricate a digital signature for a message without having access to the respective signer's private signing key.
- Among the many examples of this crime, taking another's work, whether it be written or visual, such as a artwork, and attempting to distribute it as either your own or as an original is an example of forgery.
- Likewise, either creating fake documents or producing counterfeit items is considered to be forgery as well.

Hacking

- Hacking refers to activities that seek to compromise digital devices, such as computers, smartphones, tablets, and even entire networks.

- Hacking is an attempt to exploit a computer system or a private network inside a computer. Simply put, it is the unauthorized access to or control over computer network security systems for some illicit purpose

Email bombing

- An email bomb or "mail bomb" is a malicious act in which a large number of email messages are sent to a single email address in a short period of time. The purpose of an email bomb is typically to overflow a user's inbox. In some cases, it will also make the mail server unresponsive.

Tools and Methods used in Cyber Crime

Proxy Server

- It is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers.
- A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity.
- Proxies were invented to add structure and encapsulation to distributed systems.
- Today, most proxies are web proxies, facilitating access to content on the World Wide Web and providing anonymity

Anonymizer

- An anonymizer or an anonymous proxy is a tool that attempts to make activity on the Internet untraceable.
- It is a proxy server computer that acts as an intermediary and privacy shield between a client computer and the rest of the Internet.
- It accesses the Internet on the user's behalf, protecting personal information by hiding the client computer's identifying information

Phishing

- Phishing is a cybercrime in which a target or targets are contacted by email, telephone or text message by someone posing as a genuine (legal) organization to ensnare individuals into providing sensitive data such as personally identifiable information, banking and credit card details, and passwords.

Keylogger

- Keyloggers are a form of spyware where users are unaware their actions are being tracked. Keyloggers can be used for a variety of purposes; hackers may use them to maliciously gain access to your private information, while employers might use them to monitor employee activities. Spyware is largely invisible software that gathers information about your computer use, including browsing. Key loggers are a form of

spyware that capture every keystroke you type; they can send this information to remote servers, where log-in information--including your passwords--can be extracted and used.

- A keylogger is a tool that captures and records a user's keystrokes. It can record instant messages, email, passwords and any other information you type at any time using your keyboard. Keyloggers can be hardware or software.
- Spyware is any software that installs itself on your computer and starts covertly monitoring your online behaviour without your knowledge or permission. Spyware is a kind of malware that secretly gathers information about a person or organization and relays this data to other parties.

There are two common types of keyloggers.

Software and Hardware keyloggers.

- Software Keyloggers.
- Hardware Keyloggers.
- Spear Phishing.
- Drive-by-Downloads.
- Trojan Horse.
- 2-Step Verification.
- Install Anti Malware Software
- Use Key Encryption Software

Hardware Keyloggers

- Hardware keyloggers are small hardware devices.
- These are connected to the PC and/or to the keyboard and save every keystroke into a file or in the memory of the hardware device.
- Cybercriminals install such devices on ATM machines to capture ATM Cards' PINs.
- Each keypress on the keyboard of the ATM gets registered by these keyloggers.
- These keyloggers look like an integrated part of such systems; hence, bank customers are unaware of their presence.

Software Keyloggers

Software keyloggers are software programs installed on the computer systems which usually are located between the OS and the keyboard hardware, and every keystroke is recorded. Software

keyloggers are installed on a computer system by Trojans or viruses without the knowledge of the user.

Antikeylogger

- Antikeylogger is a tool that can detect the keylogger installed on the computer system and also can remove the tool.
- Advantages of using antikeylogger are as follows:
- Firewalls cannot detect the installations of keyloggers on the systems; hence, antikeyloggers can detect installations of keylogger.
- This software does not require regular updates of signature bases to work effectively such as other antivirus and antispy programs if not updated, it does not serve the purpose, which makes the users at risk.

Spywares

Spyware is a type of malware, that is installed on computers which collects information about users without their knowledge. The presence of Spyware is typically hidden, from the user, it is secretly installed on the user's personal computer. Sometimes, however, Spywares such as keyloggers are installed by the owner of a shared, corporate or public computer on purpose to secretly monitor other users.

Password Cracking

Password cracking is the process of attempting to gain Unauthorized access to restricted systems using common passwords or algorithms that guess passwords. In other words, it's an art of obtaining the correct password that gives access to a system protected by an authentication method.

Password cracking refers to various measures used to discover computer passwords. This is usually accomplished by recovering passwords from data stored in, or transported from, a computer system. Password cracking is done by either repeatedly guessing the password, usually through a computer algorithm in which the computer tries numerous combinations until the password is successfully discovered.

Password cracking can be done for several reasons, but the most malicious reason is in order to gain unauthorized access to a computer without the computer owner's awareness. This results in cybercrime such as stealing passwords for the purpose of accessing banking information. Other, nonmalicious, reasons for password cracking occur when someone has misplaced or forgotten a password.

The purpose of password cracking is as follows:

- To recover a forgotten password
- Testing the strength of a password
- To gain unauthorized access to a system

Manual password cracking is a process of trying out different password combinations and checking if each one of them working or not and is quite a time consuming process. Manual password cracking involves:

1. Find a valid user account
2. Create a list of possible passwords (dictionary)
3. Rank the passwords from high to low probability
4. Key-in each password
5. Try again until a successful password is found

Sometimes password can be guessed with the prior knowledge of the target user's information. Different characteristics of a guessable password are as follows:

- Blank (no password)

- General passwords like password, admin, 123456, etc.
- Series of letters like QWERTY
- User's name or login name
- Name of user's friend/relative/pet
- User's birth date or birth place
- User's vehicle number, office number, residence or mobile number
- Name of a celebrity or idol
- Simple modification of the above mentioned passwords (like adding numbers)

Password Cracking Techniques

Password cracking can be classified into three types:

- Online attacks
- Offline attacks
- Non-electronic attacks (social engineering, shoulder surfing, dumpster diving etc)

SQL Injection

An SQL injection is a type of cyber-attack in which a hacker uses a piece of SQL (Structured Query Language) code to manipulate a database and gain access to potentially valuable information. ... Prime examples include notable attacks against Sony Pictures and Microsoft among others.

SQL injection (SQLi) is a type of cyberattack against web applications that use SQL databases such as IBM Db2, Oracle, MySQL, and MariaDB. As the name suggests, the attack involves the injection of malicious SQL statements to interfere with the queries sent by a web application to its database.

Using SQL injection, a hacker will try to enter a specifically crafted SQL commands into a form field instead of the expected information. The intent is to secure a response from the database that will help the hacker understand the database construction, such as table names.

Steps for SQL Injection Attack

Following are some steps for SQL injection attack:

1. The attacker looks for the webpages that allow submitting data, that is, login page, search page, feedback, etc.

2. To check the source code of any website, right click on the webpage and click on “view source” (if you are using IE – Internet Explorer) – source code is displayed in the notepad. The attacker checks the source code of the HTML, and look for “FORM” tag in theHTML code. Everything between the

<FORM< and </FORM> have potential parameters that might be useful to find the vulnerabilities.

<FORM action=Search/search.asp method=post>

<input type=hidden name=A value=C></FORM>

3. The attacker inputs a single quote under the text box provided on the webpage to accept the user- name and password. This checks whether the user-input variable is sanitized or interpreted literally by the server.

4. The attacker uses SQL commands such as SELECT statement command to retrieve data from the database or INSERT statement to add information to the database

Blind SQL Injection

Blind SQL injection is used when a web application is vulnerable to an SQL injection but the results of the injection are not visible to the attacker. The page with the vulnerability may not be the one that displays data.

Using SQL injections, attackers can:

1. Obtain some basic information if the purpose of the attack is reconnaissance.
2. May gain access to the database by obtaining username and their password.
3. Add new data to the database.
4. Modify data currently in the database

Tools used for SQL Server penetration

1. AppDetectivePro
2. DbProtect
3. Database Scanner
4. SQLPoke
5. NGSSQLCrack
6. Microsoft SQL Server Fingerprint (MSSQLFP) Tool

How to Prevent SQL Injection Attacks

SQL injection attacks occur due to poor website administration and coding. The following steps can be taken to prevent SQL injection.

1. Input validation
2. Modify error reports
3. Other preventions

Cloud Security

Cloud computing which is one of the most demanding technology of the current time, starting from small to large organizations have started using cloud computing services. Where there are different types of cloud deployment models are available and cloud services are provided as per requirement like that internally and externally security is maintained to keep the cloud system safe. Cloud computing security or cloud security is an important concern which refers to the act of protecting cloud environments, data, information and applications against unauthorized access, DDOS attacks, malwares, hackers and other similar attacks.

Community Cloud : These allow to a limited set of organizations or employees to access a shared cloud computing service environment.

Planning of security in Cloud Computing:

As security is a major concern in cloud implementation, so an organization have to plan for security based on some factors like below represents the three main factors on which planning of cloud security depends.

- Resources that can be moved to the cloud and test its sensitivity risk are picked.
- The type of cloud is to be considered.
- The risk in the deployment of the cloud depends on the types of cloud and service models.

Types of Cloud Computing Security Controls :

There are 4 types of cloud computing security controls i.e.

1. **Deterrent Controls** : Deterrent controls are designed to block nefarious attacks on a cloud system. These come in handy when there are insider attackers.
2. **Preventive Controls** : Preventive controls make the system resilient to attacks by eliminating vulnerabilities in it.
3. **Detective Controls** : It identifies and reacts to security threats and control. Some examples of detective control software are Intrusion detection software and network security monitoring tools.
4. **Corrective Controls** : In the event of a security attack these controls are activated. They limit the damage caused by the attack.