Information Security

UNIT - I

Security Attack:

→ Any action that compromises the security of information owned by an Organization is security attack

→ Security attacks is classified into two classification

1. Passive attack.
2. Active attack.

→ A passive attack attempts to learn or make use of information from the system but does not affect system resources.

→ Active attacks attempts to alter system resources or affect their operation.

Passive attack:

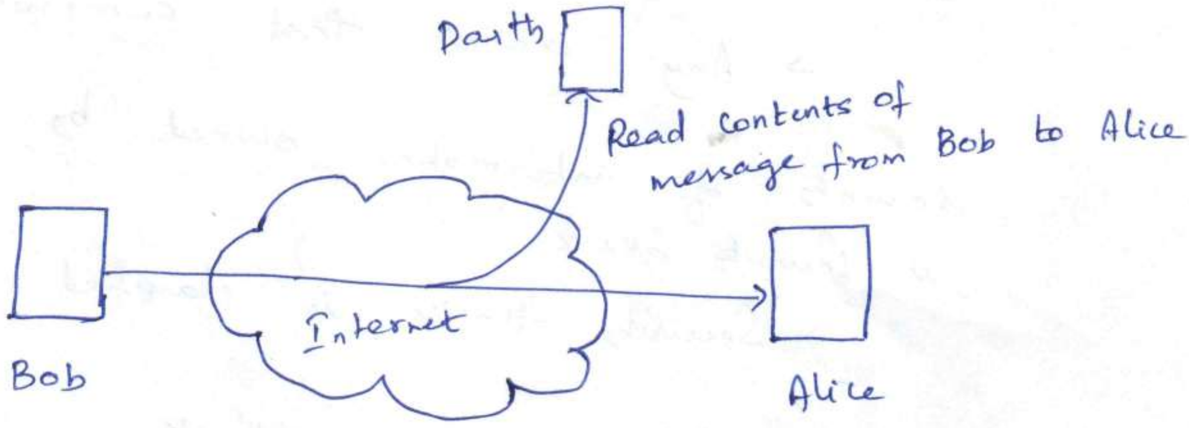
→ Passive attack are the monitoring of transmission.

→ The main goal of the opponent is to obtain information that is being transmitted

→ Two types of passive attacks are

1. Release of message content
2. Traffic analysis.

→ Release of message Contents is simply the conversation of source and destination is observed by the third party and released the Contents.

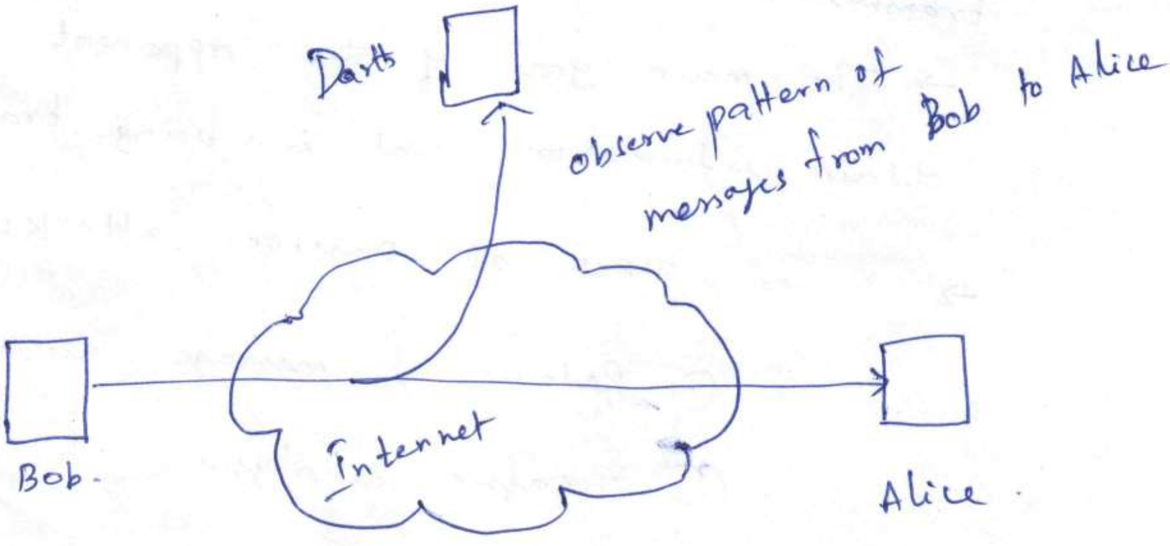


Release of message Content

→ Traffic analysis is observing the transmission between source and destination.

→ Here the data may be obtained from the transmission but it can't be understood by others.

→ The opponent could determine the pattern of the message, location and identity of communication hosts and observe the frequency and length of messages being exchanged.



Traffic analysis

Active attacks:

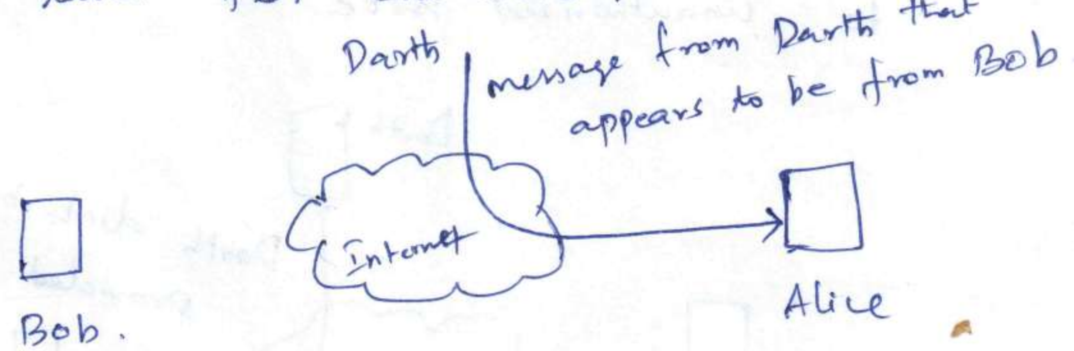
→ Active attacks involves some modification of the data stream (or) creation of a false data stream.

→ Active attacks can be subdivided into four categories

① Masquerade
② Replay
③ Modication of message
④ Denial of service.

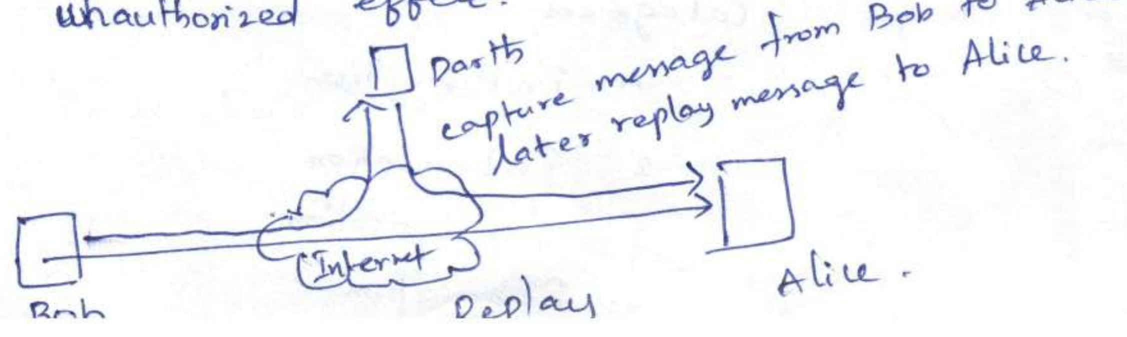
→ Masquerade takes place when one entity pretends to be a different entity

→ It captures authorization privillages and used them later for tramission.



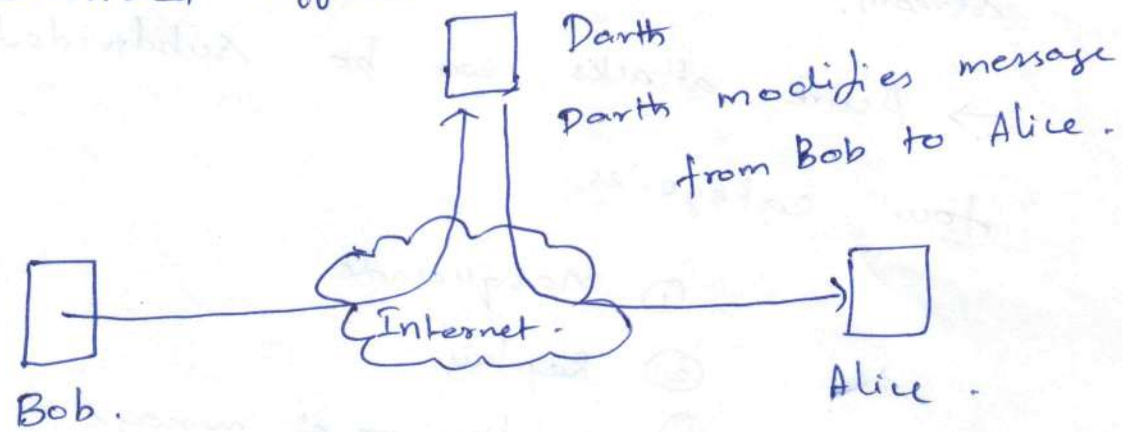
Masquerade

→ Replay: involves the capturing of data unit and its subsequent retransmission to produce an unauthorized effect.



Replay

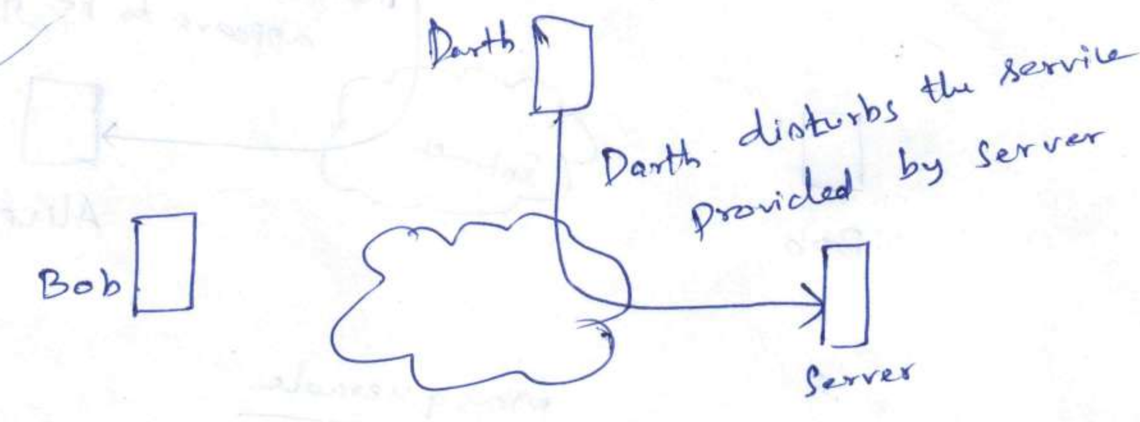
→ Modification of messages simply means that some portion of message is altered or that messages are delayed or reordered to produce an unauthorized effect.



Modification of Message

→ Denial of Service means disturbing the normal transmission.

→ The messages coming to destination are redirected to unauthorized site



Denial of Service

In general Security attacks comes under four major categories.

1. Interruption
2. Interception
3. Modification
4. Fabrication.

Interruption:

→ If the data on the transmission link is destroyed or made not available to the destination, it is called as interruption.

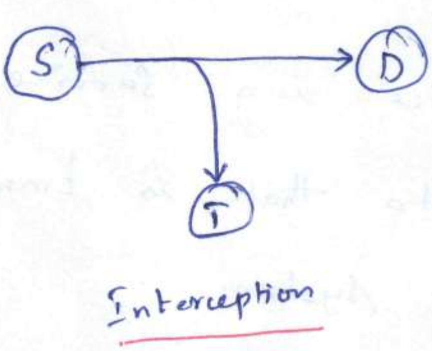
→ Once the interruption is attacked the data is unavailable at the destination.

Interruption.

Interception:

→ Interception means accessing data by an unauthorized party

→. An unauthorized party only access the data but they don't modify it, so that the source and destination can't guess that something happened in between transmission.

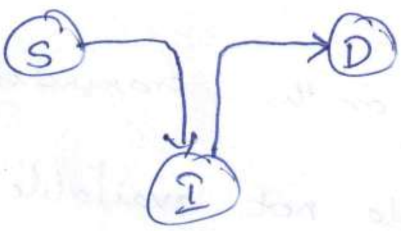


Interception

Modification:

→ when unauthorized party changes the data it is called as modification attack.

→ It includes accessing the data and changing it.

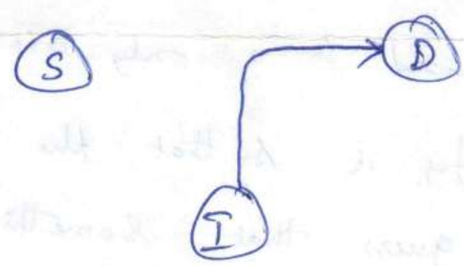


Modification

Fabrication:

→ Intruders Create a message and send to the destination party pretends as the source is sending message.

→ Eventhough there is no transmission between ~~sender~~ source and destination intruders sends message to the destination as if it was sent by source.



Fabrication.

Security Services:

→ Security service is a service that provides security for data that is transferred from source to destination system.

→ There are different security services that are needed for transferring data. They are

1. confidentiality
2. Authentication
3. Availability
4. Integrity
5. Access control
6. Non repudiation

Confidentiality:

→ when users are sending data from source to destination the data must be send in a secured mannar.

→ Confidentiality means no body should view our data except the person at destination

→ The minimum requirement for a Network data transfer is Confidentiality

Authentication:

→ Authentication means identifying origin of message correctly and it should ensure that identity is not false.

→ Authentication provides security for our data by means of identifying you are an authorized user.

→ Two major kinds of authentication are

(1) Peer entity authentication

(2) Data origin authentication

→ Peer entity authentication provides checking of identity at origin or middle of transmission. It provides security from replay and masquerade attack.

→ Data origin authentication verifies the source data is an authorized one or not. It provides. security from duplication and modification of data unit.

Integrity:

→ Integrity means data that is sent through the secured channel is not altered or tampered by others.

→ It ensures that message received is as it is sent from source.

→ Integrity is not ensuring that third party is viewing or not, only it ensures that no other party is not modifying our data.

Availability:

→ The data must be available to the authorized parties when they required to access them is called as availability

→ If any third party destroys (or) does any harm to them it is not possible to access the data.

→ Even in those cases service must be provided to recover from loss of availability.

Access control:

→ Host system and application are limited to access by communication links and no other party can access them.

→ To achieve this each authorized user trying to gain access must first be identified (or) authenticated so that access right can be given to that user.

Non repudiation:

→ Non repudiation prevents either sender or receiver from denying a transmitted message.

→ Thus when a message is sent, the receiver can prove that sender in fact sent the message.

→ Similarly when a message is received the sender can prove that receiver in fact received the message.

Security Mechanism:

→ Security Mechanisms are exist to provide security services for data transmission.

→ There are several mechanisms types They are

1. Encipherment Mechanism:

The text message can be transmitted into another form and then start the transmission (encryption) so that data flow confidentiality is provided.

2. Digital Signature Mechanism:

Signing and verifying procedure for data, so that authentication ensured with digital signature mechanism.

3. Access control Mechanism:

By using privilage concept we can control accessing of data

4. Data integrity Mechanism:

→ Protection against Modification of data is data integrity.

→ By using Hash function or Message authentication function we can provide data integrity.

A Model of Internetwork Security:-

→ A message to be transferred from one party to another across some sort of internet.

→ The two parties who are principals in this transaction must co-operate for the exchange to take place.

→ A logical information channel is estabilished by defining a route through internet from source to destination by co-operative use of communication protocols (eg: TCP/IP) by the two principals.

→ Security aspects Come into play when it is necessary to protect the information transmission from an opponent who may present to attack on confidentiality, Authenticity and so on.

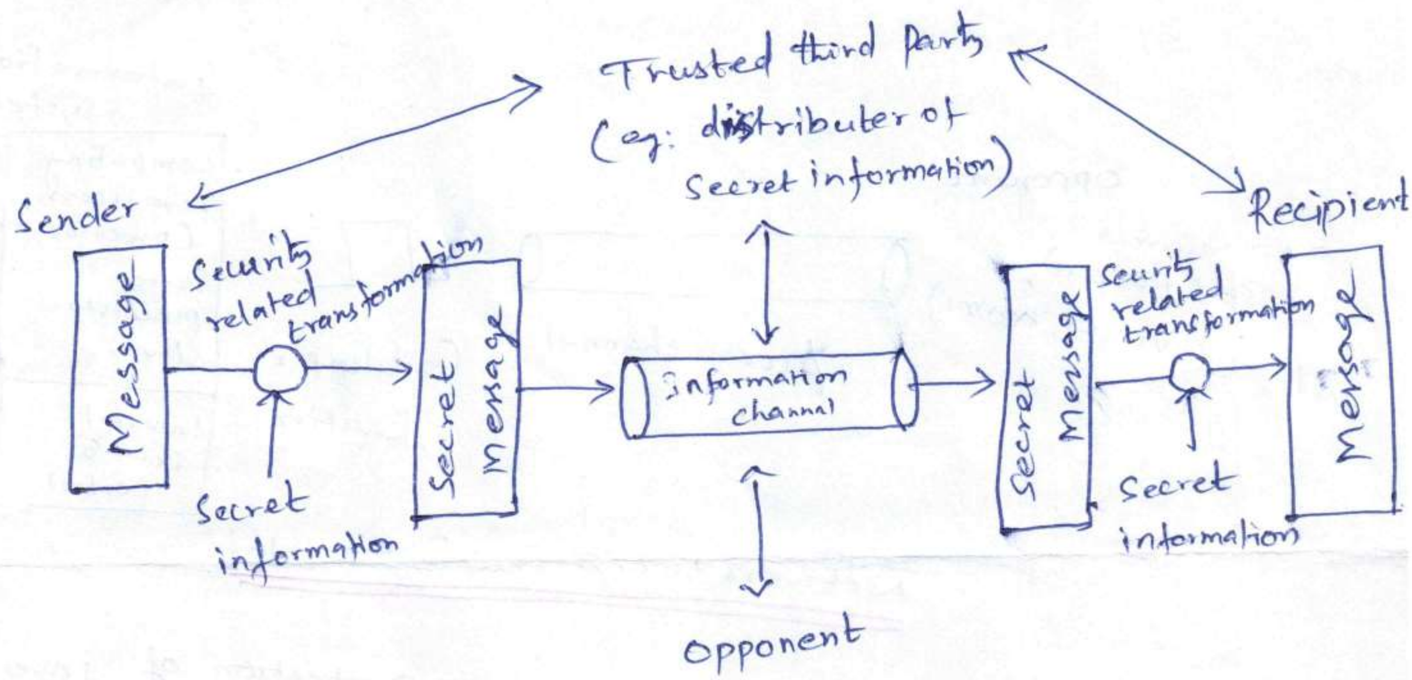
→ To provide security we should have two components

1. A security related transformation on the information to be sent.

Example: Encryption of the message which makes the information unreadable by the opponent.

2. Some secret information shared by the two principals and it should be unknown to the opponent.

Example: Encryption key used to make the message unreadable by the opponent before transmission and make it readable at the reception.



Model of Network Security

→ A trusted third party may be needed to achieve secured transmission.

Example: Trusted third party may be responsible for distributing the secret information to two principals while keeping it from any opponent.

→ The general model shows four basic tasks in desiging a particular security service:

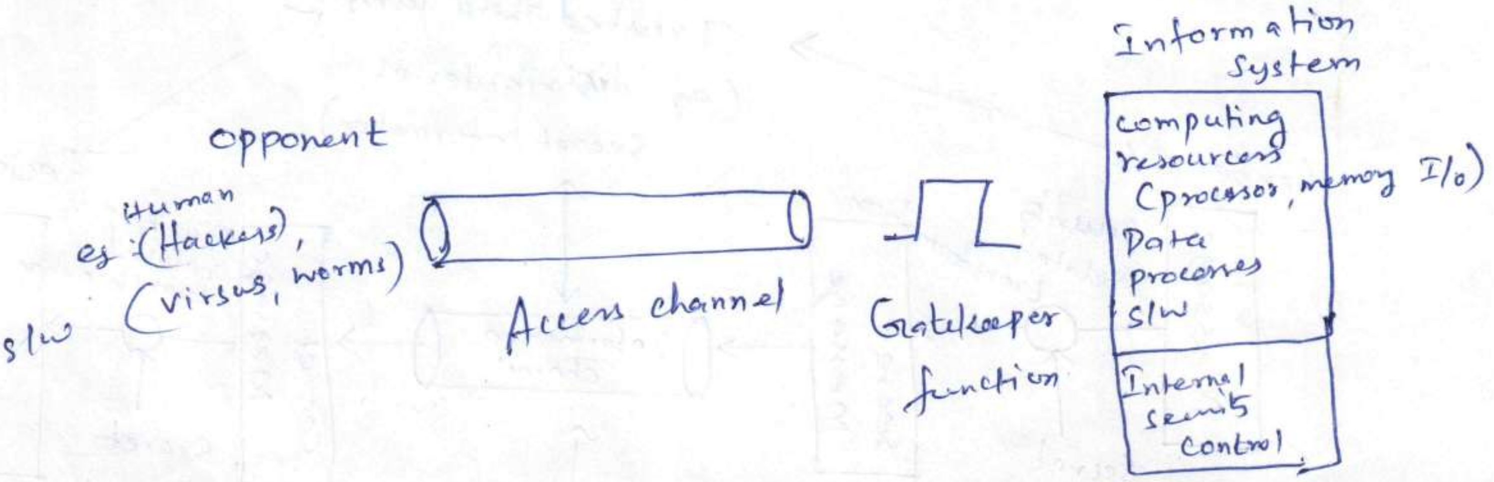
1. Design an algorithm for performing the security related transformation

2. Generate the Secret information to be used with the algorithm.

3. Develop the methods for distribution and Sharing the Secret information

4. Specify a protocol for Network access.

Network access Security Model:



Network access Security Model.

→ The above diagram gives protection of information System from unwanted access.

→ There are two types of threats

① Information access threats.

It modify data on behalf of users who should not have access to that data.

② Service threat: ~~exp~~

It exploit service flow in computers to inhibit use of authorized uses.

→ The Security Mechanism needed to cope with unwanted access fall into two broad categories

1. Gate keeper function

It includes password based login procedure that are designed to deny access to all except for authorized uses.

2. Screening logic that are designed to detect and reject worms, viruses and other similar attacks.

Internet Standards and RFC's:

→ The Internet Society is the Coordinating Committee for Internet design, engineering and Management.

→ It includes the operation of internet and Standardization of protocols used by end system on the internet for interoperability.

→ Three Organization under Internet Societs are responsible for the actual work of standards development and publication.

1. Internet architecture Board (IAB):

Responsible for defining overall architecture of the Internet, providing guidance and broad direction to the IETF

2. Internet Engineering Task force (IETF)

The protocol engineering and development arm of the internet.

3. Internet Engineering steering group (IESG)

Responsible for technical management of IETF activities and Internet Standards process.

UNIT - II

Conventional Encryption Principals:

Symmetric cipher model.

A convention Encryption scheme has five ingredients.

1. plain text:

This is the original message or data that is given as i/p to the algorithm

2. Encryption algorithm:

The encryption algorithm performs various substitutions and transformations on the plaintext.

3. Secret key:

The secret key is also i/p to the algorithm. The exact substitution and transformations are performed by the algorithm depends on the key.

4. Ciphertext:

This is the scrambled message produced as output. It depends on the plaintext and secret key.

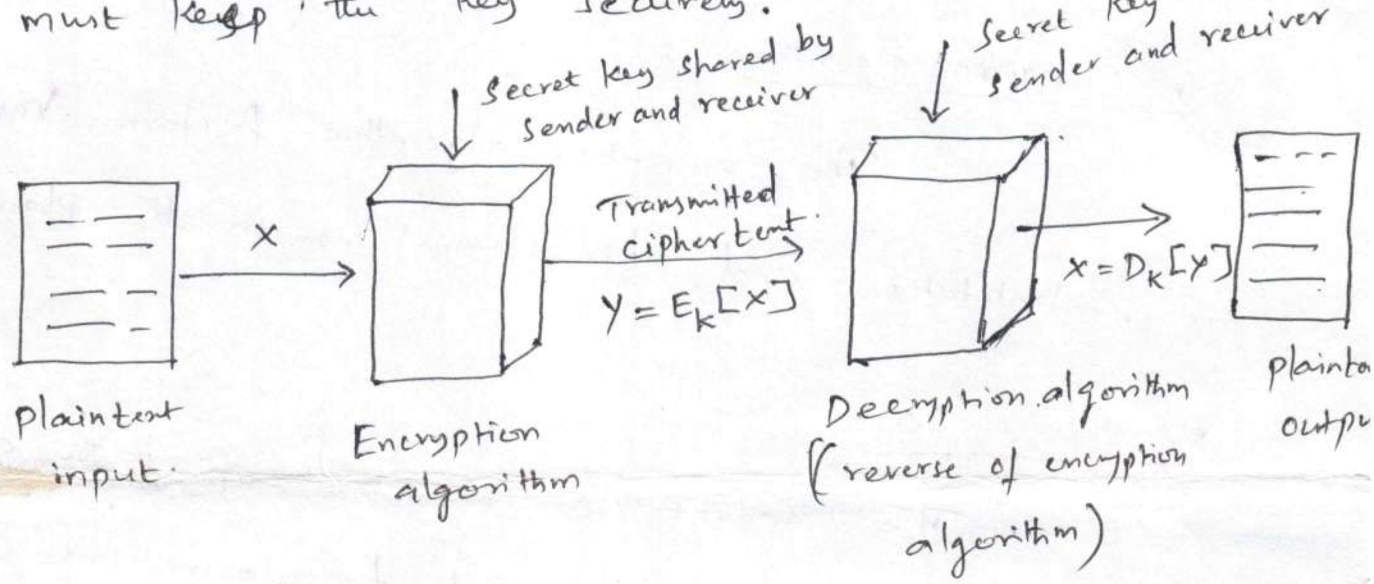
5. Decryption algorithm:

This is essentially the encryption algorithm run in reverse. It takes ciphertext and same secret key and produces the original plaintext.

→ There are two requirements for secure use of conventional encryption

1. The opponent should be unable to decrypt the ciphertext or discover the key even if he is in a position to get the ciphertext.

2. Sender and receiver must have obtained the copies of secret key in a secured fashion and must keep the key securely.



Simplified Model of Conventional Encryption

→ It is important to note that security of symmetric encryption depends on the secrecy of key and not the secrecy of algorithm.

→ That is, it is assumed that it is impractical to decrypt a message on the basis of ciphertext plus knowledge of the encryption/decryption algorithm.

Cryptography:

Algorithms that are used to transforming Plain text to ciphertext and cipher text to plaintext are called as cryptographic techniques and Study of these techniques are known as Cryptography

→ Cryptographic system are generally classified along 3 independent dimension.

1. The type of operation used for transforming plaintext to ciphertext:

→ All encryption algorithm are based on two general principals

1. substitution:
In which each element in the plaintext is mapped into another element.

2. Transposition:
In Which elements in the plaintext are rearranged.

2. The number of Keys used:

→ If both sender and receiver use the Same Key, the system is referred to as Conventional Symmetric, Single Key, Secret key or encryption

→ If the sender and receiver each use a different key, the system is refered to as asymmetric, two keys or public key encryption.

3. The way in which plaintext is processed:

→ A block cipher processes the i/p one block of elements at a time, producing an output block for each i/p block.

→ A stream cipher processes the i/p elements continuously, producing the o/p one element at a time as it goes along.

Cryptanalysis:

→ Identifying the plaintext or key from the ciphertext is known as Cryptanalysis.

→ The person who deals with Cryptanalysis is known as Cryptanalyst.

→ Cryptanalyst used different strategies to identify plaintext depend on the availability of information.

→ There are several types of attack on the encrypted message. They are

1. Cipher text only
2. known plain text.
3. chosen plain text
4. chosen cipher text.
5. chosen text.

1. cipher text only:

→ It is the most difficult type of attack. where cryptanalyst applies brute force approach of ~~trying~~ all possible keys by assuming some algorithm.

4. chosen cipher text:

→ It is an opposite attack to chosen plain text attack.

→ Cryptanalyst obtains temporary access to decryption machinery, ~~and~~ he can choose cipher text string and can construct the corresponding plaintext string.

2 approaches for attacking a conventional encryption scheme

1. cryptanalysis:

→ It rely on the nature of algorithms and knowledge of general characteristics of plaintext.

→ It exploits the characteristics of algorithm to attempt to deduce a specific PT (or) deduce a key being used.

2. Brute force attack:

→ Attacker tries every possible key on a piece of ciphertext until an proper plaintext is obtained.

→ On an average half of all possible keys must be tried to achieve success.

Classical Encryption Techniques

Substitution and Transposition ciphers

Substitution Ciphers

- Each Letters of plaintext are replaced by other letters or by numbers or symbols
- If plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

Caesar Cipher

- It is earlier known as substitution cipher
- It replaces each letter by 3rd letter on the sequence of alphabet.
- example:

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

Caesar Cipher

- We can define transformation as:

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

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

- We mathematically give each letter a number

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

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

- Caesar cipher is defined as:

c = E(p) = (p + k) mod (26)

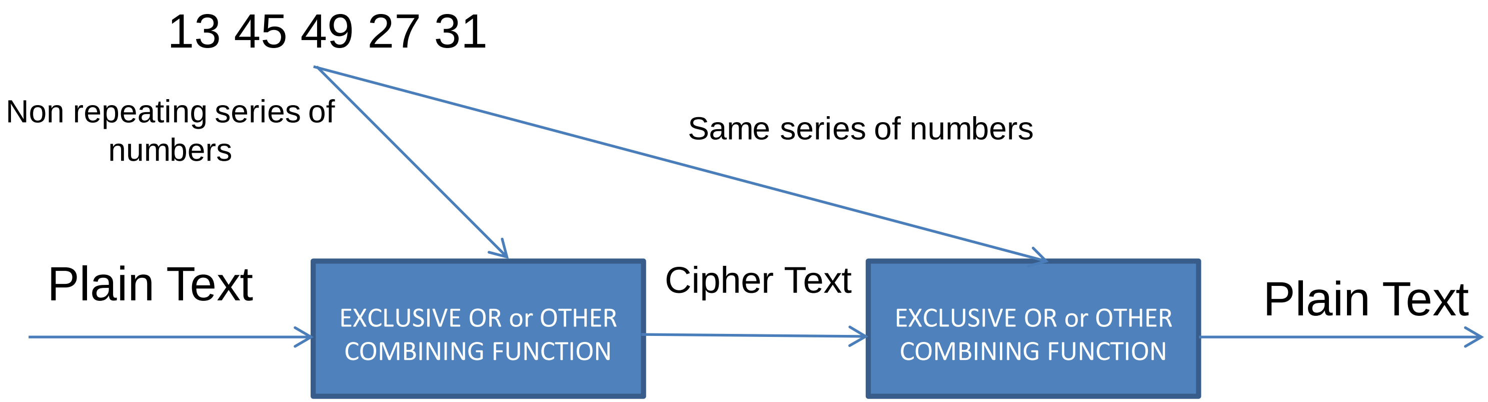
p = D(c) = (c – k) mod (26)

Cryptanalysis of Caesar Cipher

- It have only 26 possible ciphers
  - A maps to A,B,..Z
- It could simply try each in turn by using **brute force search**
- Given ciphertext, just try all shifts of letters to get meaning plaintext.
- eg. break ciphertext “CKRIUSK"

Vernam cipher

- Arbitrarily long non repeating sequence of numbers combined with plain text
- Long punched paper tape containing random numbers combined the Plaintext by XOR
- Key tape does not repeat and it is not reused which immunes to cryptanalytic attacks



Vernam Cipher

Vernam cipher Example

- Letters in alphabet are represented with numbers 0 through 25
- Sum this numerical representation with stream of random numbers
- If the message is VERNAM CIPHER the letters are converted to numerical form as

V	E	R	N	A	M	C	I	P	H	E	R
21	4	17	13	0	12	2	8	15	7	4	17

Vernam cipher Example

- Series of 2 digit Random numbers are
  76 48 16 82 44 03 58 11 60 05 48 88
- Encoded form is sum mod 26 of each coded letter
- Result is then encoded to usual 26 alphabet representation

Plain Text: V E R N A M C I P H E R

Numeric Eq: 21 4 17 13 0 12 2 8 15 7 4 17

+Random No:76 48 16 82 44 03 58 11 60 05 48 88

= sum :97 52 33 95 44 15 60 19 75 12 52 105

=mod 26 :19 0 7 17 18 15 8 19 23 12 0 1

Cipher text : T A H R S P I T X M A B

Monoalphabetic Cipher

- It is rather than just shifting the alphabet, could shuffle (jumble) the letters arbitrarily.
- Each plaintext letter maps to a different random ciphertext letter

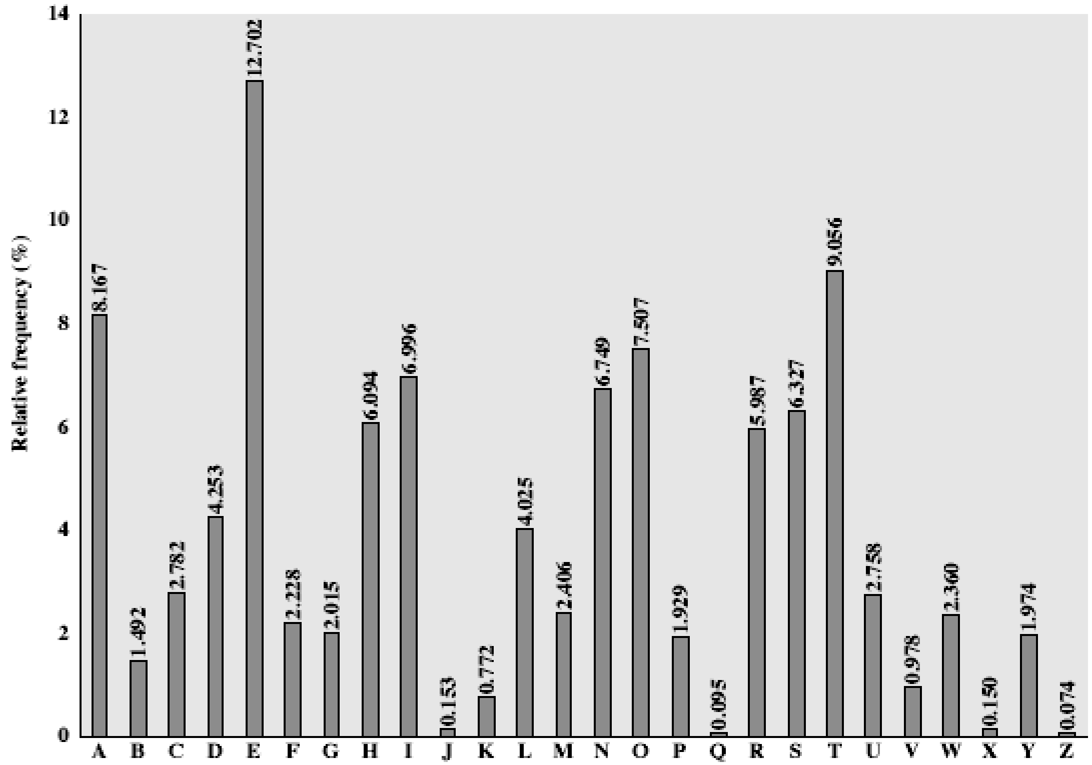
```
Plain   :  abcdefghijklmnopqrstuvwxyz
Cipher  :  DKVQFIBJWPESCXHTMYAUOLRGZN

Plaintext    :  ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA
```

Language Redundancy and Cryptanalysis

- All human languages are **redundant** , So letters are not equally commonly used.
- In English E is by far the most common letter followed by T,R,N,I,O,A,S
- Other letters like Z,J,K,Q,X are fairly rare.
- We have tables of single, double & triple letter frequencies for various languages

English Letter Frequencies



Playfair Cipher

- It is not even the large number of keys in a mono alphabetic cipher provides security.
- A good approach to improving security was to encrypt multiple letters.
- The **Playfair Cipher** is an example invented by Charles Wheatstone in 1854, but named after his friend Baron Playfair.

Playfair Key Matrix

- A 5X5 matrix of letters based on a keyword
- Fill in letters of keyword first and fill rest of matrix with other letters
- eg. Using the keyword MONARCHY

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

Encrypting and Decrypting

- Plaintext is encrypted two letters at a time
  1. If a pair is a repeated letter, insert filler like 'X' EX: OO will be encrypted as OXO
  2. If both letters fall in the same row, replace each with letter to right (wrapping back to start from end) Ex: AR is encrpted as RM
  3. If both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom) Ex: MU is encrypted as CM
  4. Otherwise each letter is replaced by the letter in the same row and in the column of the other letter of the pair Ex: HS becomes as BP

Vigenère Cipher

- A simplest poly alphabetic substitution cipher
- It effectively uses multiple caesar ciphers
- Key is multiple letters long $K = k_1\ k_2\ \ldots\ k_d$ in which i^{th} letter specifies i^{th} alphabet to use
- It uses each alphabet in turn.
- Decryption simply works in reverse.

Vigenere table

	A	B	C	D	E	F	G	.	..		W	X	Y	Z
A	A	B	C	D	E	F	G	.	..		W	X	Y	Z
B	B	C	D	E	F	G	.	..		W	X	Y	Z	A
C	C	D	E	F	G	.	..		W	X	Y	Z	A	B
D	D	E	F	G	.	..		W	X	Y	Z	A	B	C
E	E	F	G	H	I	..	W	X	Y	Z	A	B	C	D
F	F	G	.	..		W	X	Y	Z	A	B	C	D	E
G	G	.	..		W	X	Y	Z	A	B	C	D	E	F
H	H	…												
I	I	…												
J	J	…												
K														
.														
Z														

Example of Vigenère Cipher

- Write the plaintext out
- Write the keyword repeated above it
- We use each key letter as a cipher key and encrypt the corresponding plaintext letter
- eg using keyword *deceptive*

```
key:        deceptivedeceptivedeceptive
plaintext:  wearediscoveredsaveyourself
ciphertext:ZICVTWQNGRZGVTWAVZHCQYGLMGJ
```

Autokey Cipher

- Vigenère proposed the **autokey** cipher with keyword is concatenated with the plaintext itself to provide a running Key.
- eg. given key *deceptive*

```
key:        deceptivewearediscoveredsav
plaintext:  wearediscoveredsaveyourself
ciphertext:ZICVTWQNGKZEIIGASXSTSLVVWLA
```

Hill Cipher

- Takes two or three or more letter combinations to the same size combinations, e.g. “the” → “rqv”
- Uses simple linear equations
- An example of a “block” cipher encrypting a block of text at a time
- Numbered alphabet: a = 0, b = 1, c = 3, etc. (in CAP, use ASCII code)

Hill Cipher

Encryption:

$$C = K * P \mod 26$$

where K is a key matrix that should not be a singular matrix.

Hill Cipher

Decryption:

$$P = K^{-1} * P \mod 26$$

where K^{-1} is the inverse of key matrix.

Hill Cipher

- Developed by the mathematician Lester Hill in 1929.
- The encryption algorithm takes m successive plain text and substitute for them m cipher text letters.
- Each character is assigned a numerical value (a=0,…z=25).

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

$$C = KP \bmod 26$$

$$P = K^{-1}C \bmod 26 = KK^{-1}P = P$$

Hill Cipher

Example

```
C1 =  9*p1 + 18*p2 + 10*p3 (mod 26)
C2 = 16*p1 + 21*p2 +  1*p3 (mod 26)
C3 =  5*p1 + 12*p2 + 23*p3 (mod 26)
```

$$\begin{pmatrix} C1 \\ C2 \\ C3 \end{pmatrix} = \begin{pmatrix} 9 & 18 & 10 \\ 16 & 21 & 1 \\ 5 & 12 & 23 \end{pmatrix} \begin{pmatrix} p1 \\ p2 \\ p3 \end{pmatrix} \pmod{26}$$

Hill Cipher

I can't do it

8 2 0 13 19 3 14 8 19

→ **EOM TMY SVJ**

$$\begin{pmatrix} 4 \\ 14 \\ 12 \end{pmatrix} = \begin{pmatrix} 9 & 18 & 10 \\ 16 & 21 & 1 \\ 5 & 12 & 23 \end{pmatrix} \begin{pmatrix} 8 \\ 2 \\ 0 \end{pmatrix} \pmod{26}$$

$$\begin{pmatrix} 19 \\ 12 \\ 14 \end{pmatrix} = \begin{pmatrix} 9 & 18 & 10 \\ 16 & 21 & 1 \\ 5 & 12 & 23 \end{pmatrix} \begin{pmatrix} 13 \\ 19 \\ 3 \end{pmatrix} \pmod{26}$$

$$\begin{pmatrix} 18 \\ 21 \\ 9 \end{pmatrix} = \begin{pmatrix} 9 & 18 & 10 \\ 16 & 21 & 1 \\ 5 & 12 & 23 \end{pmatrix} \begin{pmatrix} 14 \\ 8 \\ 19 \end{pmatrix} \pmod{26}$$

Transposition Ciphers

- **A very different kind of mapping is achieved by some sort of permutation** on the plain text letters
- These hide the message by rearranging the letter order without altering the actual letters used.
- Example: Rail fence cipher

Rail Fence cipher

- We write message letters out diagonally over a number of rows and then read off cipher row by row
- eg. write message out as:

```
m e m a t r h t g p r y
 e t e f e t e o a a t
```

- It produces ciphertext as follows

```
MEMATRHTGPRYETEFETEOAAT
```

Row Transposition Ciphers

- It is a more complex transposition
- We write letters of message out in rows over a specified number of columns
- We then reorder the columns according to some key before reading off the rows

```
Key:          3    4    2    1    5    6    7
Plaintext:    a    t    t    a    c    k    p
              o    s    t    p    o    n    e
              d    u    n    t    i    l    t
              w    o    a    m    x    y    z
Ciphertext: APTM TTNA AODW TSUO COIX KNLY PETZ
```

Cryptography and Network Security
Unit 2

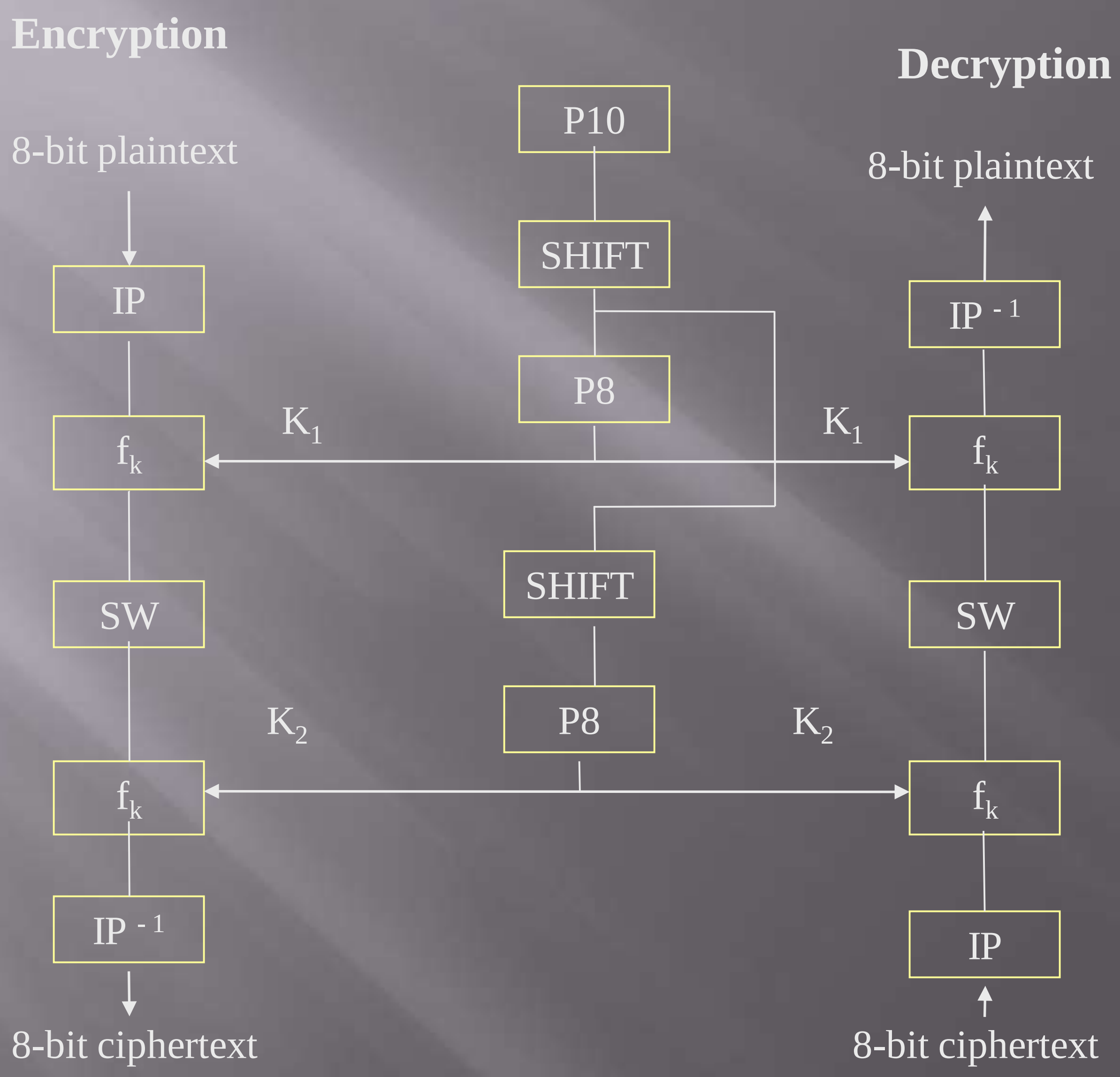
Block Ciphers- Data Encryption Standards and Public Key Cryptography

Simplified DES - Block Cipher Principles - DES – AES - Block Cipher Design Principles - Block Cipher modes of Operation - Public Key Cryptography - Principles of Public Key Cryptosystems - The RSA Algorithm -Diffie Hellman Key Exchange

What is Simplified DES

- S-DES was developed in 1996 as a teaching tool at Santa Clara University by Prof. Edward Schaefer.
- Takes an 8-bit block plaintext, a 10 –bit key and produces an 8-bit block of ciphertext
- Decryption takes the 8-bit block of ciphertext, the same 10-bit key and produces the original 8-bit block of plaintext

S-DES Scheme



Encryption
Decryption
P10
8-bit plaintext
8-bit plaintext
SHIFT
IP
IP - 1
P8
K1
K1
fk
fk
SHIFT
SW
SW
K2
P8
K2
fk
fk
IP - 1
IP
8-bit ciphertext
8-bit ciphertext

Five Functions to Encrypt

- IP – an initial permutation
- f_k - a complex, 2-input function
- SW – a simple permutation that swaps the two nybles
- f_k - a complex, 2-input function; again
- IP – inverse permutation of the initial permutation

S-DES Scheme

- Encryption process is defined as
  ciphertext = IP^{-1} (f_{k2} (SW (f_{k1} (IP (plaintext)))))
- Decryption process is defined as
  Plaintext = IP(f_{k1} (SW (f_{k2} (IP^{-1} (ciphertext)))))

Key generation process

- <u>To obtain K_1 and K_2 :</u>
- Given: $K = (k_1\ k_2\ k_3\ k_4\ k_5\ k_6\ k_7\ k_8\ k_9\ k_{10})$
- Step1: Permutation P10

P10 :

3	5	2	7	4	10	1	9	8	6

- Step2: Left shift (circular) by one bit
  - for the left half and
  - for the right half <u>separately.</u>

Key generation process

- Step3: Permutation for producing an 8 bit key K_1 from a 10 bit input.

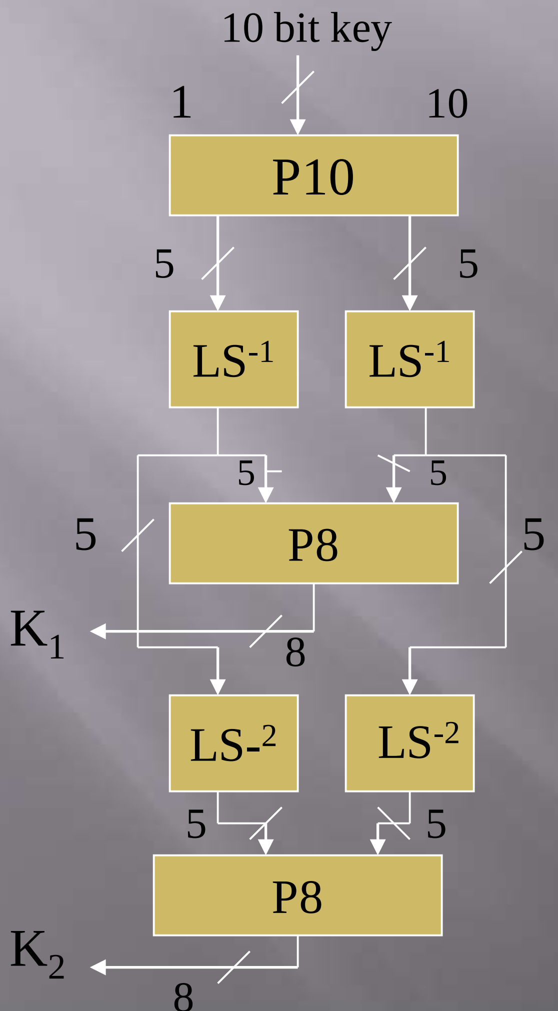
P8:

6	3	7	4	8	5	10	9

- Step4: Take the result of step2. On it use Left Shift (circular) by 2 bits
  - for the left half and
  - for the right half separately.
- Step5: Another instance of P8 is used to produce the second 8 bit key K_2 .

$$K_1 = P8\ (\ Shift\ (P10\ (Key)))$$
$$K_2 = P8\ (\ Shift\ (\ Shift\ (P10\ (Key)))).$$

Key generation process



3	5	2	7	4	10	1	9	8	6	P10

Circular left shift by 1, separately on the left and the right halves

6	3	7	4	8	5	10	9	P8

Circular left shift by 2 , separately on the left and the right halves

6	3	7	4	8	5	10	9	P8

Example: Key Generation

3	5	2	7	4	10	1	9	8	6

P10

6	3	7	4	8	5	10	9

P8

10-bit key = 1 0 1 0 0 0 0 0 1 0

1 0 0 0 0   0 1 1 0 0   P10

0 0 0 0 1   1 1 0 0 0 LS^{-1} LS^{-1}

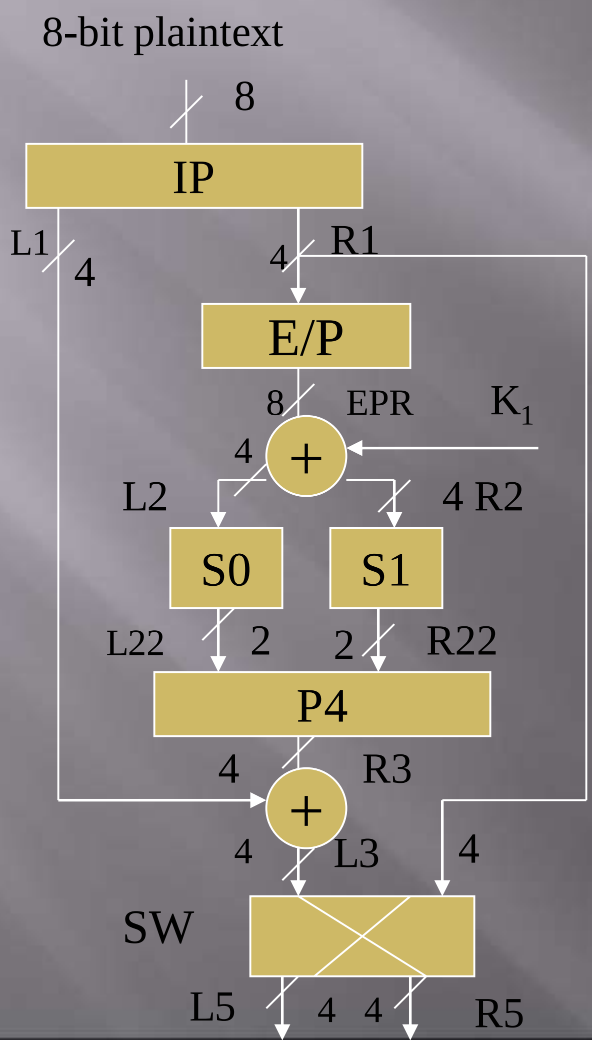
1 0 1 0   0 1 0 0 $= K_1$ P8

0 0 1 0 0   0 0 0 1 1 LS^{-2} LS^{-2}

0 1 0 0   0 0 1 1 $= K_2$ P8

Simplified DES Encryption

$$\text{ciphertext} = IP^{-1}\left(f_{k2}\left(SW\left(f_{k1}\left(IP\left(\text{plaintext}\right)\right)\right)\right)\right)$$



2 6 3 1 4 8 5 7	IP
4 1 3 5 7 2 8 6	IP^{-1}
4 1 2 3 2 3 4 1	E/P

S0 and S1 boxes:

$$S0 = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 0 & 3 & 2 \\ 1 & 3 & 2 & 1 & 0 \\ 2 & 0 & 2 & 1 & 3 \\ 3 & 3 & 1 & 3 & 2 \end{array}$$

$$S1 = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 1 & 2 & 3 \\ 1 & 2 & 0 & 1 & 3 \\ 2 & 3 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 & 3 \end{array}$$

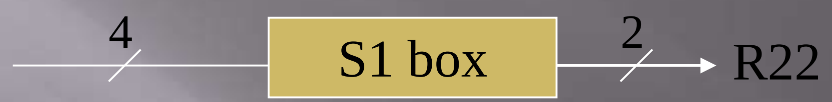
2	4	3	1

P4

The function f_k :

- Permutation IP is applied to the 8-bit plaintext to generate L1 and R1, the left and the right halves.
- INPUT to $f_{k:}$
  - 4-bit left half (L1) and
  - 4-bit Right half (R1) of a data string.
- Step1: E/P: Expansion/Permutation on R1 to produce an 8 bit data string called EPR.
- Step2: XOR of EPR with key K_1 for f_{k1} to produce the left half (L2) and right half (R2).
- Step3 (a): L2 $\xrightarrow{4}$ S0 box $\xrightarrow{2}$ L22

The function f_k (continued):



- Step3(b): R2
- Given the 4 bits of L2 (or R2) part. Pick up the ijth element of S0 (or of S1), where i = 1^{st} and 4^{th} bits; j = 2^{nd} and 3^{rd} bits.
- Then convert this element to a 2-bit binary number.

The function f_k (continued)

Step4: (L22 : R22) goes through a permutation P4 to produce a 4-bit R3.

P4

2	4	3	1

- Step5: L3 = L1 ⊕ R3
- Step6: L3 : R1 is then the input to SW .
- The second instance of f_k is similar to the first, except that the key K_2 is used.

Example: SDES Encryption

- Example:

Plaintext = 1 0 1 1 1 1 0 1

IP=0 1 1 1 1 1 1 0

L1 = 0 1 1 1

R1 = 1 1 1 0

EPR = 0 1 1 1 1 1 0 1

EPR $\oplus$ K1 = 1 1 0 1 1 0 0 1

Row : first and fourth bit

Column : 2^{nd} and 3^{rd} bit

Example: SDES Encryption (continued)

For S0:

L2 = 1 1 0 1

Therefore Row = 3 Column = 2

L22 = 3 $\Rightarrow$ 11

For S1:

R2 = 1 0 0 1

Row = 3 Column = 0

R22 = 2 $\Rightarrow$ 1 0

L22 : R22 1 1 1 0

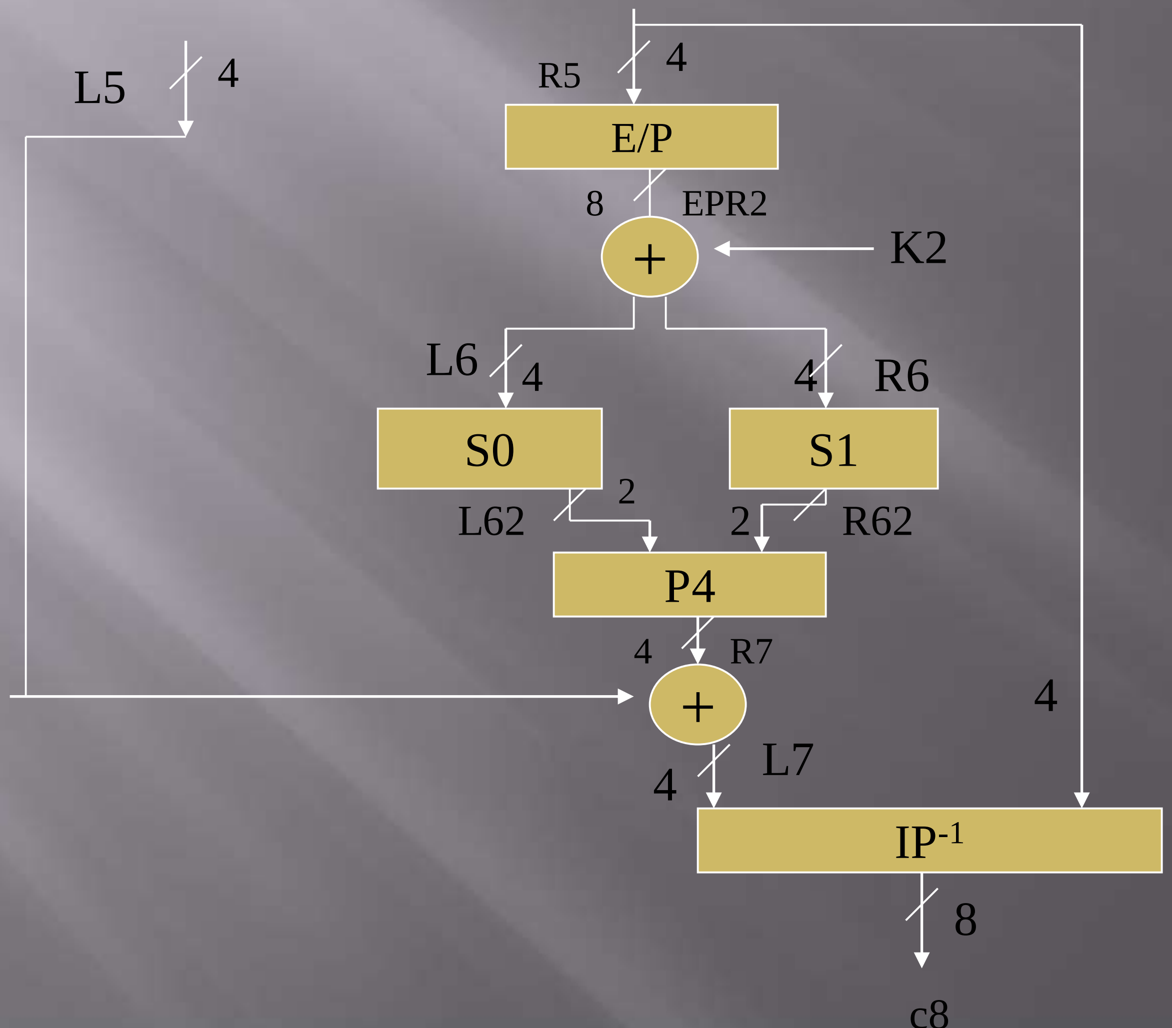
R3 = 1 0 1 1

Example: SDES Encryption (continued)

$$L3 = R3 \oplus L1$$
$$= 1\ 1\ 0\ 0$$
$$L5 = 1\ 1\ 1\ 0$$
$$R5 = 1\ 1\ 0\ 0$$

Simplified DES Encryption

ciphertext = $IP^{-1}(f_{k2}(SW(f_{k1}(IP(plaintext)))))$



Example: SDES Encryption: f_{k2}

- EPR2 = 0 1 1 0 1 0 0 1

  EPR2 $\oplus$ K2 = 0 0 1 0 1 0 1 0

  L6 = 0 0 1 0

  R6 = 1 0 1 0

For S0:

Row = 0 Column = 1

L62 = 0 $\Rightarrow$ 0 0

Example: SDES Encryption: f_{k2} (continued)

For S1:

Row = 2 Column = 1

R62 = 0 $\Rightarrow$ 0 0

R7 = 0 0 0 0

L7 = R7 $\oplus$ L5

= 1 1 1 0

L7 : L5 = 1 1 1 0 1 1 0 0

C8 = 0 1 1 1 0 1 0 1

- S-DES Decryption is the reverse process of encryption where 8 bit Ciphertext is converted back to 8 bit Plaintext.

BLOCK CIPHER PRINCIPLES

BLOCK CIPHER PRINCIPLES

- All symmetric block encryption algorithms in current use are based on a structure referred to as Fiestel block cipher.
- **A stream cipher** is one that encrypts a digital data stream one bit or one byte at a time.

  E.g: vigenere cipher, S-DES.
- **A block cipher** is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length. Typically a block size of 64 or 128 bits is used.
  - E.g: Data Encryption Standard

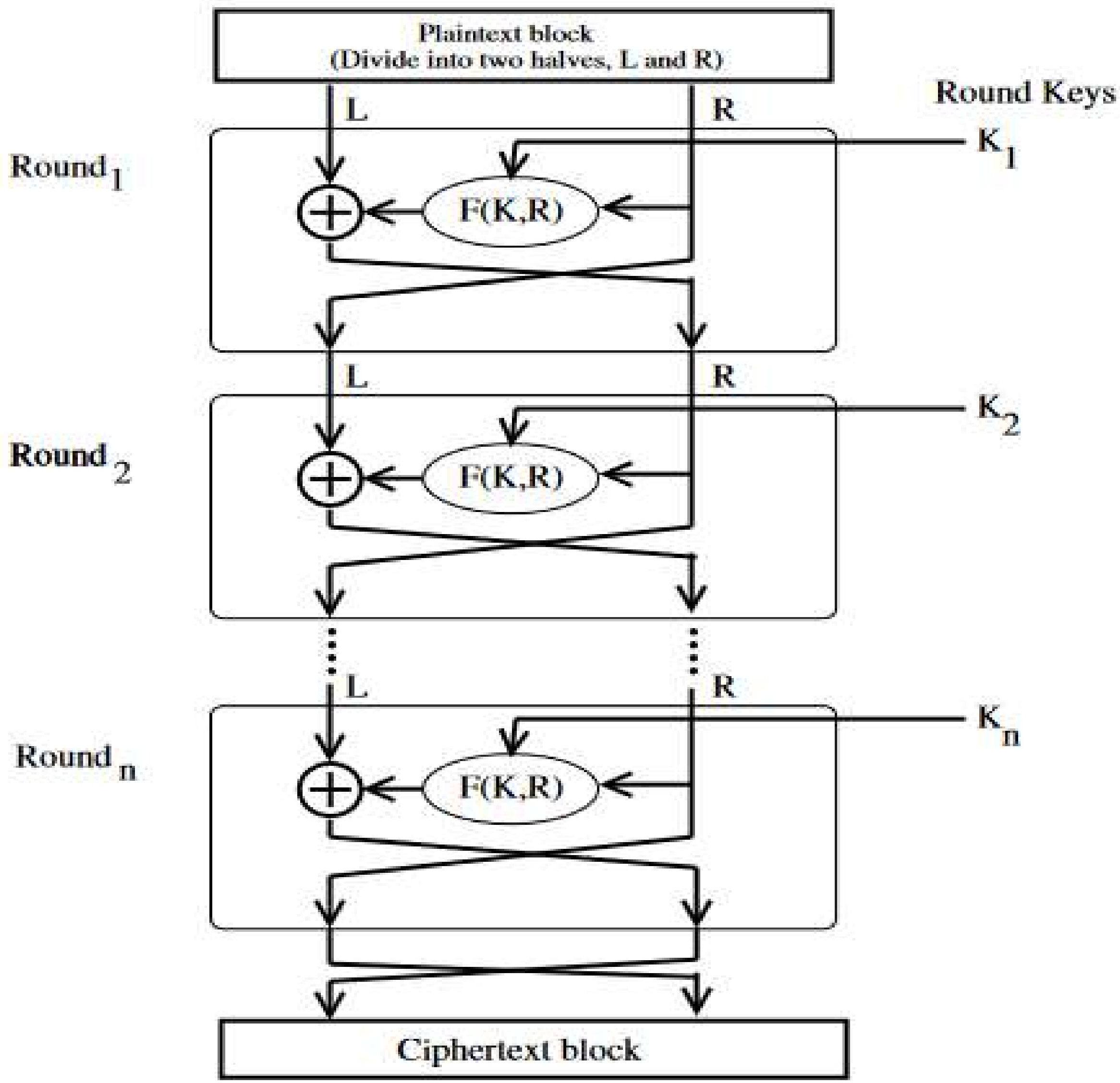
Feistel cipher structure

- The input to the encryption algorithm are a plaintext block of length 2w bits and a key K.
- The plaintext block is divided into two halves L_0 and R_0.
- The two halves of the data pass through n rounds of processing and then combine to produce the ciphertext block.
- Each round i has inputs L_{i-1} and R_{i-1}, derived from the previous round, as well as the subkey K_i, derived from the overall key K.
- In general, the subkeys K_i are different from K and from each other.

Feistel cipher structure

- All rounds have the same structure.
- A substitution is performed on the left half of the data. This is done by applying a round function F to the right half of the data and then taking the XOR of the output of that function with the left half of the data.
- The round function has the same general structure for each round but is parameterized by the round subkey k_i.
- Following this substitution, a permutation is performed that consists of the interchange of the two halves of the data.
- This structure is a particular form of the substitution-permutation network.

Feistel cipher structure



Feistel cipher structure

- The exact realization of a Feistel network depends on the choice of the following parameters and design features:
- **Block size**
  - Larger the block size improves security, but it reduces the encryption and decryption speed
  - A block size of 128bit is a reasonable tradeoff.
- **Key size**
  - Increasing size improves security, makes exhaustive key searching harder, but may slow cipher
  - The most common key length in modern algorithms is 128 bits.

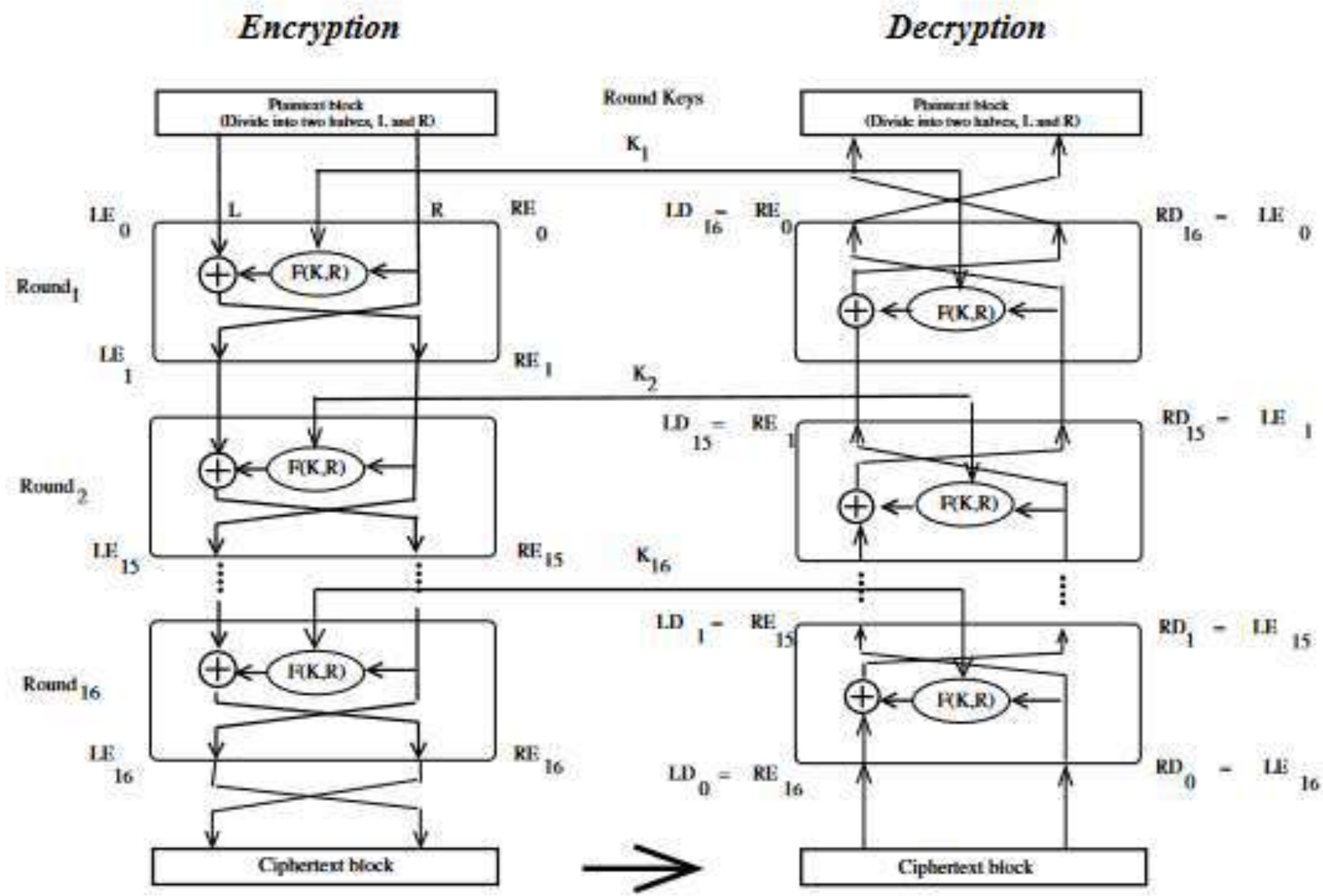
Feistel cipher structure

- **Number of rounds**
  - Increasing number of rounds improves security, but slows cipher.
  - A typical size is 16 rounds
- **Subkey generation**
  - Greater complexity can make cryptanalysis harder, but slows cipher.
- **Round function**
  - Greater complexity can make analysis harder, but slows cipher

Feistel cipher structure

- The process of decryption is essentially the same as the encryption process.
- The rule is as follows:
  - Use the cipher text as input to the algorithm, but use the subkey k_i in reverse order. i.e., k_n in the first round, k_{n-1} in second round and so on.
  - For clarity, we use the notation LE_i and RE_i for data traveling through the decryption algorithm.
  - The diagram below indicates that, at each round, the intermediate value of the decryption process is same (equal) to the corresponding value of the encryption process with two halves of the value swapped.

Feistel cipher structure



Feistel cipher structure

- First consider the encryption process,

$$LE_{16} = RE_{15}$$

$$RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$$ On the decryption side,

$$LD_1 = RD_0 = LE_{16} = RE_{15}$$

$$\begin{aligned} RD_1 &= LD_0 \oplus F(RD_0, K_{16}) \\ &= RE_{16} \oplus F(RE_{15}, K_{16}) \\ &= [LE_{15} \oplus F(RE_{15}, K_{16})] \oplus F(RE_{15}, K_{16}) \\ &= LE_{15} \end{aligned}$$

- Therefore,

$$LD_1 = RE_{15}$$
$$RD_1 = LE_{15}$$

Feistel cipher structure

- In general, for the i^{th} iteration of the encryption algorithm,

$$LE_i = RE_{i-1}$$

$$RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i)$$

- Finally, the output of the last round of the decryption process is $RE_0 \,||\, LE_0$. A 32-bit swap recovers the original plaintext.

Data Encryption Standard

Data Encryption Standard

- Data Encryption Standard (DES) was issued in 1977 as Federal Information Processing Standard 46 (FIPS 46) by the National Institute of Standards and Technology (NIST).
- Data are encrypted in 64-bit blocks using a 56-bit key.
- The algorithm transforms 64-bit input in a series of steps to produce 64-bit output.

Data Encryption Standard

- There are two inputs to the encryption function: the plaintext to be encrypted and the key.
- The function expects a 64-bit key out of which only 56 are used; other 8 bits can be set arbitrarily.

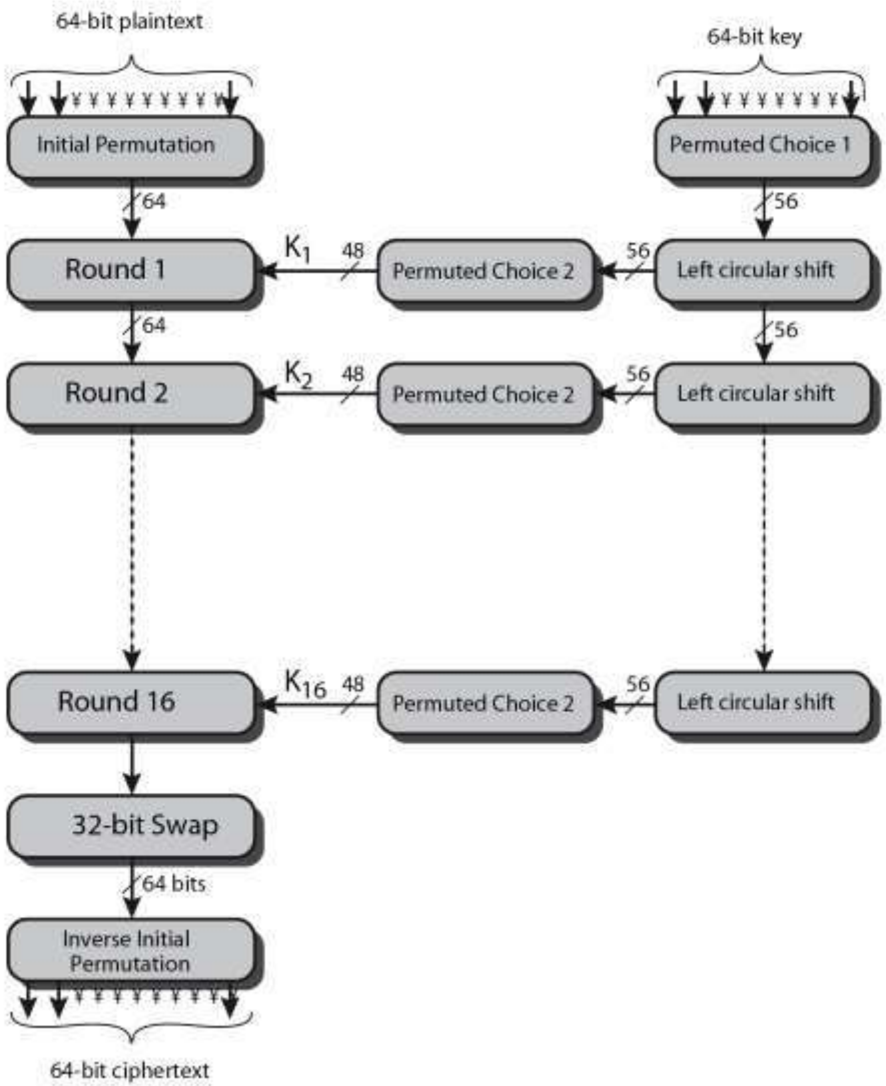
Data Encryption Standard

- Plaintext proceeds in three phases.
- **First**, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input.
- The **2nd phase** consists of 16 rounds of the same function, which involves both permutation and substitution functions.
- The output of the last round consists of 64 bits that are a function output of the input plaintext and the key.

Data Encryption Standard

- The left and right halves of the output are swapped to produce preoutput.
- **Finally**, the preoutput is passed through a permutation that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

DES Encryption Overview

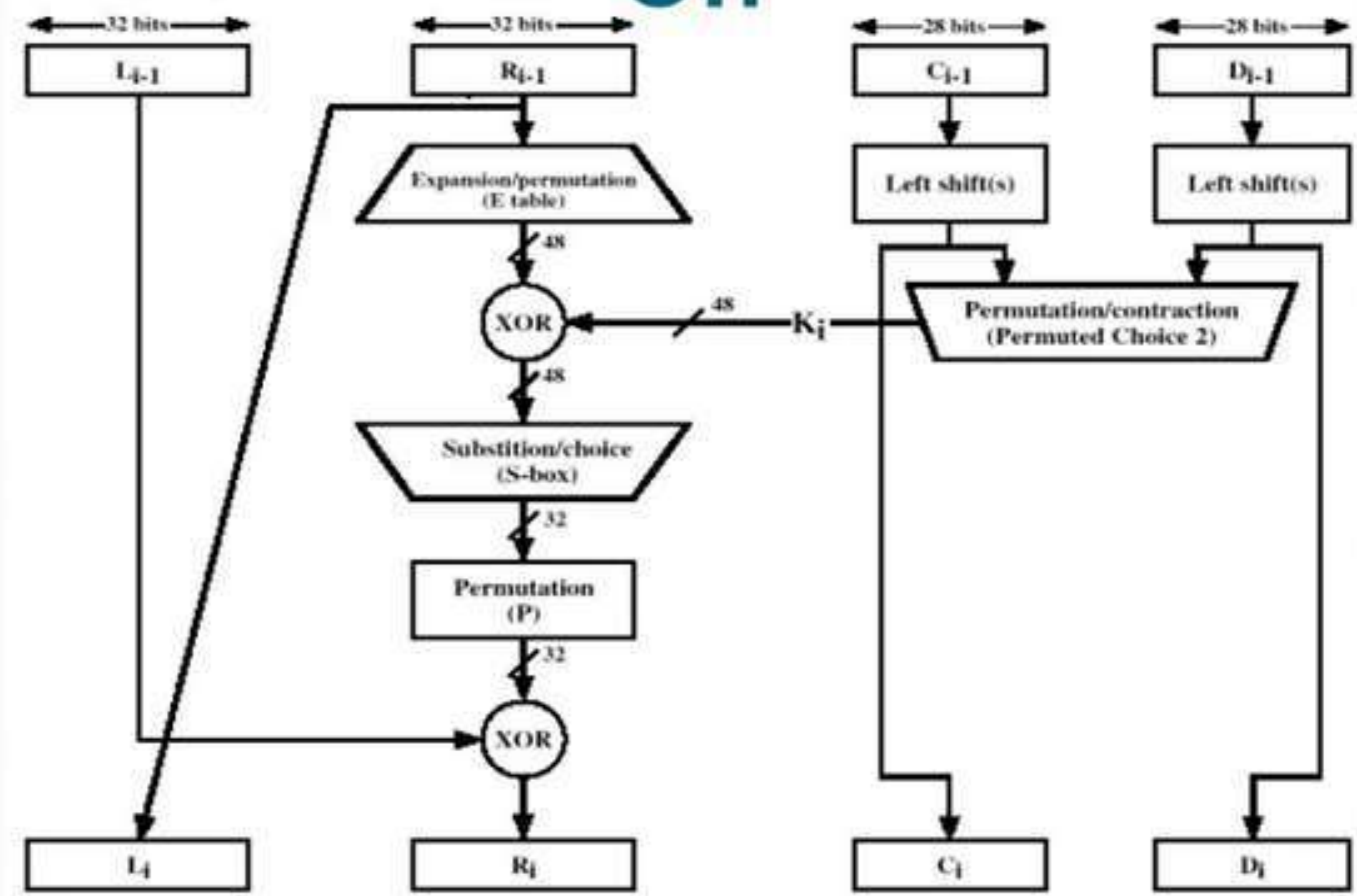


Initial Permutation IP

- It is the first step of the data computation.
- IP reorders the input data bits according to the following initial permutation table.

Initial Permutation							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

The 16 Rounds of F Consist Of:



DES Round Structure

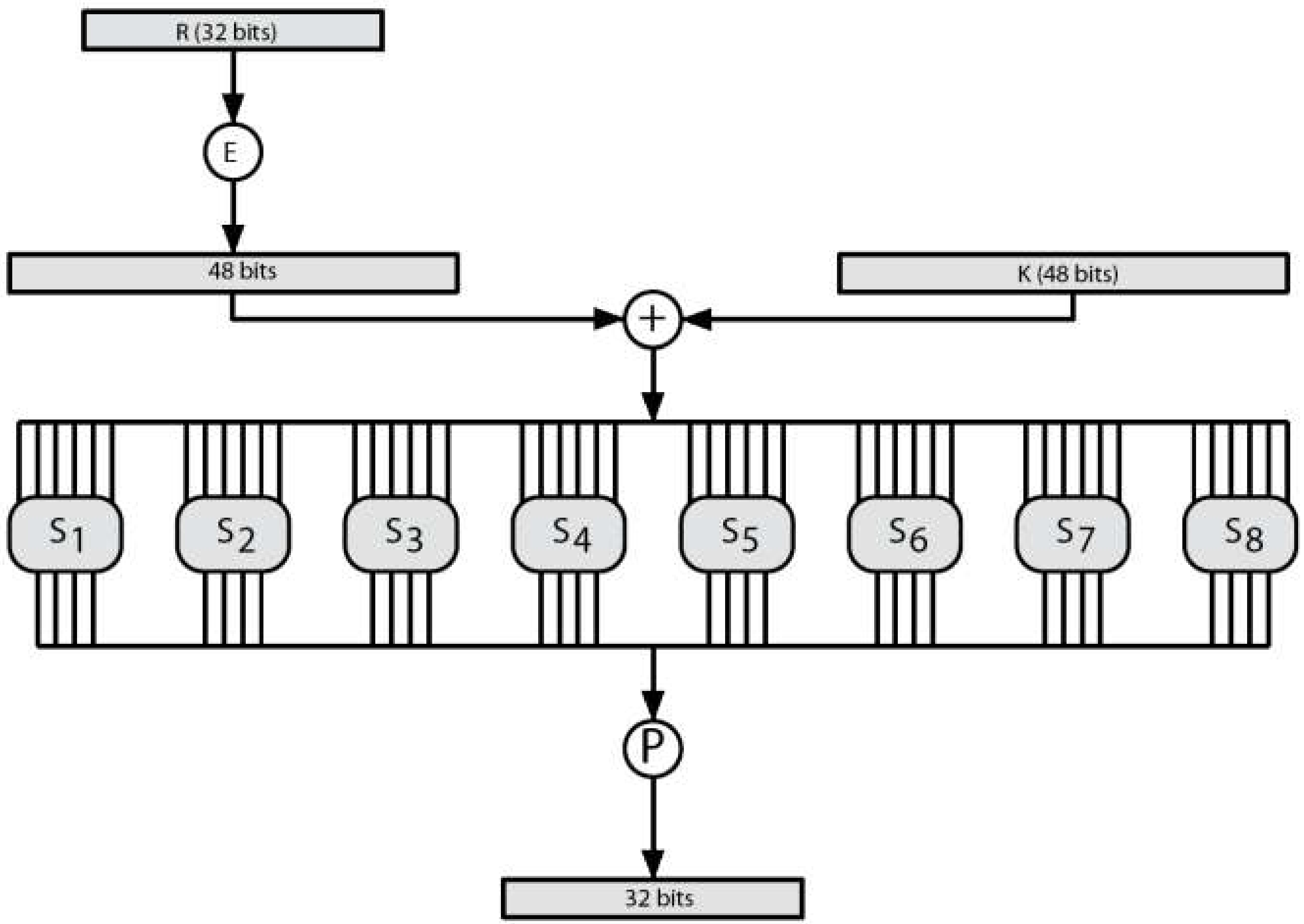
- It divides the 64 bit output of IP into two 32-bit Left & Right halves.
- Function F takes 32-bit Right half and 48-bit subkey and performs the following:
  - It expands 32 bit right half to 48-bits using Expansion P-Box.
  - It adds to subkey using XOR
  - It passes through 8 S-boxes to get 32-bit result
  - It then passes 32 bit result to Permutation.
  - It performs 16 rounds of processing and performs XOR with left 32 bits.

Expansion P-box Table

Expansion P-box table

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

DES Round Structure



Substitution Boxes S

S-box 1

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

S-Boxes

S-box 1:

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S-box 2:

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S-box 3:

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S-box 4:

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S-box 5:

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S-box 6:

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S-box 7:

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S-box 8:

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Substitution Boxes S

- It have eight S-boxes which map 6 to 4 bits
- Each S-box is actually four rows and 16 columns.
- Outer bits 1 & 6 (**row** bits) select one row out of 4 rows.
- Inner bits 2-5 (**col** bits) selects one column out of 16 columns.
- Result is 8 lots of 4 bits, or 32 bits is passed through straight P-Box.

Straight Permutation Table

Straight permutation table

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Data Encryption Standard

- The left and right halves of the output are swapped to produce preoutput.
- **Finally**, the preoutput is passed through a final permutation that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

Data Encryption Standard

The Inverse Initial Permutation is:

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

DES Key Schedule

- It forms subkeys used in each round as follows
  – Initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  – 16 stages consisting of:
    - rotating **each half** separately either 1 or 2 places depending on the **key rotation schedule** K

Number of Key Bits Shifted per Round

Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

    - selecting 24-bits from each half & permuting them by PC2 to produce the subkey for use in round function F.

DES Key Schedule

Every eighth bit is ignored and produces 56 bits.

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

DES Key Schedule

These 56 bits pass through a Permutation Choice one (PC-1) and displays as follows:

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

DES Key Schedule
Permuted Choice-2

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

DES Decryption

- Decryption must be done in reverse order of the encryption process.
- Based on Feistel design, do decryption steps again using subkeys in reverse order (K16 … K1)
  - IP inverse undoes final step of encryption
  - 1st round with K16 undoes 16th encrypt round
  - 16th round with K1 undoes 1st encrypt round
  - Finally the output undoes initial encryption with IP to recover the original data value

Avalanche Effect

- Encryption key is the desirable property of encryption algorithm where a change of **one** input or key bit results in changing approximately **half** output bits
- It makes attempts to guess the key is impossible

Block Cipher Modes of operation

Block Cipher Modes of operation

- Encryption algorithms are divided into two categories based on the input type, as a block cipher and stream cipher.
- **Block cipher** is an encryption algorithm that takes a fixed size of input say *b* bits and produces a ciphertext of *b* bits again.
- If the input is larger than *b* bits it can be divided further.

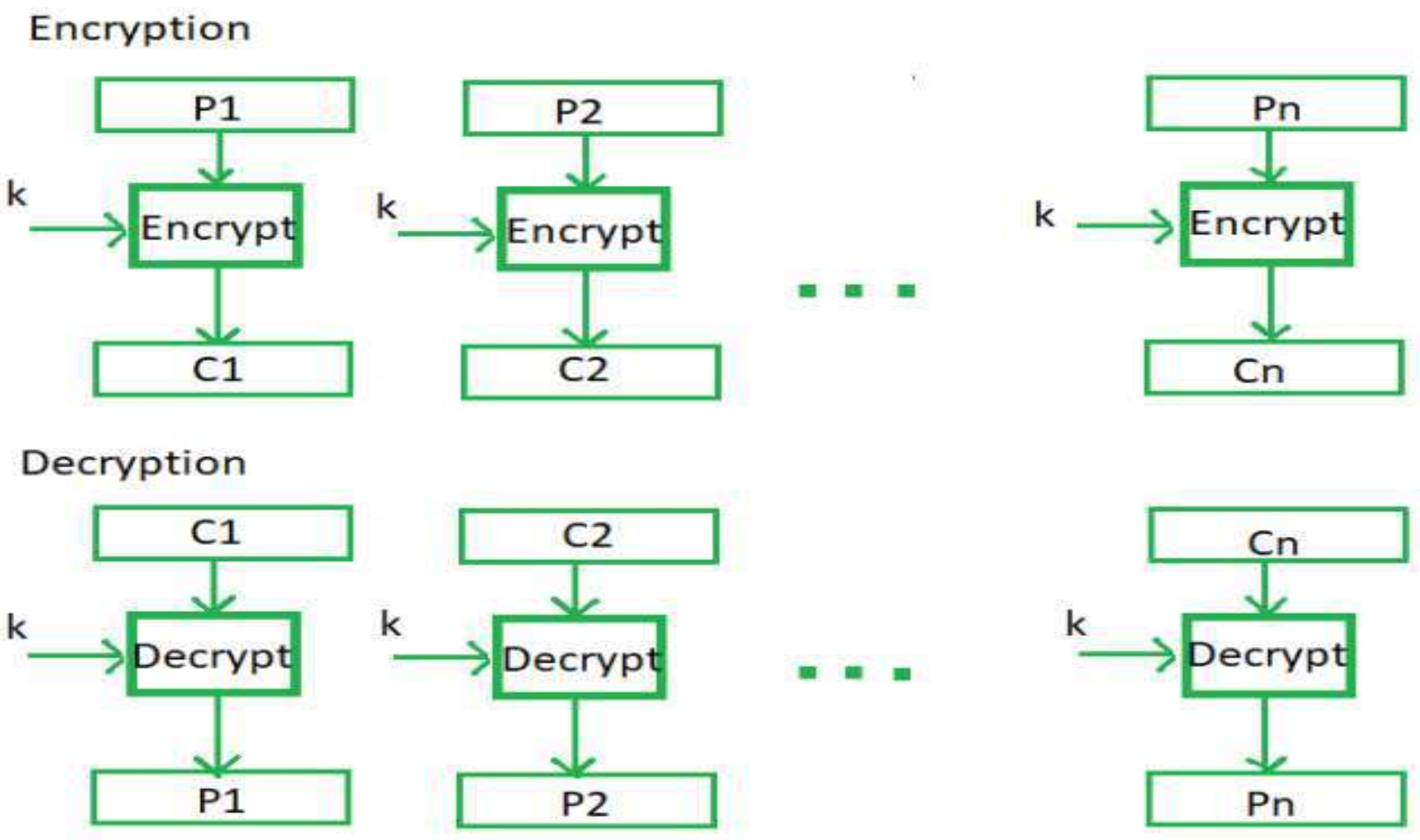
Block Cipher Modes of operation

- For different applications and uses, there are several modes of operations for a block cipher as follows:
  - Electronic code book Mode
  - Cipher Block Chaining
  - Cipher Feedback Mode
  - Output Feedback Mode
  - Counter Mode

Electronic Code Book (ECB)

- Electronic code book is the easiest block cipher mode of functioning.
- It is easier because of direct encryption of each block of input plaintext and output is in form of blocks of encrypted ciphertext.
- Generally, if a message is larger than *b* bits in size, it can be broken down into a bunch of blocks and the procedure is repeated.

Electronic Code Book (ECB)



Electronic Code Book (ECB)

Advantages of using ECB –

- Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.
- Simple way of the block cipher.

Disadvantages of using ECB –

- Prone to cryptanalysis since there is a direct relationship between plaintext and ciphertext.

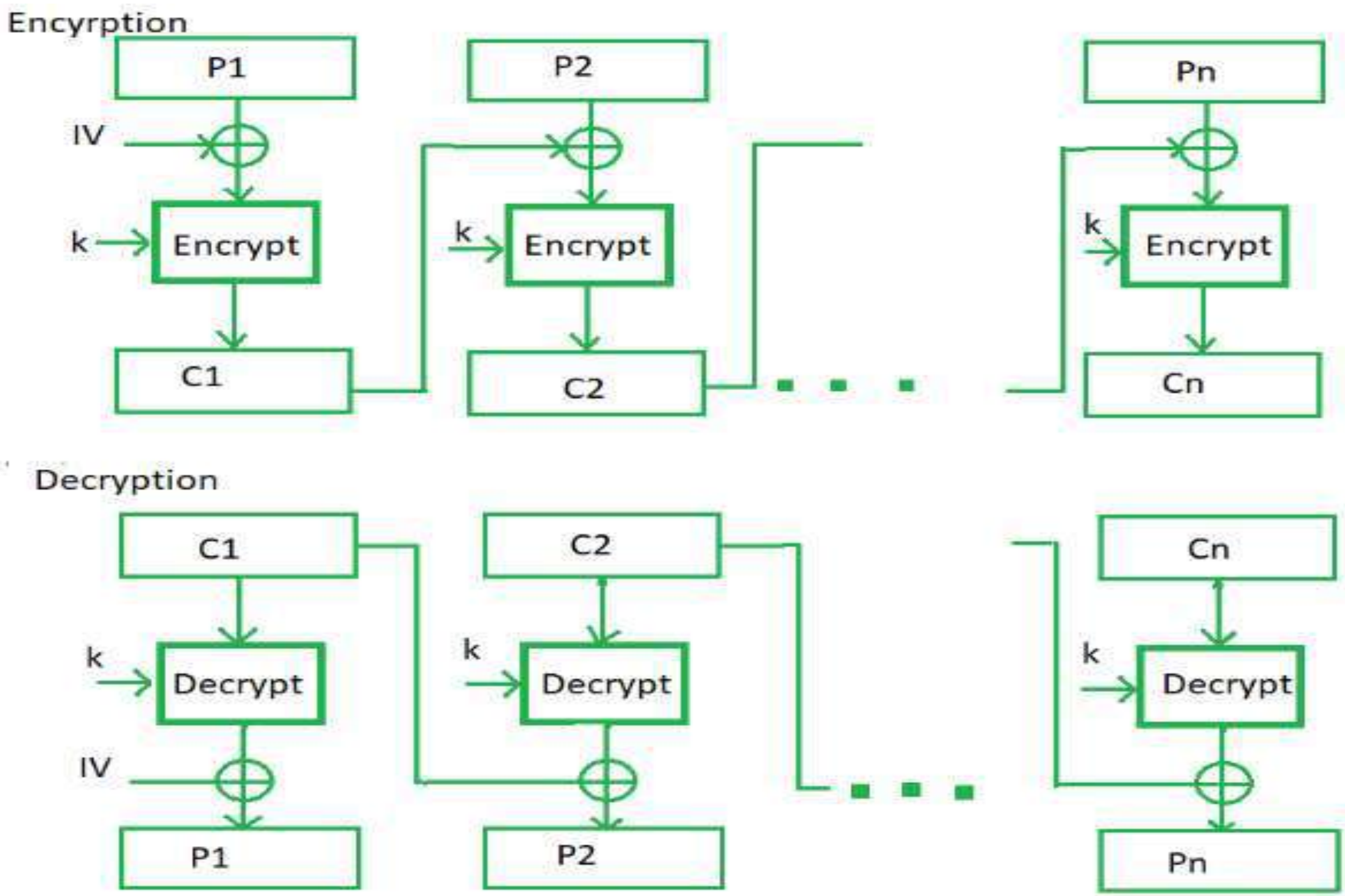
Cipher Block Chaining

- Cipher block chaining or CBC is an advancement made on ECB since ECB compromises some security requirements.
- In CBC, the previous cipher block is given as input to the next encryption algorithm after XOR with the original plaintext block.
- Cipher block is produced by encrypting an XOR output of the previous cipher block and present plaintext block.
- Use Initial Vector (IV) to start process

$$C_i = E_K (P_i \text{ XOR } C_{i-1})$$

$$C_0 = IV$$

Cipher Block Chaining



Cipher Block Chaining

Advantages of CBC –

- CBC works well for input greater than *b* bits.
- CBC is a good authentication mechanism.
- Better resistive nature towards cryptanalysis than ECB.

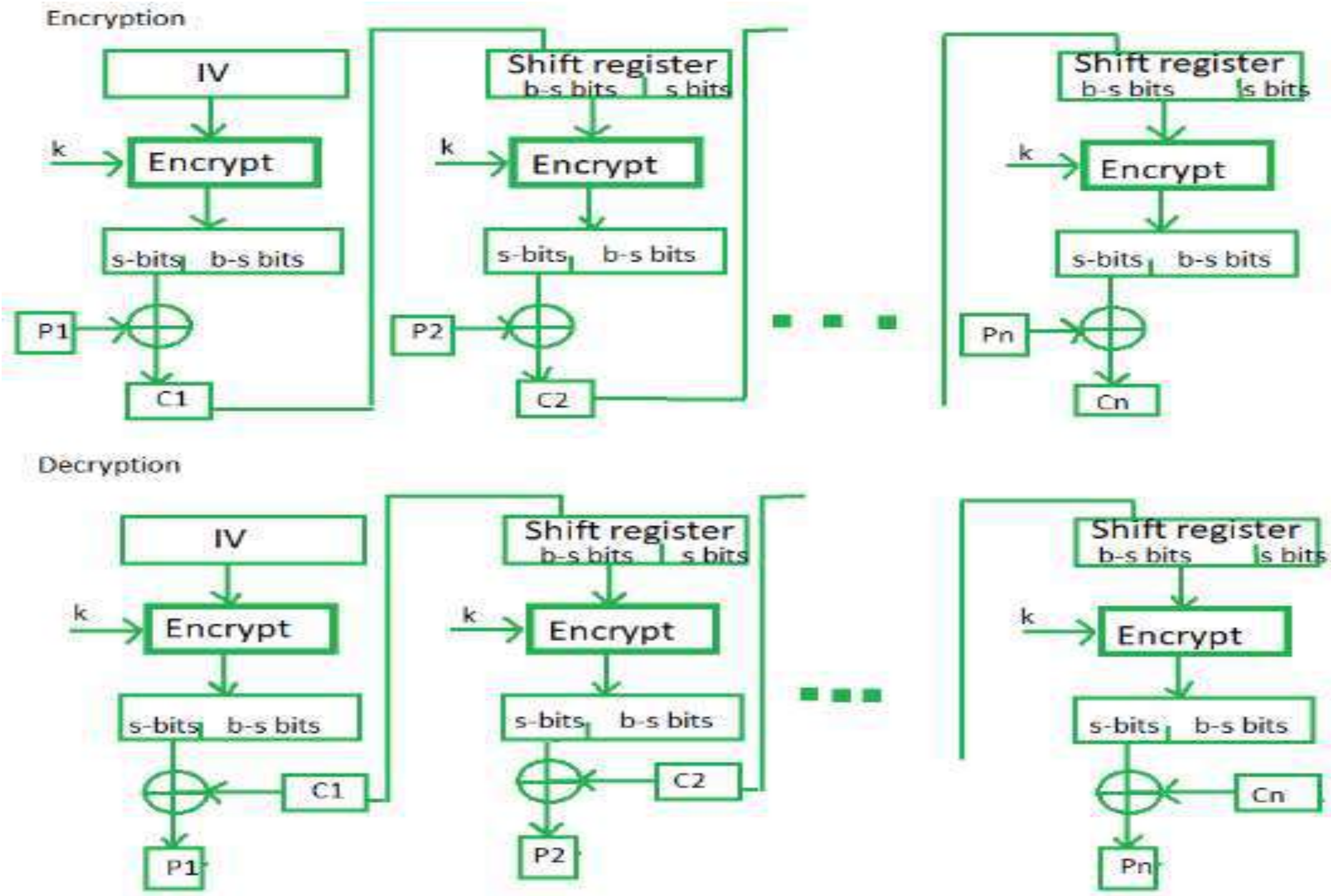
Disadvantages of CBC –

- Parallel encryption is not possible since every encryption requires a previous cipher.

Cipher Feedback Mode (CFB)

- In this mode the cipher is given as feedback to the next block of encryption with some new specifications.
- first, an initial vector IV is used for first encryption
- Output bits are divided as a set of *s* and *b-s* bits, the left-hand side *s* bits are selected and are applied an XOR operation with plaintext bits.
- The result is given as input to a shift register to concatenate to b-s bits and the process continues.
- The encryption and decryption process for the same is shown below, both of them use encryption algorithms.

Cipher Feedback Mode (CFB)



Cipher Feedback Mode (CFB)

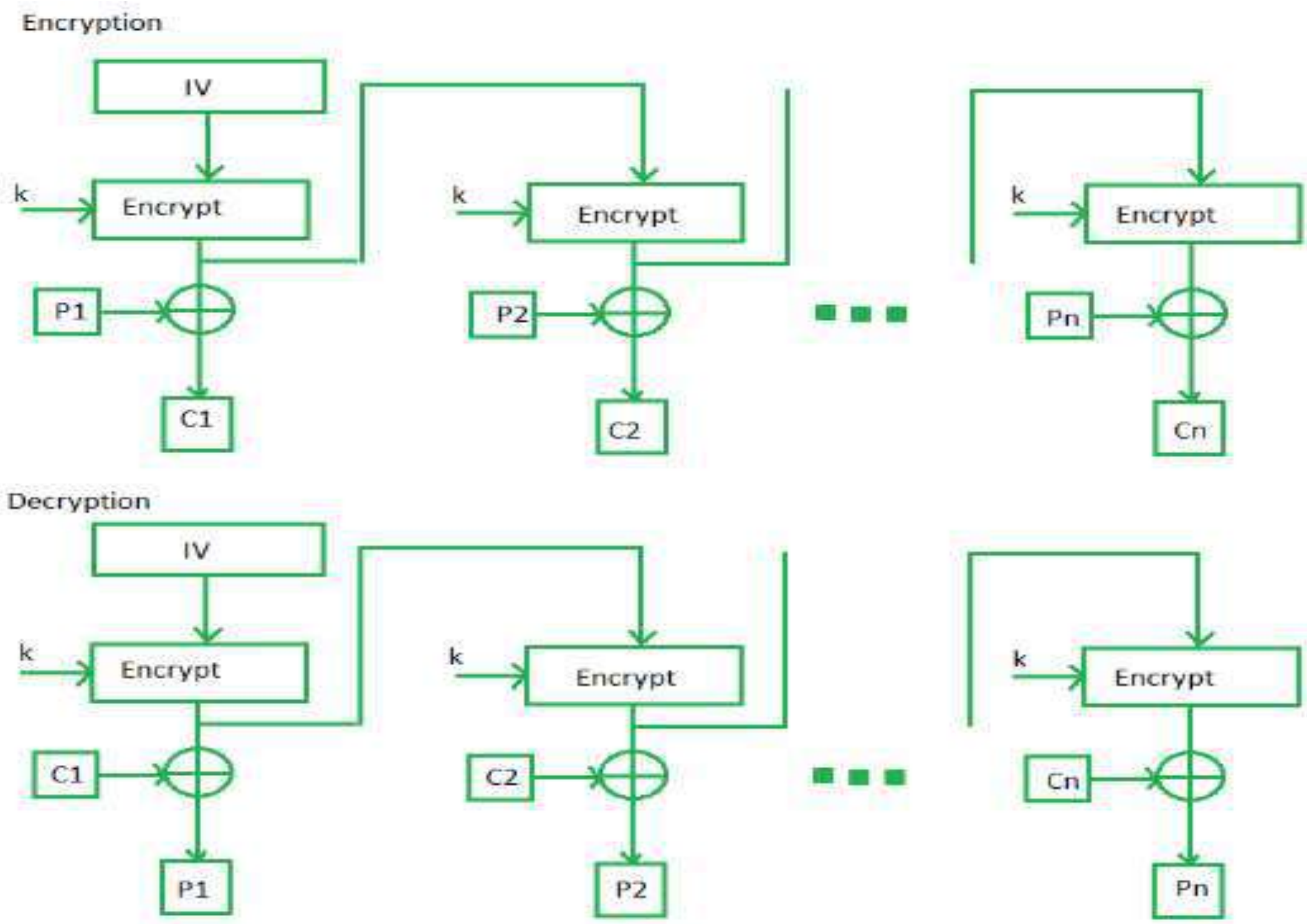
Advantages of CFB –

- Since, there is some data loss due to the use of shift register, thus it is difficult for applying cryptanalysis.

Output Feedback Mode

- The output feedback mode follows nearly the same process as the Cipher Feedback mode except that it sends the encrypted output as feedback instead of the actual cipher which is XOR output.
- In this output feedback mode, all bits of the block are sent instead of sending selected *s* bits.
- The Output Feedback mode of block cipher holds great resistance towards bit transmission errors.
- It also decreases the dependency or relationship of the cipher on the plaintext.

Output Feedback Mode



Output Feedback Mode

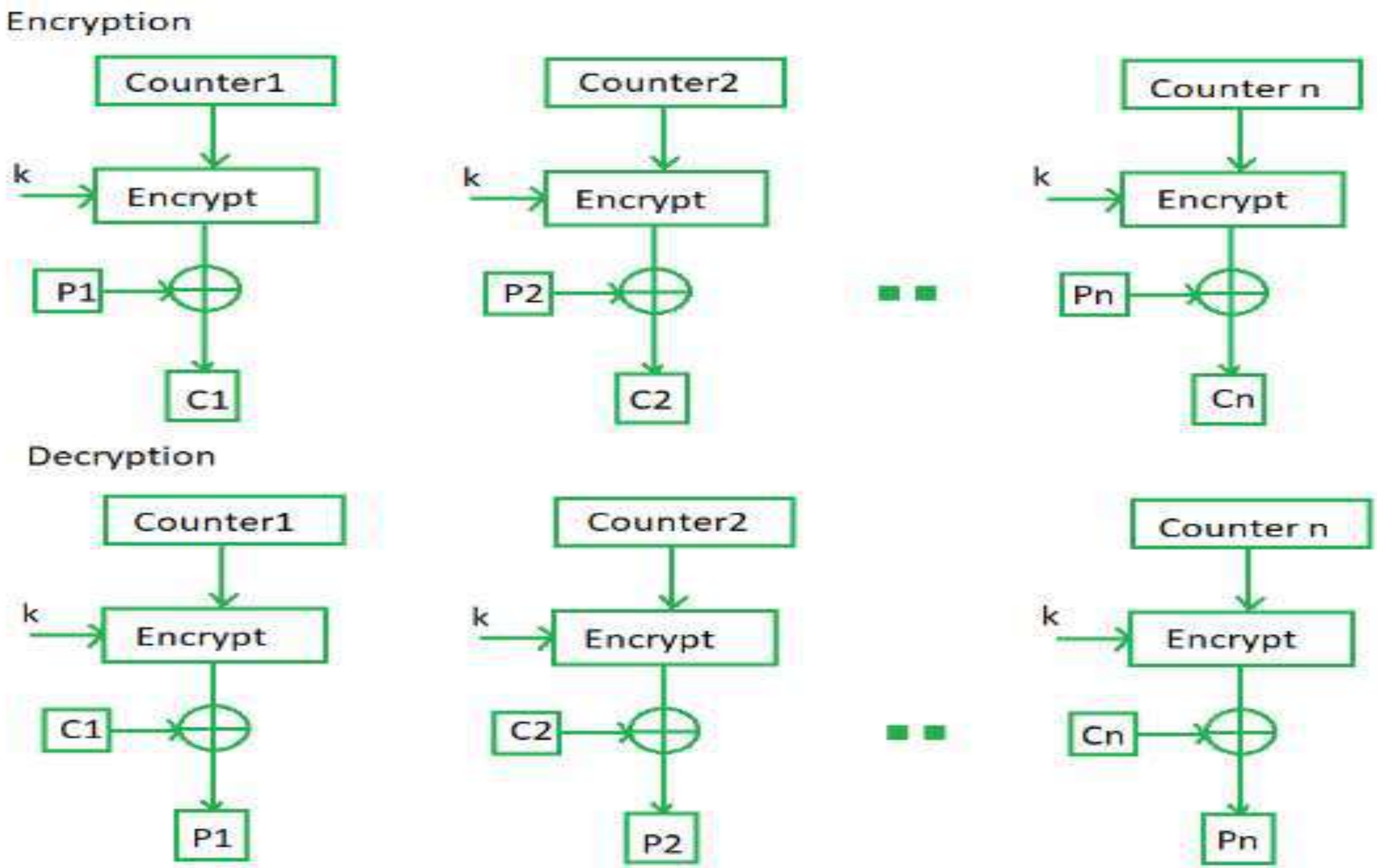
Advantages of OFB –

- In the case of CFB, a single bit error in a block is propagated to all subsequent blocks. This problem is solved by OFB as it is free from bit errors in the plaintext block.

Counter Mode

- Counter Mode or CTR is a simple counter-based block cipher implementation.
- Every time a counter-initiated value is encrypted and given as input to XOR with plaintext which results in ciphertext block.
- The CTR mode is independent of feedback use and thus can be implemented in parallel.

Counter Mode



Counter Mode

Advantages of Counter –

- Since there is a different counter value for each block, the direct plaintext and ciphertext relationship is avoided. This means that the same plain text can map to different ciphertext.
- Parallel execution of encryption is possible as outputs from previous stages are not chained as in the case of CBC.

Public Key Cryptography

Public Key Cryptography

- It is used two keys for encryption and for decryption.
  - a public-key, which may be known by anybody, and can be used to encrypt messages
  - a private-key, known only to the recipient, used to decrypt messages

- It has six ingredient

1 Plain text

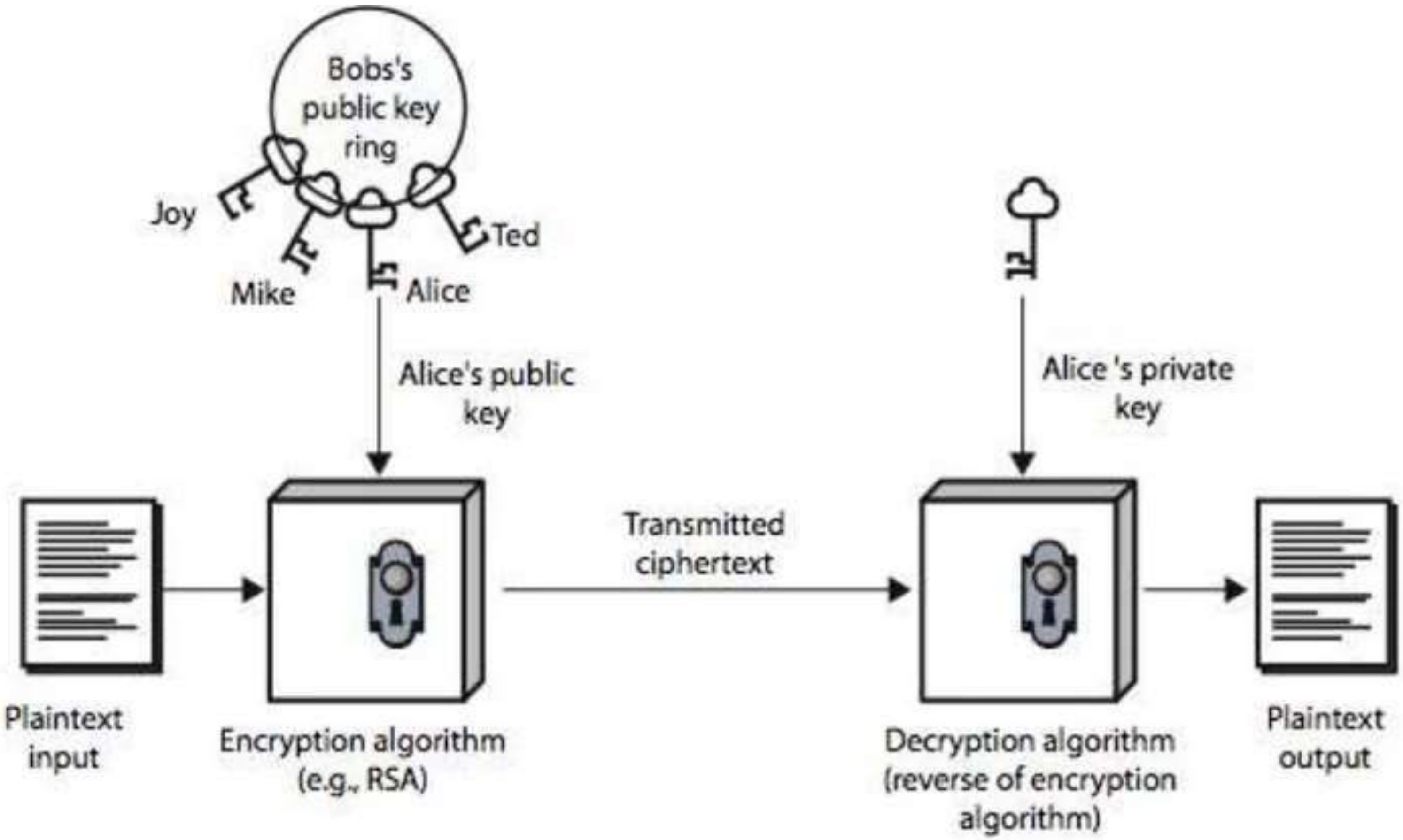
2 Encryption algorithm

3 Public and private keys

4 Ciphertext

5 Decryption algorithm

Public Key Cryptography

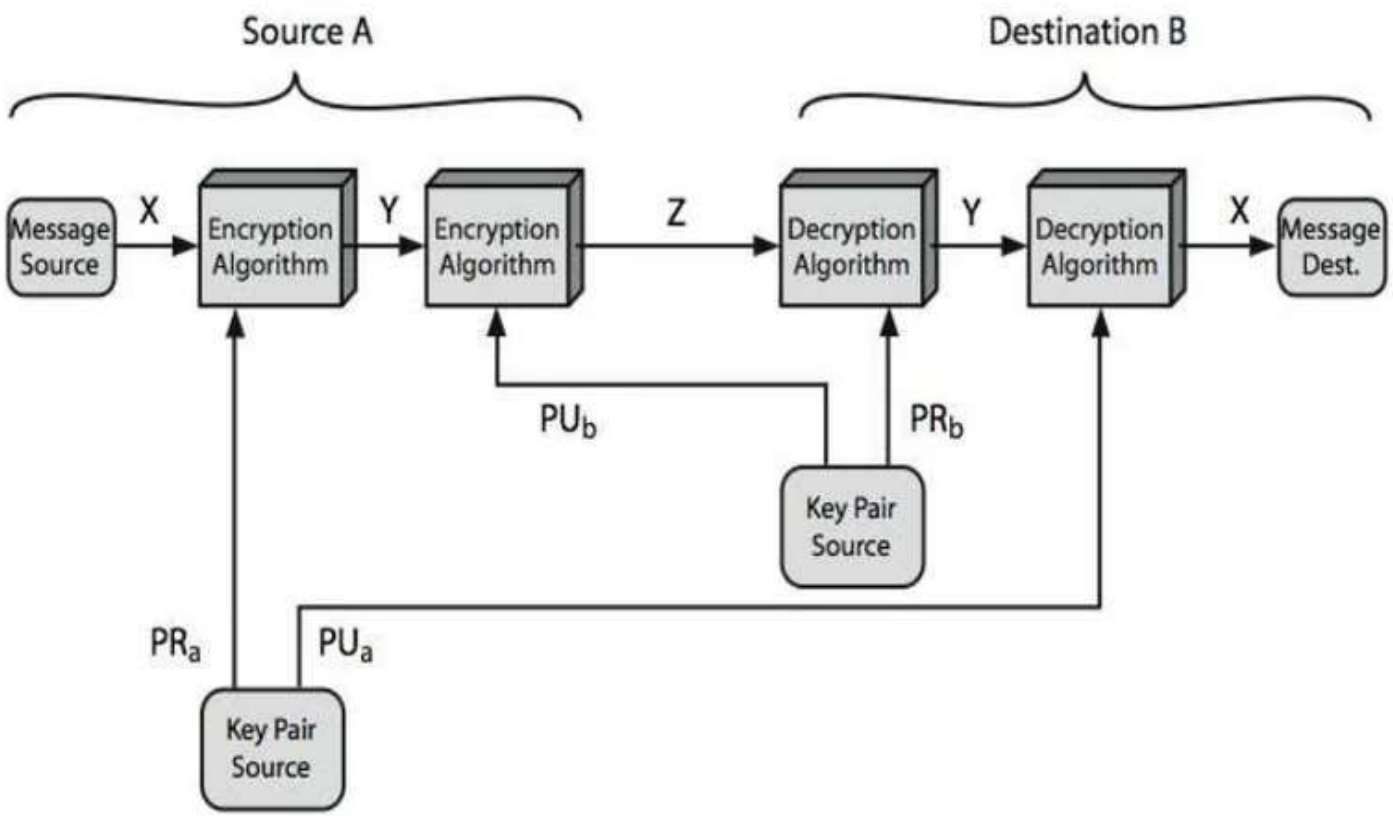


(a) Encryption

Public Key Characteristics

- Public-Key algorithms rely on two keys where:
  - it is computationally infeasible to find decryption key knowing only algorithm & encryption key
  - it is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
  - either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)

Public Key Cryptosystem: Authentication and secrecy



Public key Cryptosystem : Authentication and secrecy

Requirements of Public Key Cryptography

1. It is easy for party **B** to generate a pair of keys (public key $\mathbf{PU_b}$, Private key $\mathbf{PR_b}$).
2. It is easy for a sender **A** , knowing the public key and message to be encrypt. $\mathbf{C=E(PU_b, M)}$
3. It is easy for receiver **B** to decrypt the resulting ciphertext using the private key . $\mathbf{M=D(PR_b,C)=D[PR_b,E(PU_b,M)]}$
4. It is infeasible for an any person , to know the public key $\mathbf{PU_b}$ to determine the private key $\mathbf{PR_b}$.
5. It is infeasible for any person to know the public key $\mathbf{PU_b}$ and a ciphertext **C** to recover the original message **M.**
6. Two keys can be applied in either order

$$\mathbf{M=DP[PU_b, E(PR_b,M)] = D[PR_b,E(PU_b, M)]}$$

RSA Algorithm

RSA Algorithm

RSA

- Invented by **R**ivest, **S**hamir & **A**dleman of MIT in 1977
- It is a best known & widely used public-key scheme.
- It is a **block cipher algorithm** in which palintext and ciphertext integers between 0 to n-1 for some ***n*.**
- A typical size for ***n*** is 1024 bits or 309 decimal digits.

RSA Algorithm

Key Generation

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

Encryption

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

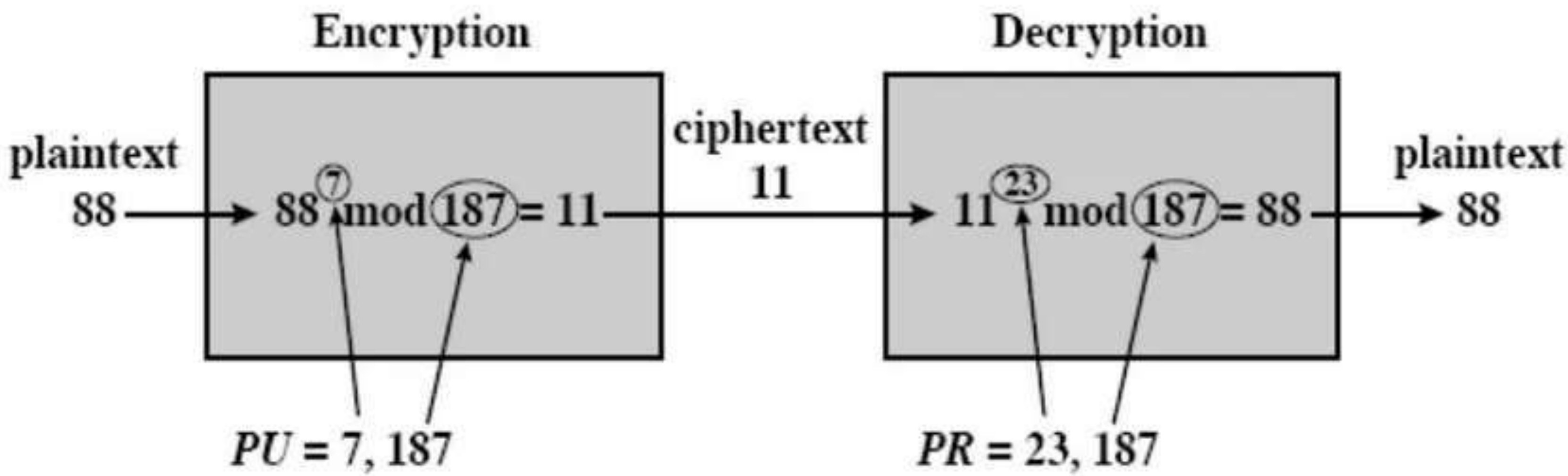
Decryption

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

RSA Algorithm: Example

- Select two large primes: p, q, p ? q
  p = 17, q = 11
- $n = p \times q = 17 \times 11 = 187$
- Calculate $\Phi = (p-1)(q-1) = 16x10 = 160$
- Select e, such that $lcd(\Phi, e) = 1$; $0 < e < \Phi$
  say, e = 7
- Calculate d such that $de \bmod \Phi = 1$
  - 160k+1 = 161, 321, 481, 641
  - Check which of these is divisible by 7
  - 161 is divisible by 7 giving d = 161/7 = 23
- Key 1 = {7, 187}, Key 2 = {23, 187}

RSA Algorithm



Example of RSA Algorithm

RSA Example-2

- Let p= 3 and q=5,
- n= 3 X 5 =15
- Q(n)= (3-1) * (5-1) = 2 x 4= 8
- Select e such that **gcd**(Q(n), e) =1 where, 1<e<Q(n)
- Say e=3 (any prime number)
- Calculate d , such that d e mod Q(n)=1
- 8k+1= 9, 17,25, 33, 41........where k=1,2,3,4....
- Now check which number is divisible by 3.
- 33 is divisible by 3 .So, d=33/3=11. //9 is also divisible by 3.
- Now k1=(3,15) and K2=(11,15)
- Take plan text M =**13 , where (M<n)**
- Encryption C= 13^3 mod 15 =7
- Decryption D= 7^{11} mod 15 =**13**

RSA Exercise Problems

- Perform encryption and decryption using the RSA algorithm for the following

1. p=3, q=11, e=7, M=5
2. P=5,q=11, e=3 , M=9

Diffie Hellman Key Exchange

Diffie –Hellman Key Exchange

in 1976

- It is used by two users to **securely exchange a key** that can be used for subsequent encryption of messages.

a public-key distribution scheme
  - cannot be used to exchange an arbitrary message
  - rather it can establish a common key
  - known only to the two participants

value of key depends on the participants (and their private and public key information)

based on mathematical principles

Diffie Hellman Key Exchange

Global Public Elements
q = prime number(300 decimal, i.e. 1024 bits)
α = Integer

User A key Generation
Select private Xa , Xa < q
Calculate public Ya , Ya= α^{Xa} mod q

User B Key Generation
Select private Xb , Xb < q
Calculate public Yb , Yb= α^{Xb} mod q

Diffie Hellman Key Exchange

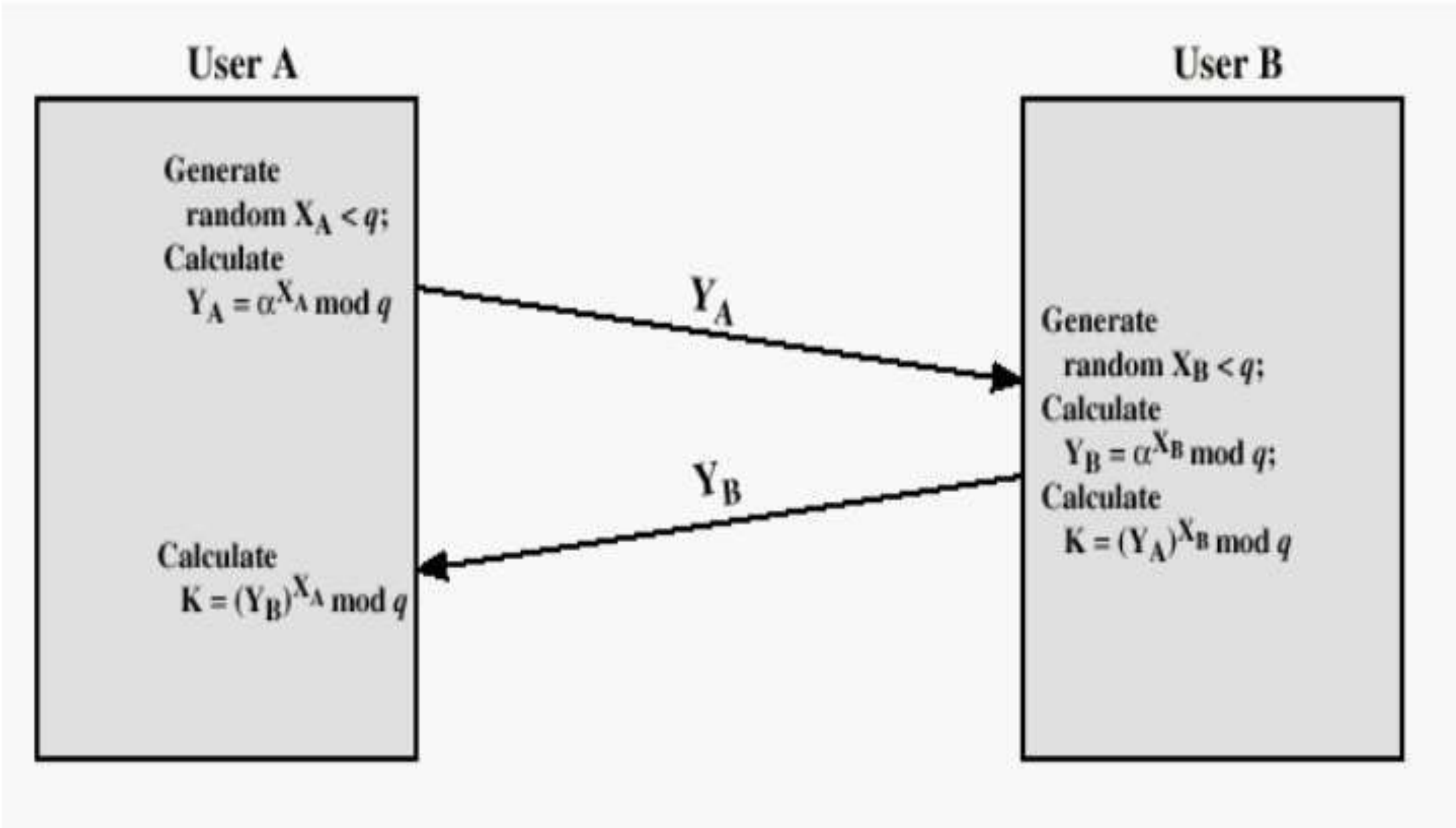
Generation of secret key by user A

$$K=(Y_b)^{X_a} \bmod q$$

Generation of secret key by user B

$$K=(Y_a)^{X_b} \bmod q$$

Diffie Hellman Key Exchange



Diffie Hellman Key Exchange

- users Alice & Bob who wish to swap keys:
- agree on prime $q=\mathbf{353}$ and $\alpha=\mathbf{3}$
- select random secret keys:
  - A chooses $x_A=\mathbf{97}$, B chooses $x_B=\mathbf{233}$
- compute respective public keys:
  - $y_A=\mathbf{3}^{97} \bmod \mathbf{353} = \mathbf{40}$ (Alice)
  - $y_B=\mathbf{3}^{233} \bmod \mathbf{353} = \mathbf{248}$ (Bob)
- compute shared session key as:
  - $K_{AB}= y_B^{xA} \bmod \mathbf{353} = \mathbf{248}^{97} = 160$ (Alice)
  - $K_{AB}= y_A^{xB} \bmod \mathbf{353} = \mathbf{40}^{233} = 160$ (Bob)

Diffie Hellman Key Exchange

users Alice & Bob who wish to swap keys:
agree on prime q=5 and α=7
select random secret keys:
- A chooses x_A= 8, B chooses x_B= 13

Diffie Hellman Key Exchange

Using diffie- hellman key exchange techniques ,Find A's public key Y_A and B's public key Y_B . If, $q=71$ and $\alpha = 7$, $X_A = 5$ and $X_B = 12$

Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES)

- Advanced Encryption Standard (AES) is a highly trusted **encryption algorithm** used to secure data by converting it into an unreadable format without the proper key.
- It is developed by the National Institute of Standards and Technology (NIST) in 2001.
- **AES encryption** uses various **key lengths** (128, 192, or 256 bits) to provide strong protection against unauthorized access.
- This **data security** measure is efficient and widely implemented in securing **internet communication**, protecting **sensitive data**, and encrypting files.

Advanced Encryption Standard (AES)

- AES is a Block Cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each. That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text.
- AES relies on the substitution-permutation network principle, which is performed using a series of linked operations that involve replacing and shuffling the input data.

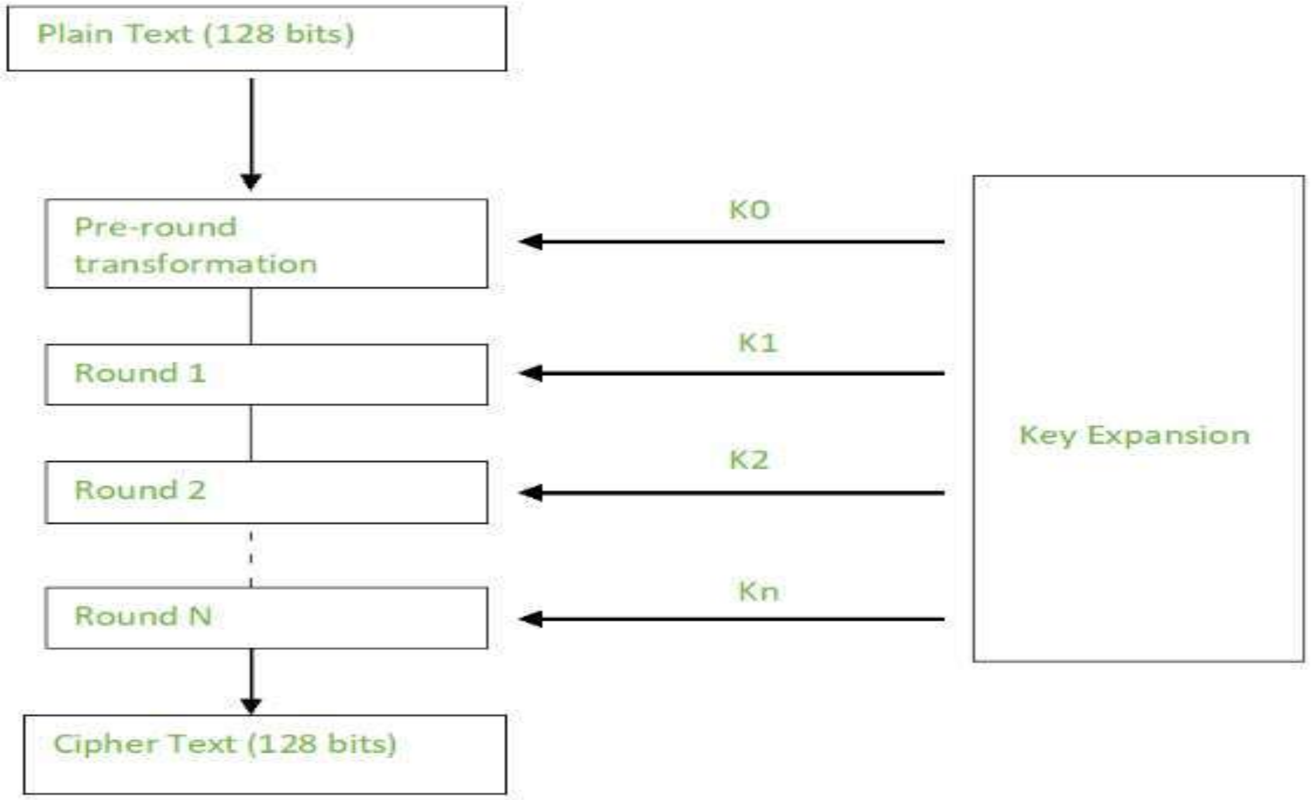
Working

- AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.
- The number of rounds depends on the key length as follows :

N (Number of Rounds)	Key Size (in bits)
10	128
12	192
14	256

Working

- A Key Schedule algorithm calculates all the round keys from the key.
- The initial key is used to create many different round keys which will be used in the corresponding round of the encryption.



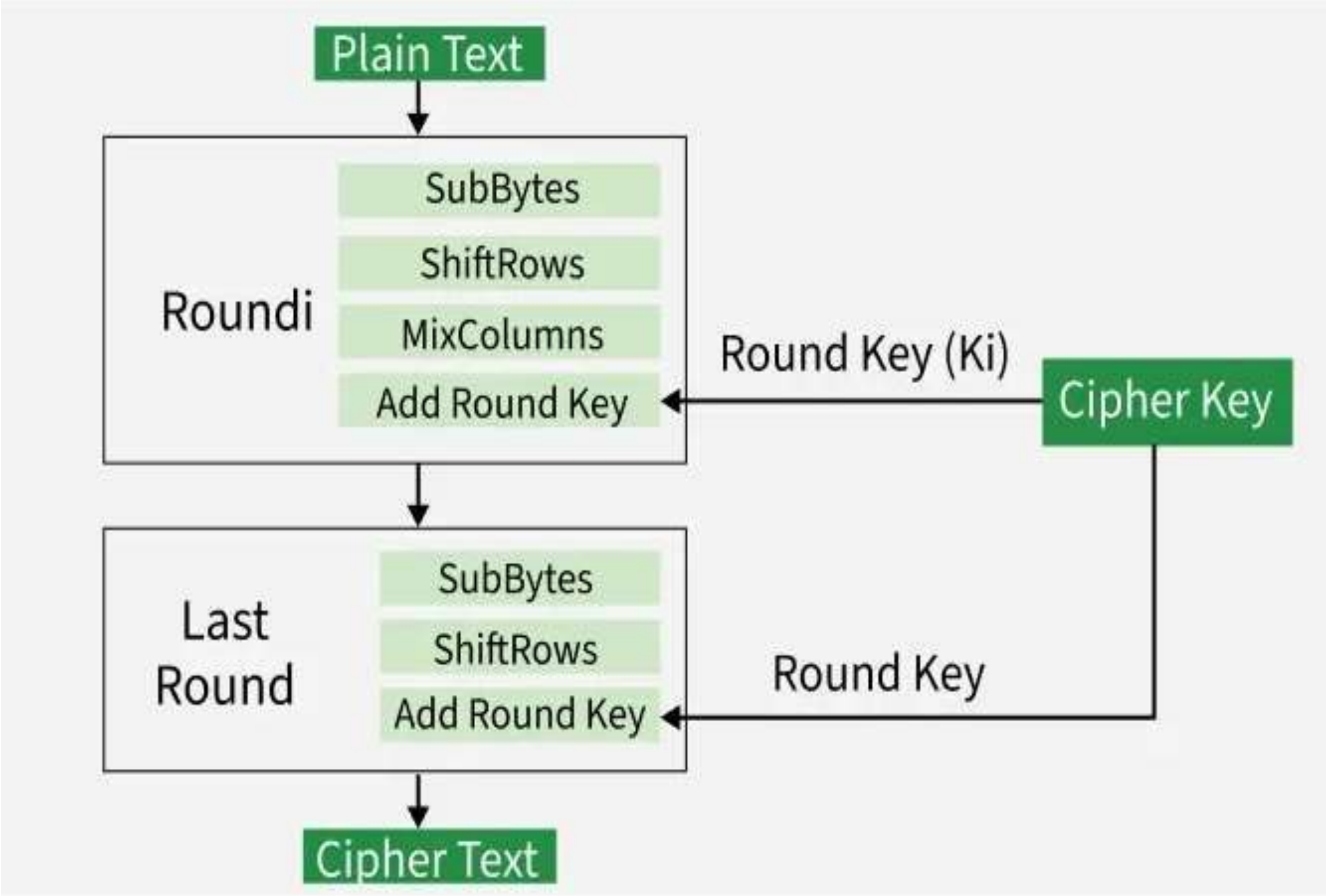
Key Generation

- KeyExpansions- In the key Expansion process the given 128 bits cipher key is stored in [4]x[4] bytes matrix (16*8=128 bits) and then the four column words of the key matrix is expanded into a schedule of 44 words (44*4=176) resulting in 11 round keys (176/11=16 bytes or 128 bits).
- Number of round keys = Nr + 1. Where Nr is the number of rounds (which is 10 in case of 128 bits key size) So here the round keys = 11.

Encryption

- AES considers each block as a 16-byte (4 byte x 4 byte = 128) grid in a column-major arrangement.

[b0 | b4 | b8 | b12 |
| b1 | b5 | b9 | b13 |
| b2 | b6 | b10| b14 |
| b3 | b7 | b11| b15]



Step1. Sub Bytes

- This step implements the substitution.
- In this step, each byte is substituted by another byte. It is performed using a lookup table also called the S-box.
- This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16-byte (4 x 4) matrix like before.
- The next two steps implement the permutation.

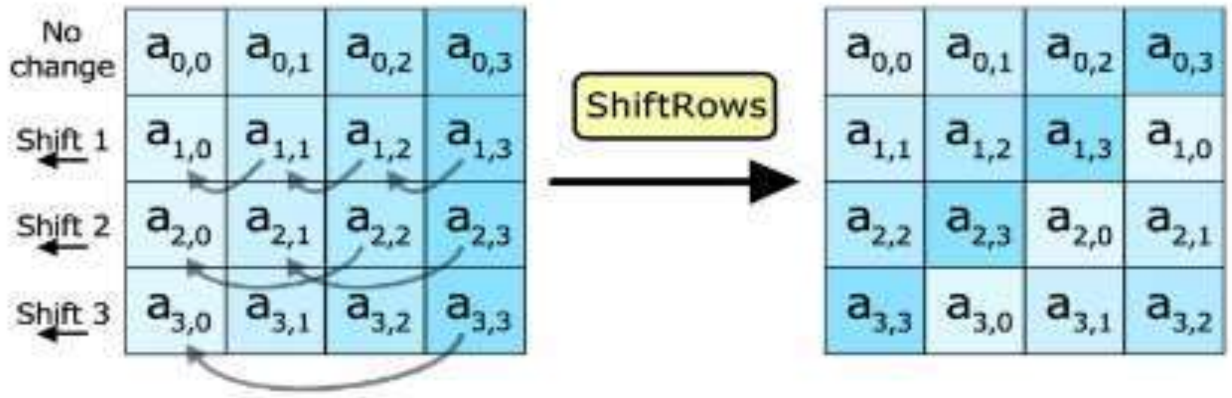
Step1. Sub Bytes

For an element {d1} corresponding value is {3e}

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

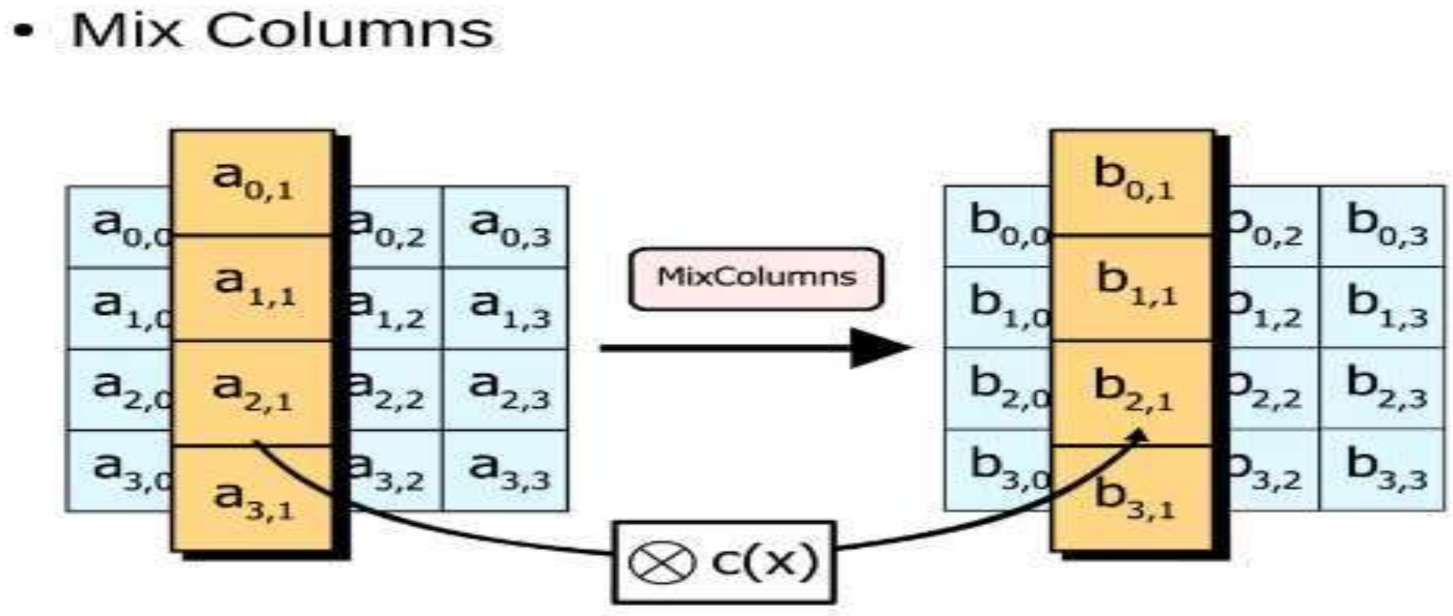
Step2. Shift Rows

- This step is just as it sounds. Each row is shifted a particular number of times.
- The first row is not shifted
- The second row is shifted once to the left.
- The third row is shifted twice to the left.
- The fourth row is shifted thrice to the left.
- A left circular shift is performed



Step 3: Mix Columns

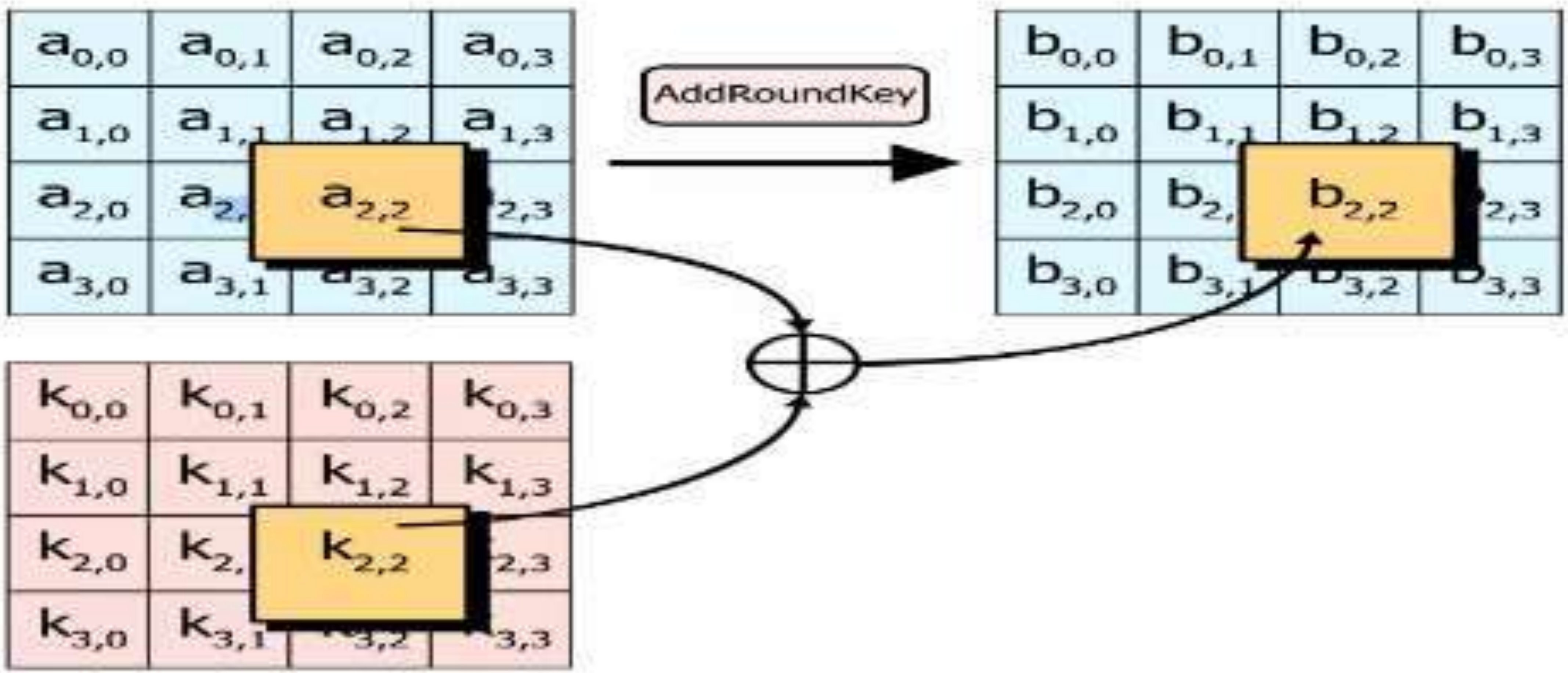
- This step is a matrix multiplication.
- Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.
- **This step is skipped in the last round.**



Step 4: Add Round Keys

- Now the resultant output of the previous stage is XOR-ed with the corresponding round key.
- Here, the 16 bytes are not considered as a grid but just as 128 bits of data.
- After all these rounds 128 bits of encrypted data are given back as output.
- This process is repeated until all the data to be encrypted undergoes this process.

Step 4: Add Round Keys



Decryption

- The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes.
- Each 128 blocks goes through the 10,12 or 14 rounds depending on the key size.
- The stages of each round of decryption are as follows :
  - Add round key
  - Inverse Mix Columns.
  - Shift Rows
  - Inverse Sub Byte

  The decryption process is the encryption process done in reverse order.

Applications of AES

- AES is widely used in many applications which require secure data storage and transmission.
- Some common use cases include:
  - **Wireless security**
  - **Database Encryption**
  - **Secure communications**
  - **Data storage**
  - **Virtual Private Networks (VPNs)**
  - **Secure Storage of Passwords**
  - **File and Disk Encryption**

Unit 3
Message Authentication Codes – Hash Functions And Digital Signatures

Message Authentication Codes – Hash Functions And Digital Signatures

- Message Authentication Requirements – Message Authentication Functions – Message Authentication Codes – Hash Functions – Security of Hash Functions and MACs – Hash algorithms – SHA – HMAC
- Digital Signatures –Digital Signature Standard(DSS) – Authentication applications – Kerberos – X.509 Authentication Service

Authentication

- The assurance that the communicating entity is one that it claims to be.

Message Authentication requirements:

- The following attacks can be identified across the network.

1. Disclosure- Releasing the message content to any person, not possessing the appropriate cryptographic key
2. Traffic analysis- Identify the traffic between the parties i.e., the number and length of the message.
3. Masquerade- one entity pretending to act like another entity

Message Authentication requirements:

4. Content modification– changes to the content of msg like insertion, deletion, modification, transposition.

5. Sequence modification–any modification to a sequence of msg between parties like insertion, deletion, etc

6. Timing modification–delay or replay of msgs (In connection-oriented and connectionless)

7. Source repudiation–Denial of transmission of message by source

8. Destination repudiation– Denial of receipt of msg by destination

Message authentication functions

- Authentication and digital signature mechanism has 2 levels of functions:

1. Lower level– some sort of function that produces an authenticator: a value to be used to authenticate a message
2. Higher level– The lower level function is used as a primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message.

- Types of function used to produce authenticator

1. Hash function- A function that maps the message of any length into a fixed length hash value (serves as a authenticator)
2. Message encryption- The cipher text of the entire message serves as a authenticator
3. Message authentication code (MAC)- The function of a message and secret key that produces a fixed length value which is the authenticator

BASIC USES OF MESSAGE ENCRYPTION

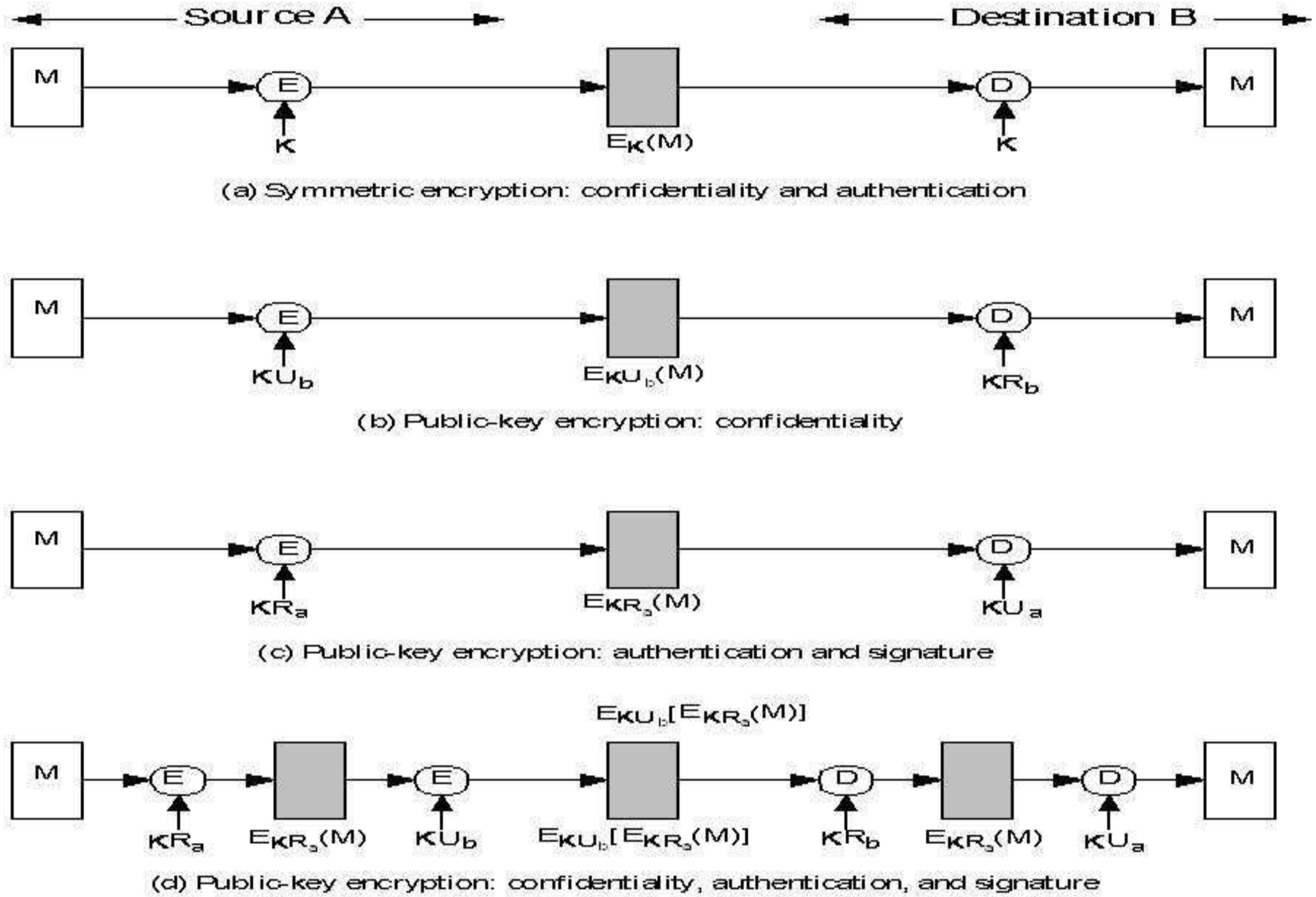


Figure 11.1 Basic Uses of Message Encryption

Message encryption

- In symmetric encryption , only a and b knows the key. Same key is used at both sender and receiver side.
- Public key encryption provides confidentiality, using public key of the receiver for encryption.
- Public key encryption provides authentication and digital signature using private key of the sender for encryption.
- Public key encryption provides confidentiality, authentication and digital signature.

Message authentication code (MAC)

- An authentication technique which involves the use of a secret key to generate a small fixed size block of data known as cryptographic checksum or MAC that is appended to the message.
- MAC is calculated as

MAC=c(k,m)

Where c=MAC function

m=Input message

k=shared key

- The MAC + message is transmitted to the intended recipient.

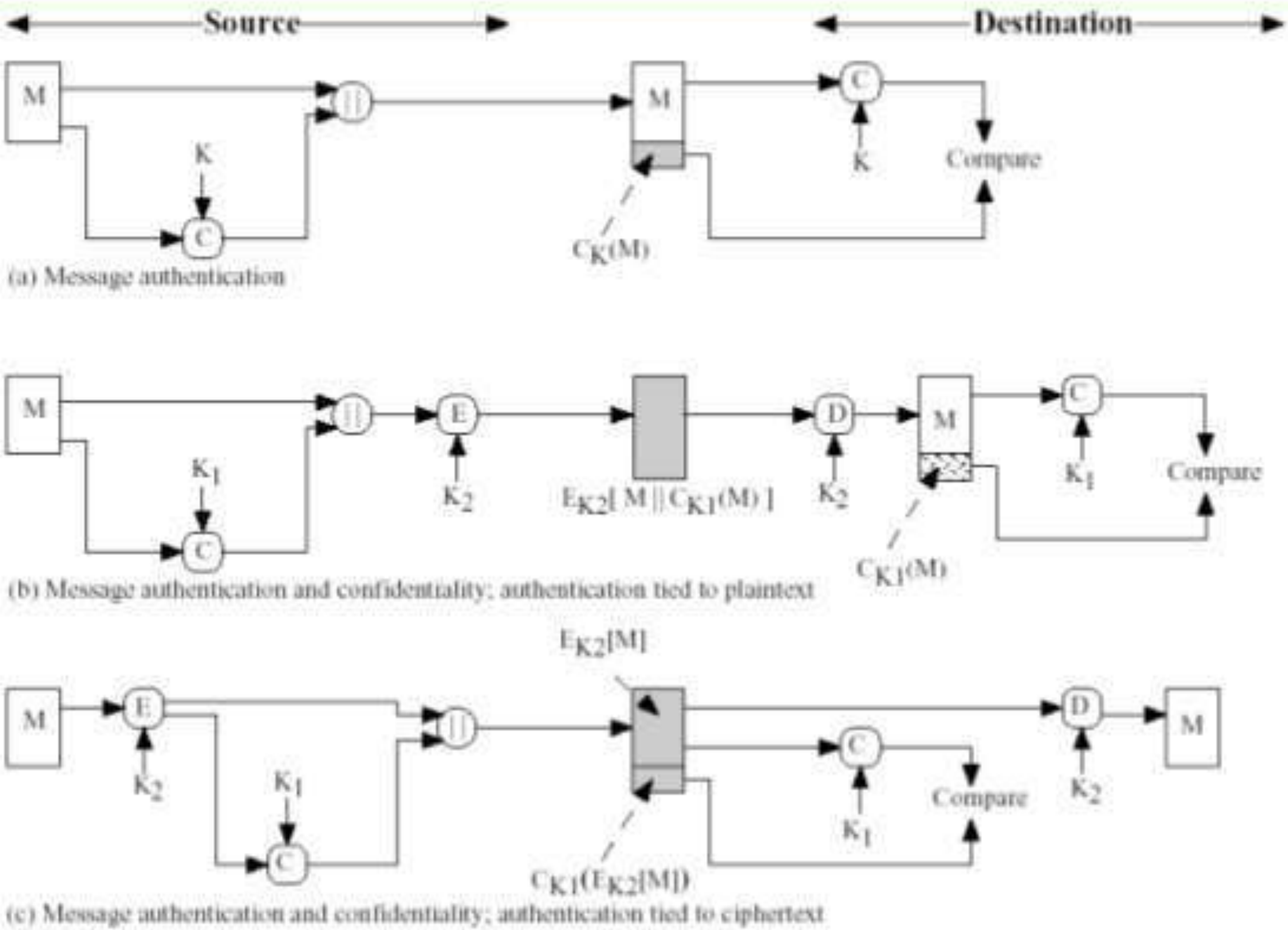
Message authentication code (MAC)

- The recipient performs the same calculation on the received message, using the same secret key to generate a new MAC.
- The received MAC is compared to the calculated MAC and if it matches then

1. The receiver is assured that the message has not been altered
2. The receiver is assured that the message is from a right sender
3. If the message includes a sequence number then the receiver can be assured of proper sequence , because an attacker can’t alter the sequence number.

- MAC function is similar to encryption, one difference is that the MAC algorithm need not be reversible, as it for decryption.
- It is a many to one function
- If there is a n-bit MAC, then there are 2^n possible MACs.
- For example, 100 bit message and 10 bit MAC will produce 2^100/2^10=2^90 different messages.
- If a 5-bit key is used, then there are 2^5=32 different mappings from the set of MAC values.

Basic Uses of MAC



Situations when MAC is used

1. There are a number of applications in which the same message is to be broadcasted to a number of destinations. Eg. Notifications to the user that the network is now unavailable.
2. When there is a heavy load of messages, where client can't be afford to decrypt all incoming messages, authentication is carried out on selective basis (random checking)
3. The computer program can be executed without having to decrypt it everytime. However if MAC is attached, it could be checked whenever the assurance was required.

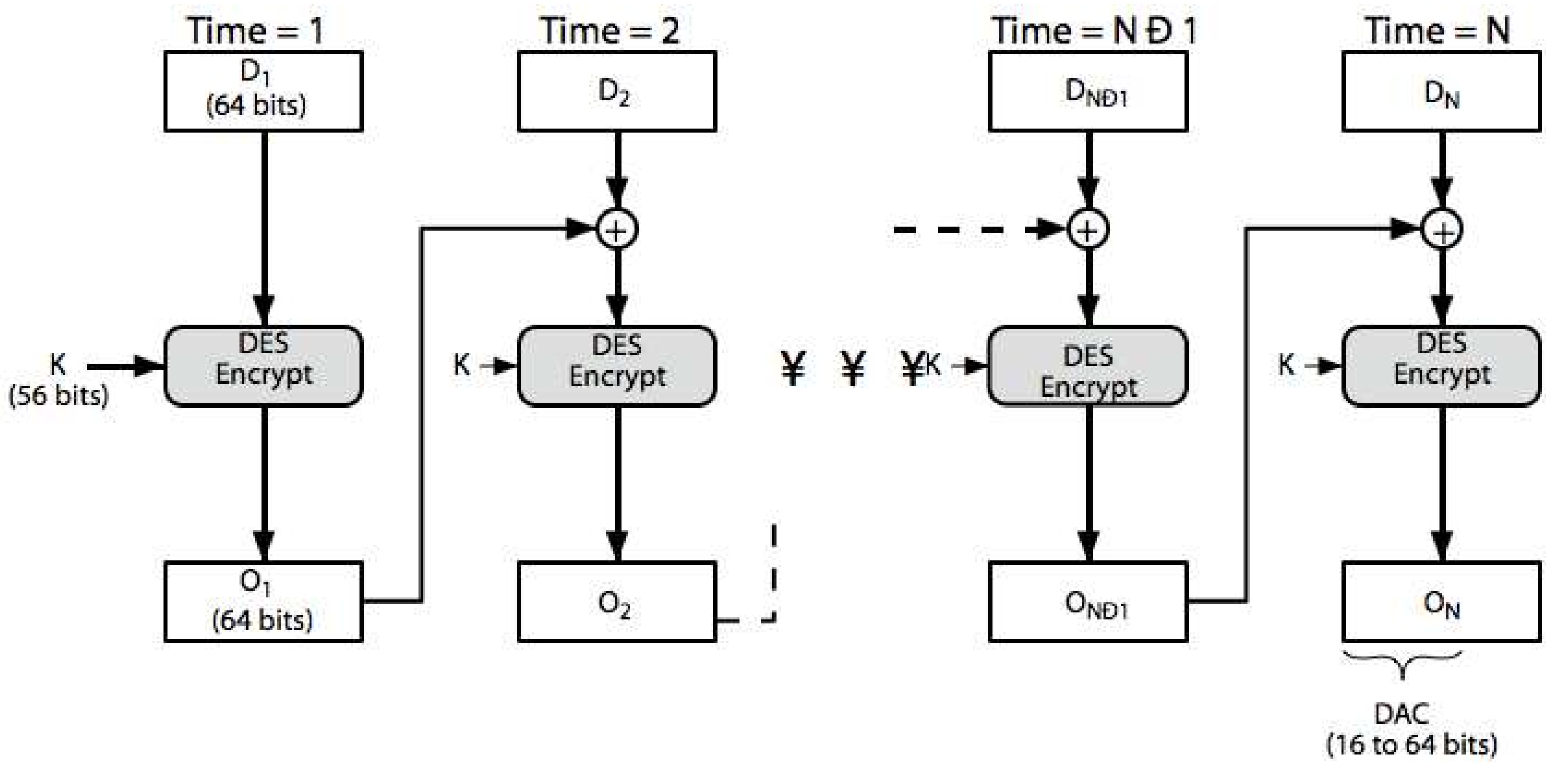
- MAC doesn’t provide a digital signature, because both the sender & receiver share the same key.
- MAC is also called as cryptographic checksum, which is equal to c(k,m), which condenses a variable length message M using a secret key k to produce a fixed length authenticator.
- MAC is called as a many to one function, because potentially many messages have same MAC.
- For a hundred bit message, 20 bit MAC is enough.

Requirements of MAC

- Taking into account about the types of message, the MAC needs to satisfy the following:

1. Knowing M, MAC pair, it is infeasible to find another message with same MAC. It is called as computation resistance property.
2. MAC should be uniformly distributed.
3. MAC should depend equally on all bits of the message

Data Authentication Algorithm



- It can use any block cipher chaining mode and use final block as a MAC
- **Data Authentication Algorithm (DAA)** is a widely used MAC based on DES-CBC
  - using IV=0 and zero-pad of final block
  - encrypt message using DES in CBC mode
  - and send just the final block as the MAC
    - or the leftmost M bits ($16 \leq M \leq 64$) of final block
- but final MAC is now too small for security

Hash Functions

- It condenses arbitrary message to fixed size

  $h = H(M)$

- It is usually assume that the hash function is public and not keyed.
- Hash function is used to detect changes to message
- It can be used in various ways with message
- It is most often used to create a digital signature

Requirements for Hash Functions

1. It can be applied to any sized message M
2. It produces fixed-length output h
3. It should be easy to compute h=H(M) for any message M
4. One way property:

   For a given hash code h,it is infeasible to find the message M such that H(M)=h
5. Week collision resistance:

   For given message M, it is infeasible to find N such that H(M)=H(M)
6. Strong collision resistance:

   It is infeasible to find any M,N such that H(N)=H(M)

Simple Hash Functions

- There are several proposals for simple functions based on XOR of message blocks.
- Message M is divided into L blocks of n bits.
- If the M length message is not the exact multiple of n bits, then padding the message with stuffing bit at end of message M to make the length as multiple of n.
- Simple Hash function uses bit by bit XOR operation on every block for generating Hash code.

$$H_i = M_{i1} \oplus M_{i2} \oplus M_{i3} \ldots\ldots \oplus M_{iL}$$

where H_i is the ith bit of hash code

L is the number of n bit blocks

Mij is the the ith bit in jth block

Simple Hash Functions

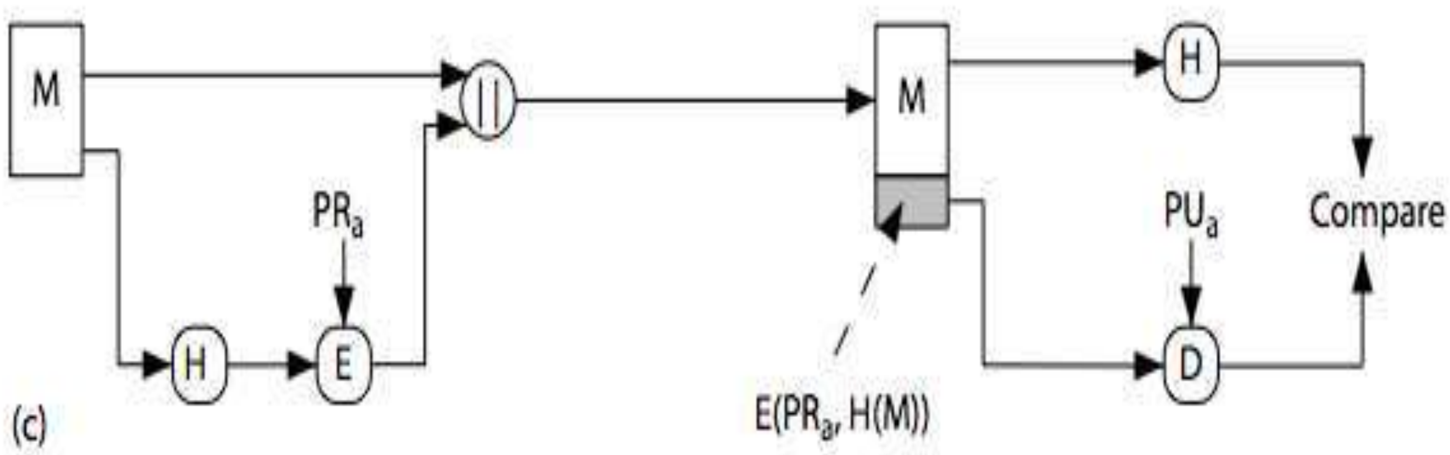
	Bit1	Bit2	…………..	Bit n
Block 1	M11	M21	……………	Mn1
Block 2	M12	M22	……………	Mn2
Block 3	M13	M23	……………	Mn3
Block L	M1L	M2L	……………	MnL
Hash Code(h)	h1	h2		hn

Simple Hash Functions

- Hash code is for each expression is as follows
  $h1=M11 \oplus M12 \oplus M13 \oplus \ldots\ldots \oplus M1L$
  $h2=M21 \oplus M22 \oplus M23 \oplus \ldots\ldots \oplus M2L$

  $hn=Mn1 \oplus Mn2 \oplus Mn3 \oplus \ldots\ldots \oplus MnL$
- The resulting hash value is encrypted with senders private key and is appended to the message M.
- At the receiving side, receiver computes the new hash value with the received message and compares with received hash value after decrypting the encrypted hash value using senders public key

Simple Hash Functions



Security of Hash Functions and Macs

- Just as with symmetric and public-key encryption, we can group attacks on hash functions and MACs into two categories:
  - Brute-force attacks
  - cryptanalysis.
- **Brute-Force Attacks-**
- The process trying all possible key one by one and check the resulting plaintext is meaningful.
- **Cryptanalysis-**
- It depends on the nature of algorithm and knowledge of general characteristics of plaintext.

Security of Hash Functions and Macs

- **Hash Functions**:-
- The strength of a hash function against brute-force attacks depends on the length of the hash code produced by the algorithm.
- There are three desirable properties:
- One-way:
  - For any given code h, it is computationally infeasible to find x such that $H(x) = h$.
- Weak collision resistance:
  - For any given block x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.
- Strong collision resistance:
  - It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.

Security of Hash Functions and Macs

Message Authentication Codes:

- A brute-force attack on a MAC is a more difficult in undertaking because it requires known message-MAC pairs.
- To attack a hash code, we can proceed in the following way.
- Given a fixed message x with n-bit hash code h = H(x), a brute-force method of finding a collision is to pick a random bit string y and check if H(y) = H(x). The attacker can do this repeatedly off line.

Security of Hash Functions and Macs

- we need to state the desired security property of a MAC algorithm, which can be expressed as follows:
- Computation resistance:
  - Given one or more text-MAC pairs $(x_i, C_K[x_i])$, it is computationally infeasible to compute any text-MAC pair $(x, C_K(x))$ for any new input $x \neq x_i$.

Cryptanalysis:

- It depends on the nature of algorithm and knowledge of general characteristics of plaintext.
- There are much more variety in the structure of MAC than Hash function, so it is difficult to generalize about cryptanalysis of MAC.

Secure Hash Algorithm

- SHA originally designed by National Institute of Standard and Technology(NIST) & National Security Agency(NSA) in 1993
- It was revised in 1995 as SHA-1
- It is based on design of MD5 with key differences
- It takes as input a message with a maximum length of 2^64 bits and produces a output of 160-bit Message digest.

SHA Overview

1. ***Padding:*** Length of the message is 64 bits short of multiple of 512 after padding.
2. ***Append*** a 64-bit ***length*** value of original message is taken.
3. ***Divide the input into 512-bit blocks***
4. ***Initialise Constant Value(CV)*** 160-bit buffer (A,B,C,D,E) to

(*A*=01 23 45 67,

B=89 AB CD EF,

C=FE DC BA 98,

D=76 54 32 10,

E=C3 D2 E1 F0)

Continue…

5. ***Process*** ***Blocks*** now the actual algorithm begins.Message in 16-word (512-bit) chunks:
   - Copy 160 bit buffer(A,B,C,D,E) into single register for storing temporary intermediate as well as the final results.
   - Divide the current 512-bit blocks into 16 sub-blocks, each consisting of 32 bits.
   - It has 4 Rounds of processing, each round consisting of 20 ***step iteration*** operations on message block & buffer
   - Expand 16 words into 80 words(20*4) by mixing & shifting.K[t] constant= ***Where t=0 to 79***
   - Form new buffer value by adding output to input.

6. output hash value is the final buffer value

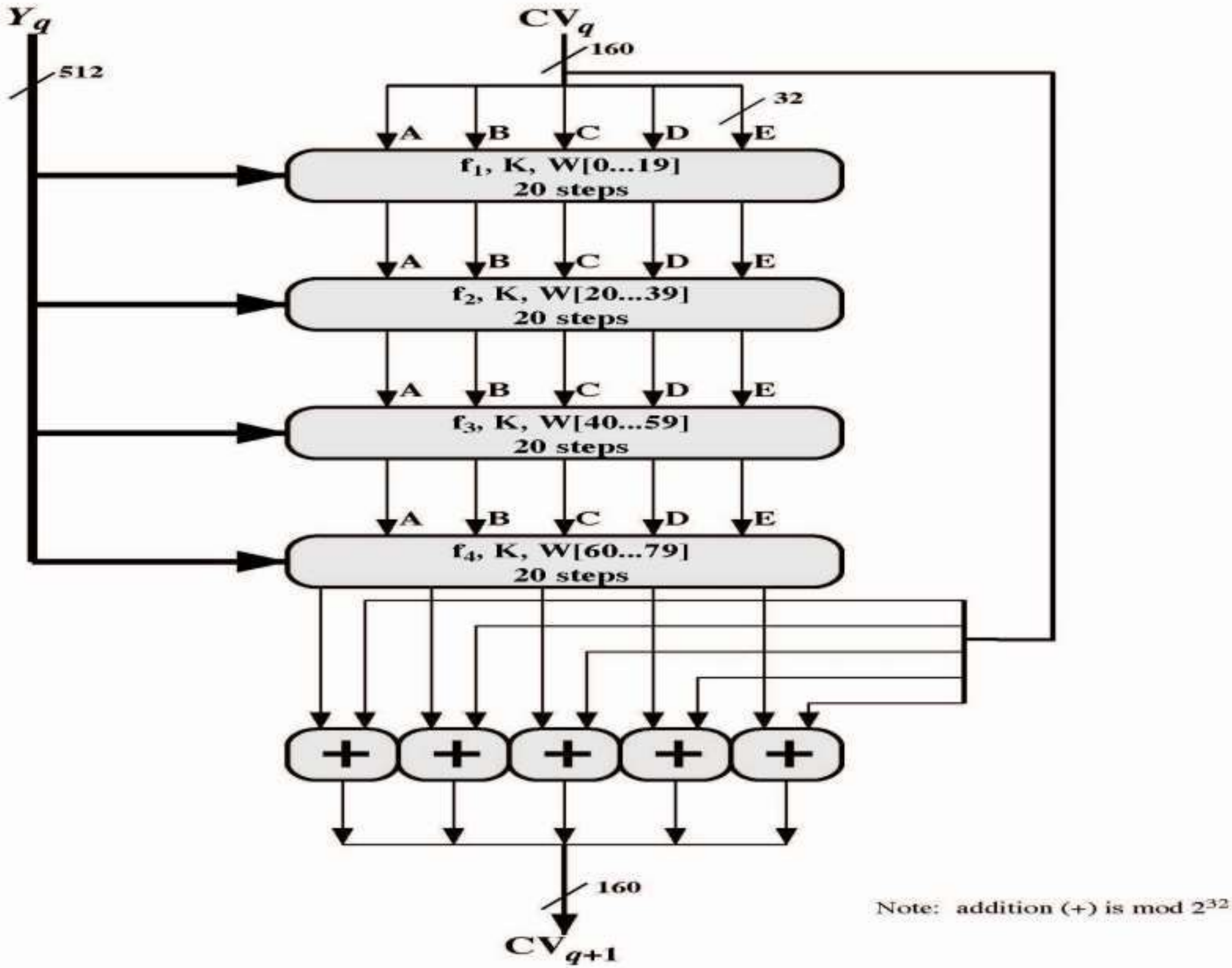
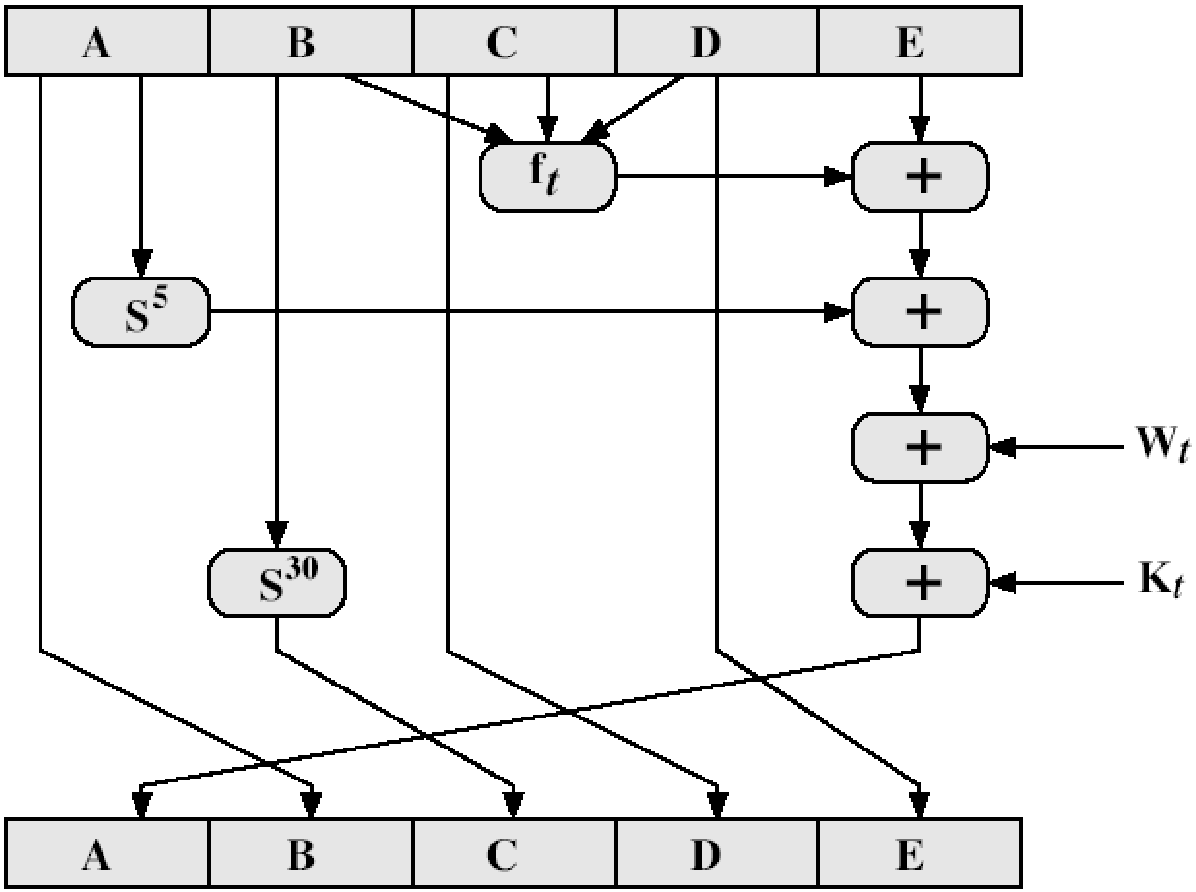


Figure 12.5 SHA-1 Processing of a Single 512-bit Block (SHA-1 Compression Function)

SHA-1 Compression Function



ABCDE=(F[t]+E+S5(A)+W[t]+K[t]),>>>Shift right by 1 bit for next iteration

SHA-1 Compression Function terms

- each round has 20 steps which replaces the 5 buffer words thus:

  $(A,B,C,D,E) <-$
  $(E+f(t,B,C,D)+(A<<5)+W_t+K_t),A,(B<<30),C,D)$
- ABCDE refer to the 5 words of the buffer
- t is the step number
- f(t,B,C,D) is nonlinear function for round
- W_t is derived from the message block
- K_t is a constant value
- S^t circular left shift of 32 bit sub-block by t bits

Process F(t) in each SHA-1 round

- where g can be expressed as:

ROUND 1: (b AND c) OR ((NOT b) AND (d))

ROUND 2: b XOR c XOR d

ROUND 3: (b AND c) OR (b AND d) OR (c AND d)

ROUND 4: b XOR c XOR d

Creation of 80-word input W_t

- Adds redundancy and interdependence among message blocks

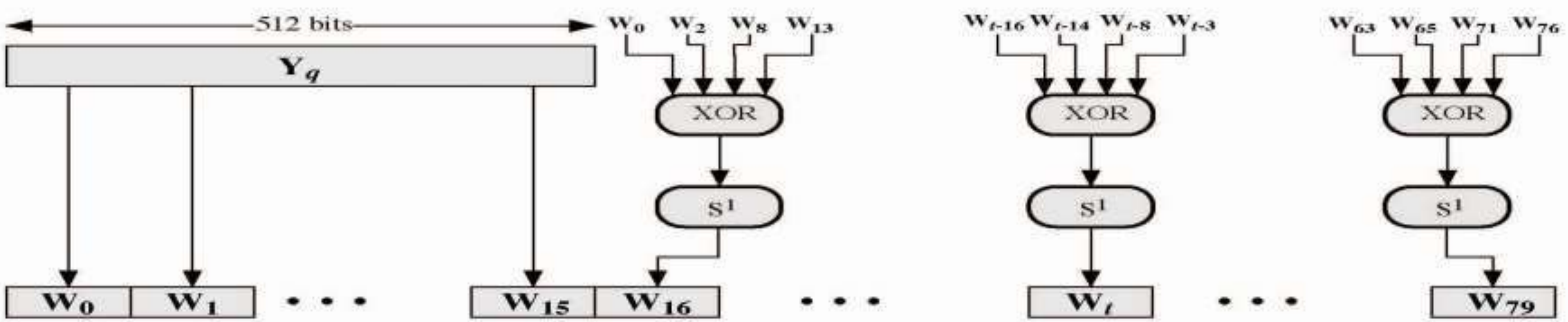


Figure 12.7 Creation of 80-word Input Sequence for SHA-1 Processing of Single Block

Kt Constant Value

5a827999 $0 \leq t \leq 19$
6ed9eba1 $20 \leq t \leq 39$
8 f 1bbcdc $40 \leq t \leq 59$
ca62c1d6'' $50 \leq t \leq 79$

Revised Secure Hash Standard

- NIST issued revision in 2002 by adding 3 additional versions of SHA
  - SHA-256, SHA-384, SHA-512
  - Different lengths of Message Digest in bits
- It is designed for compatibility to provide increased security.
- structure & detail is similar to SHA-1
- hence analysis should be similar
- but security levels are rather higher.

Revised Secure Hash Standard

Table 12.3 Comparison of SHA Properties

	SHA-1	SHA-256	SHA-384	SHA-512
Message digest size	160	256	384	512
Message size	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block size	512	512	1024	1024
Word size	32	32	64	64
Number of steps	80	80	80	80

HMAC

- A hash function such as SHA was not designed for use as a MAC and cannot be used directly for that purpose because it does not rely on a secret key.
- There have been a number of proposals for the incorporation of a secret key into an existing hash algorithm.
- The approach that has received the most support is HMAC.

HMAC

The following are the design objectives for HMAC:-

- To use, without any modifications available in hash functions.
- To allow for easy replace ability of the embedded hash function in case of faster or more secure hash functions are required.
- To preserve the original performance of the hash function without incurring a significant degradation.
- To use and handle keys in a simple way.
- To have a well understood cryptographic analysis of the strength of the authentication mechanism

HMAC Algorithm

H = embedded hash function (e.g., MD5, SHA-1, RIPEMD-160)

IV = initial value input to hash function

M = message input to HMAC(including the padding specified in the embedded hash function)

Y_i = ith block of M, $0 \leq i \leq (L - 1)$

L = number of blocks in M

b = number of bits in a block

n = length of hash code produced by embedded hash function

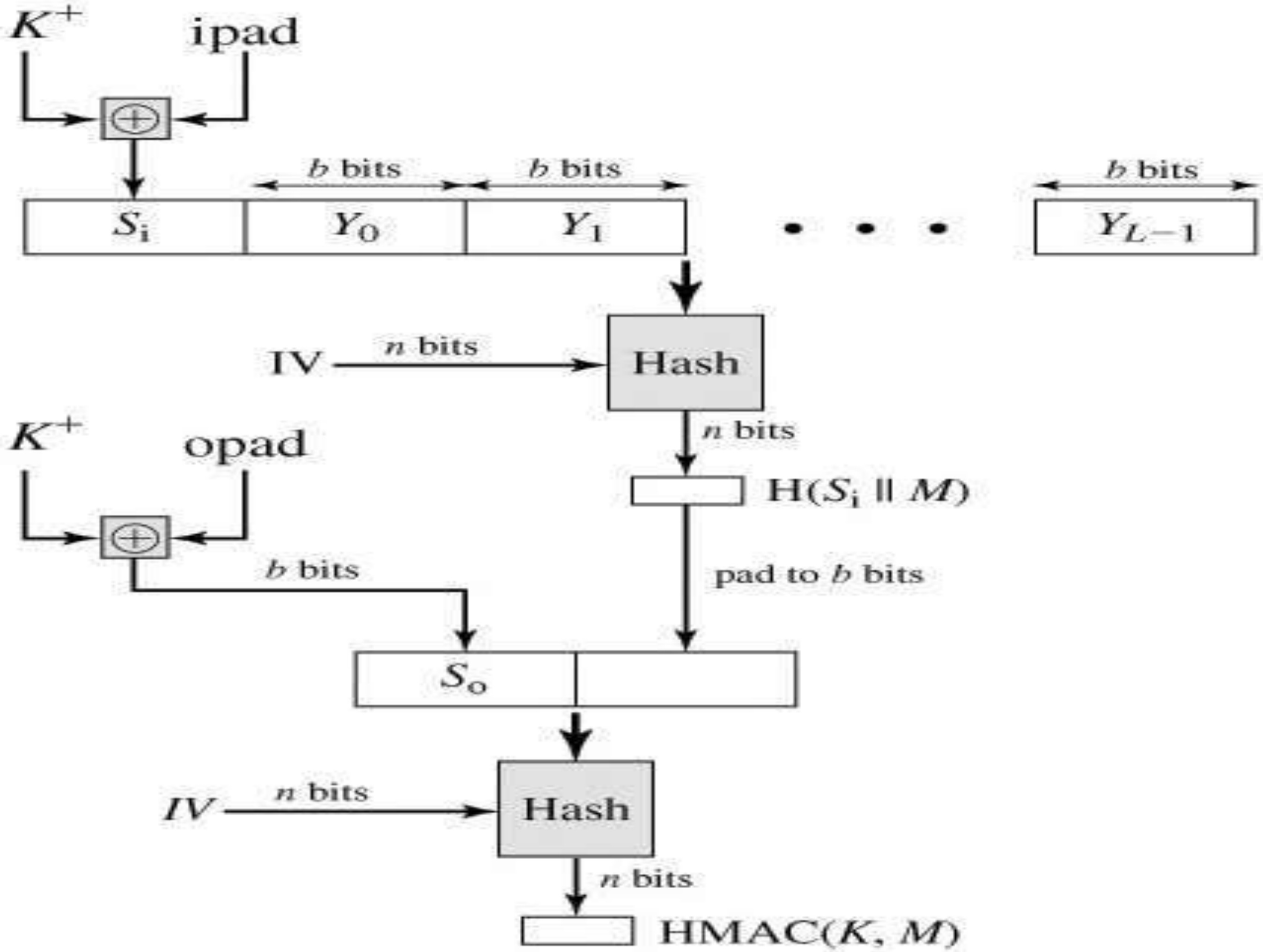
K = secret key recommended length is $\geq n$; if key length is greater than b; the key is input to the hash function to produce an n-bit key

K^+ = K padded with zeros on the left so that the result is b bits in length

ipad = 00110110 (36 in hexadecimal) repeated $b/8$ times

opad = 01011100 (5C in hexadecimal) repeated $b/8$ times

HMAC Algorithm



HMAC Algorithm

Then HMAC can be expressed as follows:

$HMAC(K, M) = H[(K \oplus opad)||H[(K \oplus ipad)||M]]$

In words,

1. Append zeros to the left end of K to create a b-bit string K (e.g., if K is of length 160 bits and b = 512 then K will be appended with 44 zero bytes 0 x 00).
2. XOR (bitwise exclusive-OR) K with ipad to produce the b-bit block S_i.
3. Append M to S_i.
4. Apply H to the stream generated in step 3.
5. XOR K with opad to produce the b-bit block S_o
6. Append the hash result from step 4 to S_o
7. Apply H to the stream generated in step 6 and output the result.

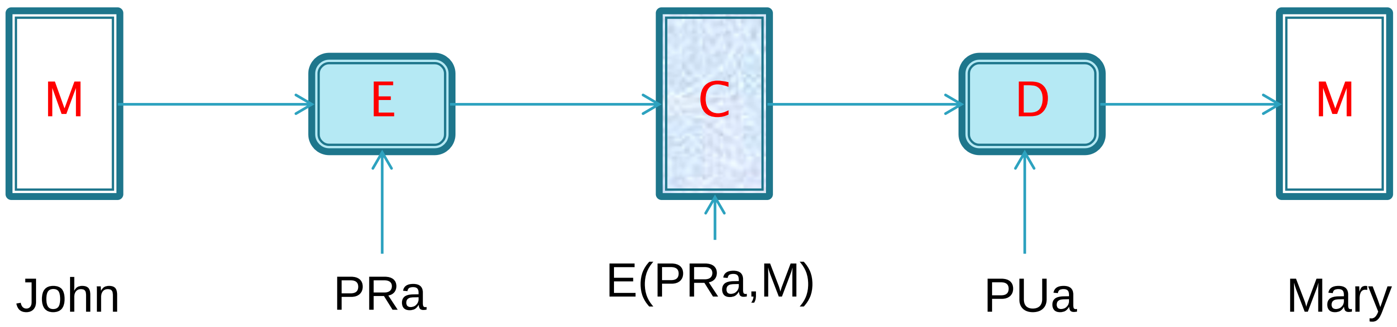
Digital Signature

Digital Signatures

- ✓ A digital signature is an **authentication mechanism** that enables the creator of a message **to attach a code** that acts as a signature.
- ✓ Typically the signature is formed by taking the **hash of the message and encrypting the message** with the creator's private key.
- ✓ The signature guarantees the source and **integrity** of the message.
- ✓ The digital signature standard (DSS) is an NIST standard that uses the **secure hash algorithm (SHA).**

Properties

- ✓ Message authentication protects two parties who exchange messages from any third party. However, it does not protect the two parties against each other. Several forms of dispute between the two are possible.



- ✓ For example, suppose that John sends an authenticated message to Mary, using one of the schemes of Figure. Consider the following disputes that could arise.
  - ✓ Mary may forge a different message and claim that it came from John. Mary would simply have to create a message and append an authentication code using the key that John and Mary share.
  - ✓ John can deny sending the message. Because it is possible for Mary to forge a message, there is no way to prove that John did in fact send the message.

Properties

- ✓ In situations where there is not complete trust between sender and receiver, something more than authentication is needed.
- ✓ The most attractive solution to this problem is the digital signature.
- ✓ The digital signature must have the following properties:
  - ✓ **It must verify the author and the date and time of the signature.**
  - ✓ **It must authenticate the contents at the time of the signature.**
  - ✓ **It must be verifiable by third parties, to resolve disputes.**
- ✓ Thus, the digital signature function includes the authentication function.

Requirements

- ✓ The signature must be a bit pattern that depends on the message being signed.
- ✓ The signature must use some information unique to the sender to prevent both forgery and denial.
- ✓ It must be relatively easy to produce the digital signature.
- ✓ It must be relatively easy to recognize and verify the digital signature.
- ✓ It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
- ✓ It must be practical to retain a copy of the digital signature in storage.

Drawbacks of using digital signature

- ✓ The process of generation and verification of digital signature requires considerable amount of time. So, for frequent exchange of messages the speed of communication will reduce.

- ✓ When the digital signature is not verified by the public key, then the receiver simply marks the message as invalid but he does not know whether the message was corrupted or the false private key was used.

- ✓ Although digital signature provides authenticity, it does not ensure secrecy of the data. To provide the secrecy, some other technique such as encryption and decryption needs to be used.

Approaches for the digital signature function

- ✓ **A variety of approaches has been proposed for the digital signature function.**
- ✓ **These approaches fall into two categories:**
  - ✓ **Direct Digital Signature X → Y**
  - ✓ **Arbitrated Digital Signature X→ A → Y**

Direct Digital Signature X → Y

- ✓ involve only sender & receiver
- ✓ assumed receiver has sender's public-key
- ✓ digital signature made by sender signing entire message or hash with private-key
- ✓ can encrypt using receivers public-key for confidentiality
- ✓ important thing is that, sign first then encrypt message & signature
- ✓ security depends on sender's private-key

Arbitrated Digital Signatures

- involves use of arbiter A
  - validates any signed message
  - then dated and sent to recipient
- requires suitable level of trust in arbiter
- can be implemented with either private or public-key algorithms
- arbiter may or may not see message

Arbitrated Digital Signatures

$$(1)\ X \rightarrow A:\ ID_X \,\|\, E(PR_x, [ID_X \,\|\, E(PU_y, E(PR_x, M))])$$
$$(2)\ A \rightarrow Y:\ E(PR_a, [ID_X \,\|\, E(PU_y, E(PR_x, M)) \,\|\, T])$$

(c) Public-Key Encryption, Arbiter Does Not See Message

Notations:

X=sender
Y=recipient
A=Arbiter
ID_X=ID of X

M=message
T=time stamp
PR_X=X's private key
PU_Y=Y's public key
PR_A=A's private key

Weakness: twice public-key encryptions on the message

Digital Signature Standards (DSS)

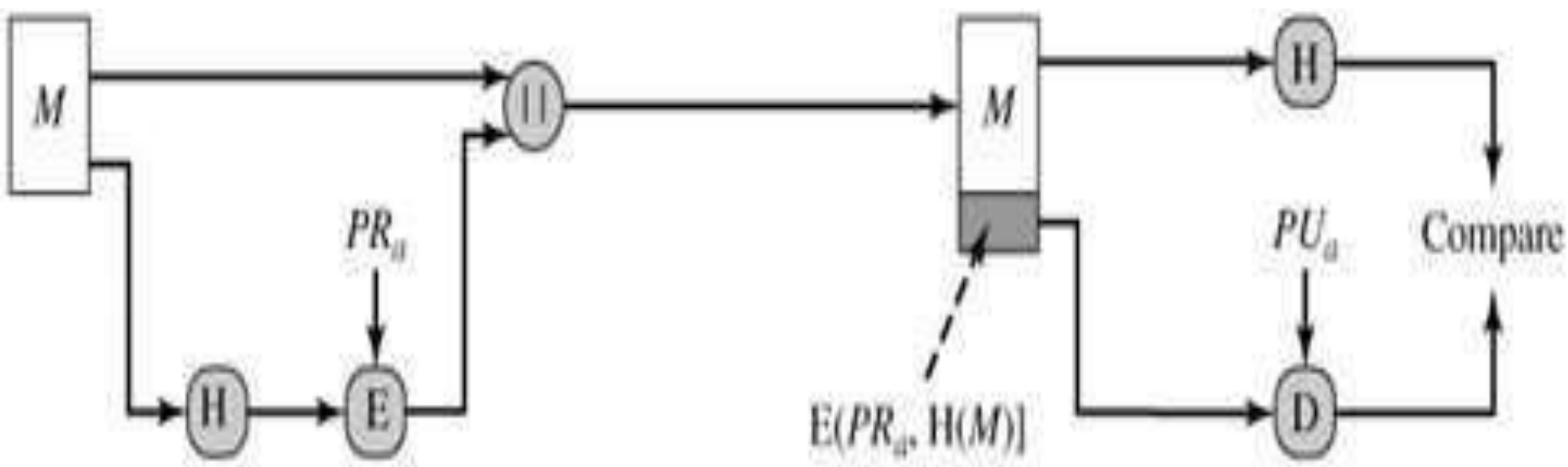
History:

- ✓ Designed by NIST & NSA in early 90's
- ✓ uses the SHA hash algorithm
- ✓ DSS is the standard, DSA is the algorithm
- ✓ Creates a 320 bit signature, but with 512-1024 bit security
- ✓ Security depends on difficulty of computing discrete logarithms

Two Approaches to Digital Signatures

1. **RSA approach**
2. **DSS Approach**

1. RSA approach

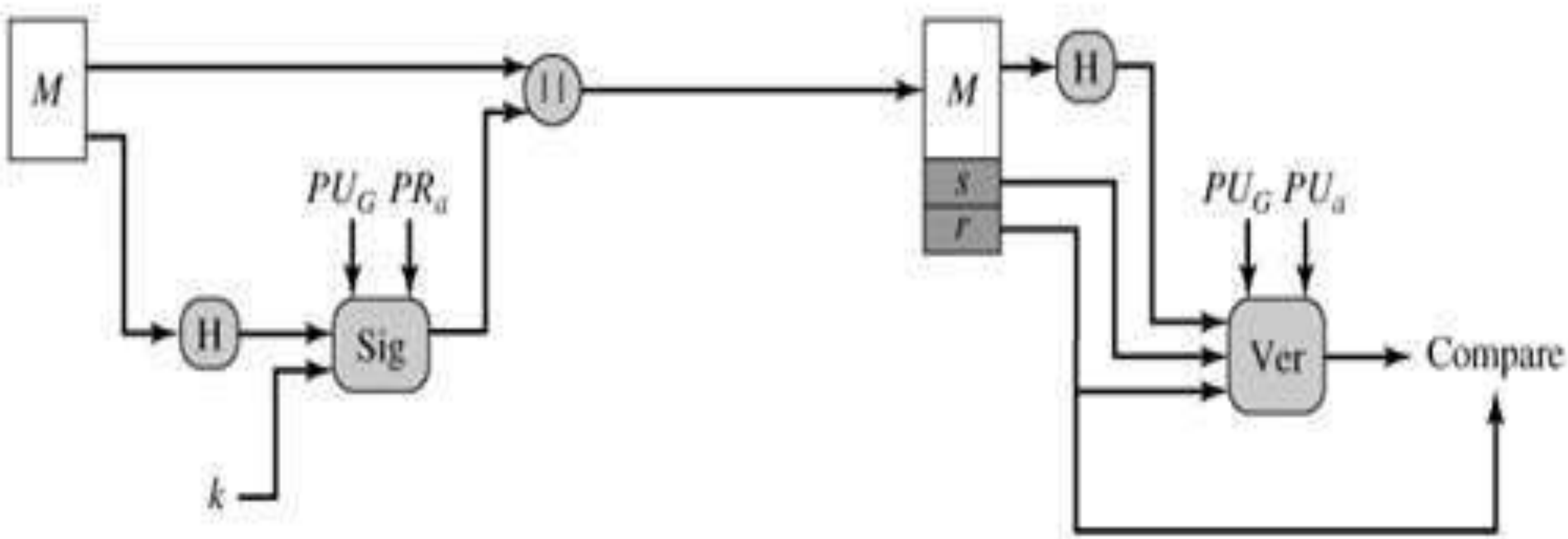


Two Approaches to Digital Signatures

- In RSA approach the message to be signed is the input to hash function that produces secure hash code of fixed length.
- The hash code is then encrypted using senders private key to form the signature.
- Both the message and signature are then transmitted.
- The recipient takes the message and produces a hash code and also decrypts the signature using senders public key.
- If the calculated hash code matches the decrypted signature, then the signature is accepted as valid

Two Approaches to Digital Signatures

2. DSS Approach



Two Approaches to Digital Signatures

- In this approach the message M is given to the hash function for producing fixed length hash code.
- The hash code is given as input to the signature function.
- It also takes random number(K), senders private key(PRa) and global public key(PUg) as the inputs and produces signature as the output.
- The signature consist of 2 components r and s.
- At the destination, the receiver computes the hash code on the received message.

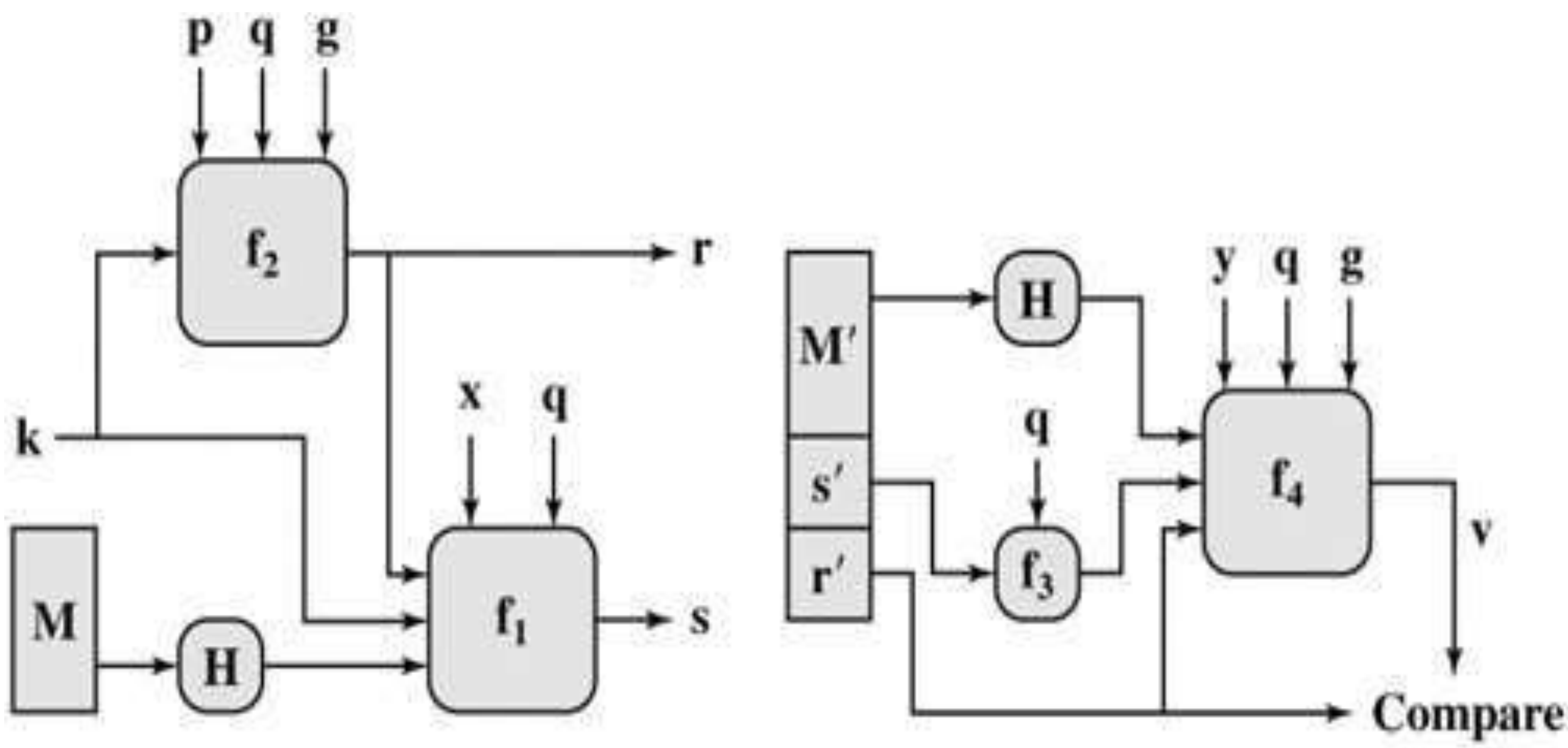
Two Approaches to Digital Signatures

- The Computed hash code and signature are given as input for the verification block.
- The verification function takes senders public key(PUa) and global public key(PUg) as input and produces signature component as output.
- If the produced signature component is equal to the received signature component r, then the signature is valid.

Digital Signature Algorithm (DSA)

// Global Public-Key Components	
p	prime number where $2^{L-1} < p < 2^{L}$ for $512 \leq L \leq 1024$ and L a multiple of 64; i.e., bit length of between 512 and 1024 bits in increments of 64 bits
q	prime divisor of (p-1), where $2^{159} < q < 2$; i.e., bit length of 160 bits
g	$g = h^{(p-1)/q} \bmod p$, where h is any integer with $1 < h < (p-1)$ such that $h^{(p-1)/q} \bmod p > 1$
// User's Private Key	
x	random or pseudo random integer with $0 < x < q$
// User's Public Key	
y	$= g^{x} \bmod p$
// User's Per-Message Secret Number	
k	random or pseudo random integer with $0 < k < q$
// Signing	
r	$= (g^{k} \bmod p) \bmod q$
s	$= [k^{-1} (H(M) + xr)] \bmod q$
	Signature = (r, s)
// Verifying	
w	$= (s')^{-1} \bmod q$
u1	$= [H(M')w] \bmod q$
u2	$=(r')w \bmod q$
v	$= [(g^{u1} y^{u2}) \bmod p] \bmod q$
// Test :	**v = r'**

DSS Signing and Verifying



$s = f_1(H(M), k, x, r, q) = (k^{-1}(H(M) + xr)) \bmod q$

$r = f_2(k, p, q, g) = (g^k \bmod p) \bmod q$

(a) Signing

$w = f_3(s', q) = (s')^{-1} \bmod q$

$v = f_4(y, q, g, H(M'), w, r')$

$= ((g^{(H(M')w) \bmod q}\, y^{r'w \bmod q}) \bmod p) \bmod q$

(b) Verifying

Authentication Applications

Authentication Applications

- Applications will consider authentication functions
- It is developed to support application–level authentication & digital signatures
- It will consider Kerberos – a private–key authentication service and then X.509 – a public–key directory authentication service

Kerberos

- Authentication service -Distributed Network(DN)
- Trusted Third party authentication-establish client server communication
- It provides centralised private-key and third-party authentication in a distributed network
- Threats in DN:

1. User gain access to workstation and pretend to be another user
2. Alters the network address and sends request appears to come from authenticated user.
3. Access the message and use replay attack

- two versions in use: 4 & 5

Kerberos Motivation

In Distributed Network, workstation and server requires 3 approaches of security

1. Client workstation–Assure their identity of its user to server
2. Client system–authenticate themselves to server
3. Client user– prove their identity for each services invoked and server must prove their identity to clients

Kerberos Requirements

- its identified requirements are:
  - **Secure:**- Strong enough that opponent does not able find weak link
  - **Reliable:**- Distributed server architecture with one system able to backup with another
  - **Transparent:**- User should not be aware that authentication is taking place
  - **Scalable:**- Support large number of clients and servers
- implemented using an authentication protocol based on Needham-Schroeder

Kerberos version 4
A Simple authentication scheme

- In Unprotected n/w, any client can apply to any server
- Opponent –pretend to be another client –to obtain unauthorized access to server
- To counter this threat, Server must confirm the identity of client becomes heavy burden to server.
- An alternative is to use authentication server(AS)–knows the passwords of all users stored in database.

A Simple authentication scheme

- Consider the following dialogues

1. $C\text{->}AS : ID_c \mid\mid PWD_c \mid\mid ID_s$
2. $AS\text{->}C : Ticket=E_{ks}[ID_c \mid\mid Netadd_c \mid\mid ID_s]$
3. $C\text{->}S : ID_c \mid\mid Ticket$

Drawbacks:

1. Password of client C – sent to AS without any encryption, so intruder can access it and use it.
2. For each transaction needs ticket

More Secured authentication scheme

- a basic third-party authentication scheme
- have an Authentication Server (AS)
  - users initially negotiate with AS to identify self
  - AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
- have a Ticket Granting server (TGS)
  - users subsequently request access to other services from TGS on basis of users TGT

More Secured authentication scheme

1. obtain ticket granting ticket from AS
   - once per session

   1.C->AS: ID_c || ID_{TGS}

   2. AS->C: $E_{kc}(Ticket_{TGS})$
2. obtain service granting ticket from TGS
   - for each distinct service required

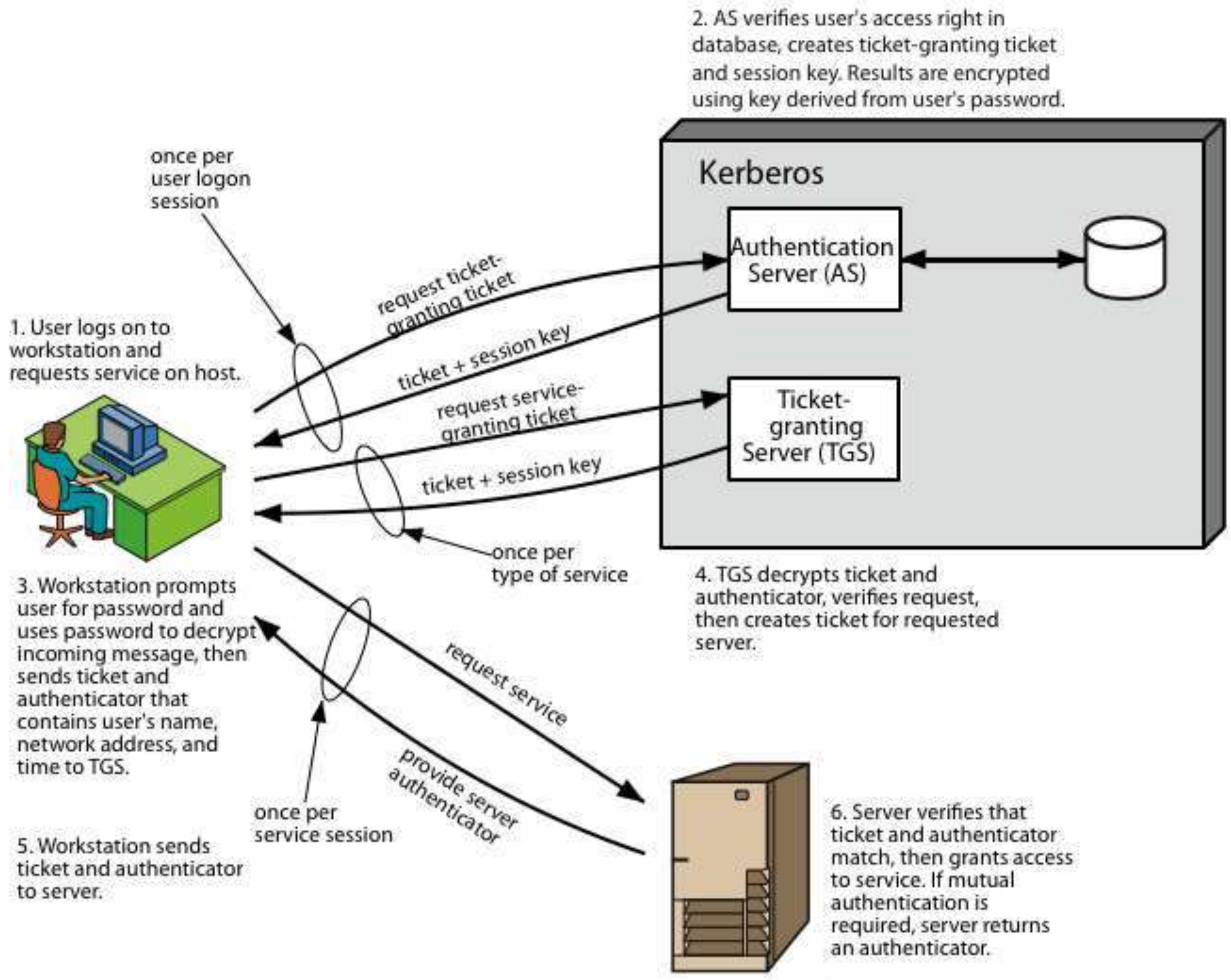
   3.C->TGS: ID_c || ID_s ||$Ticket_{TGS}$

   4.TGS->C: $Ticket_s$

3.Client/server exchange to obtain service
   - on every service request

   5. C->S: IDc || $Ticket_s$

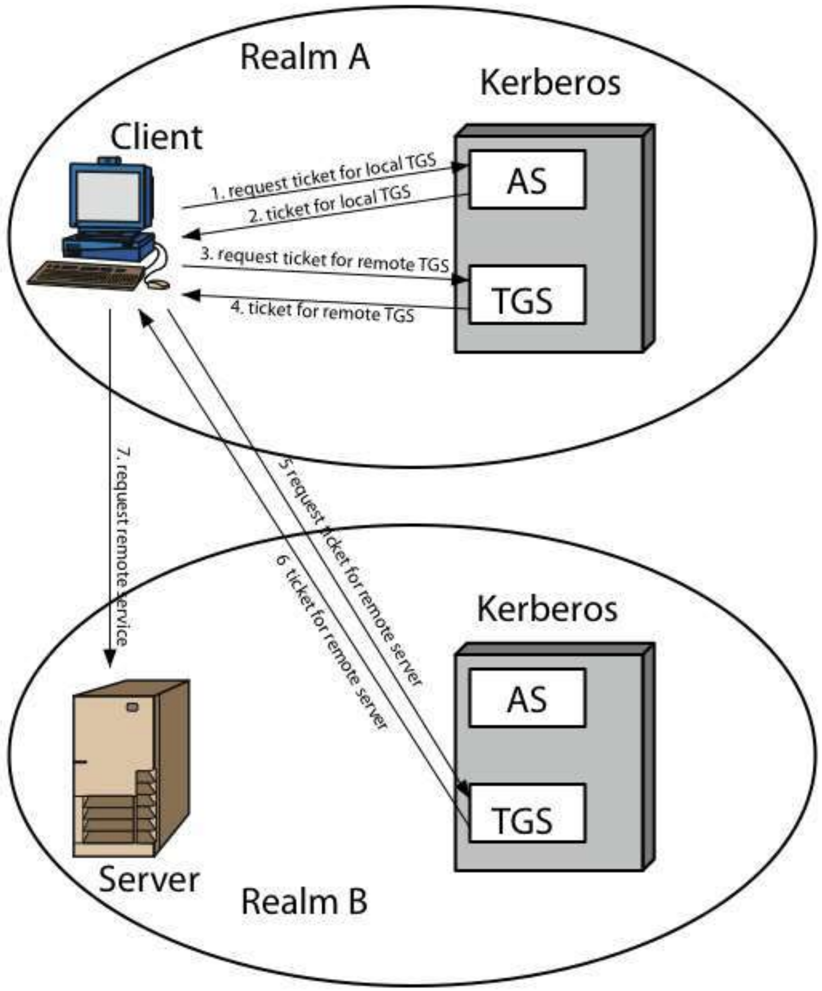
Kerberos 4 Overview



Kerberos Realms

- a Kerberos environment consists of:
  - a Kerberos server
  - a number of clients, all registered with server
  - application servers, sharing keys with server
- this is termed a realm
  - typically a single administrative domain
- if have multiple realms, their Kerberos servers must share keys and trust

Kerberos Realms



Kerberos Version 5

- developed in mid 1990's
- specified as Internet standard RFC 1510
- provides improvements over v4
  - addresses environmental shortcomings
    - encryption alg, network protocol, byte order, ticket lifetime, authentication forwarding, interrealm auth
  - and technical deficiencies
    - double encryption, non-std mode of use, session keys, password attacks

X.509 AUTHENTICATION SERVICE

X.509 Authentication Service

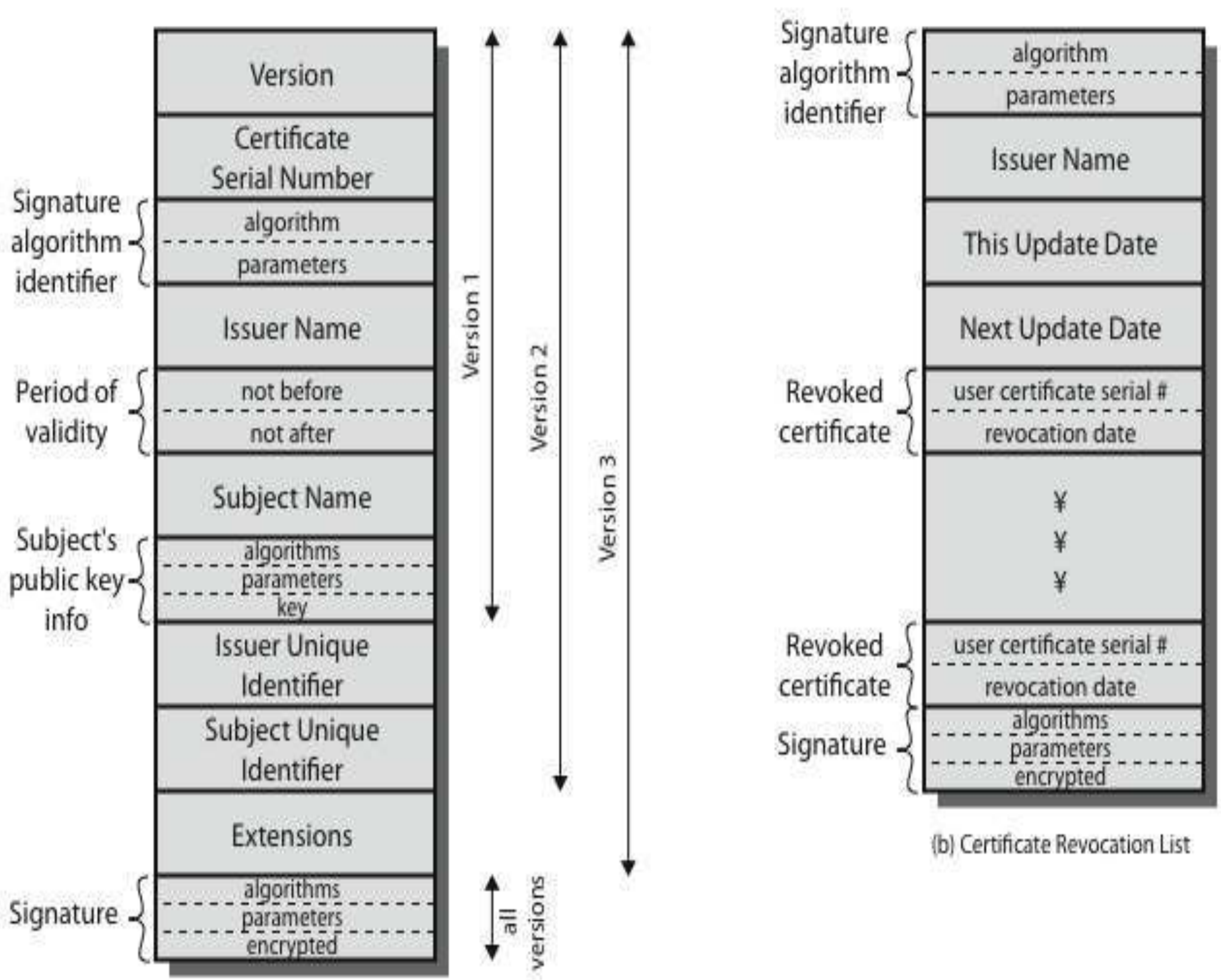
- It is distributed server maintaining user info database
- It defines framework for authentication services
  - The directory may store public-key certificates with public key of user signed by certification authority
- It uses public-key crypto & digital signatures
  - Algorithms are not standardised, but RSA is recommended
- X.509 certificates are widely used

  –VISA and MASTER CARDS for secured electronic transactions

X.509 Certificates

- It is issued by a Certification Authority (CA), containing:
  - version (1, 2, or 3)
  - serial number (unique within CA) identifying certificate
  - signature algorithm identifier
  - Issuer name (CA)
  - period of validity (from - to dates)
  - Subject name (name of owner)
  - subject public-key info (algorithm, parameters, key)
  - issuer unique identifier (v2+)
  - subject unique identifier (v2+)
  - extension fields (v3)
  - signature (of hash of all fields in certificate)
- It is notated as `CA<<A>>` that denotes certificate for A signed by CA

X.509 Certificates



(b) Certificate Revocation List

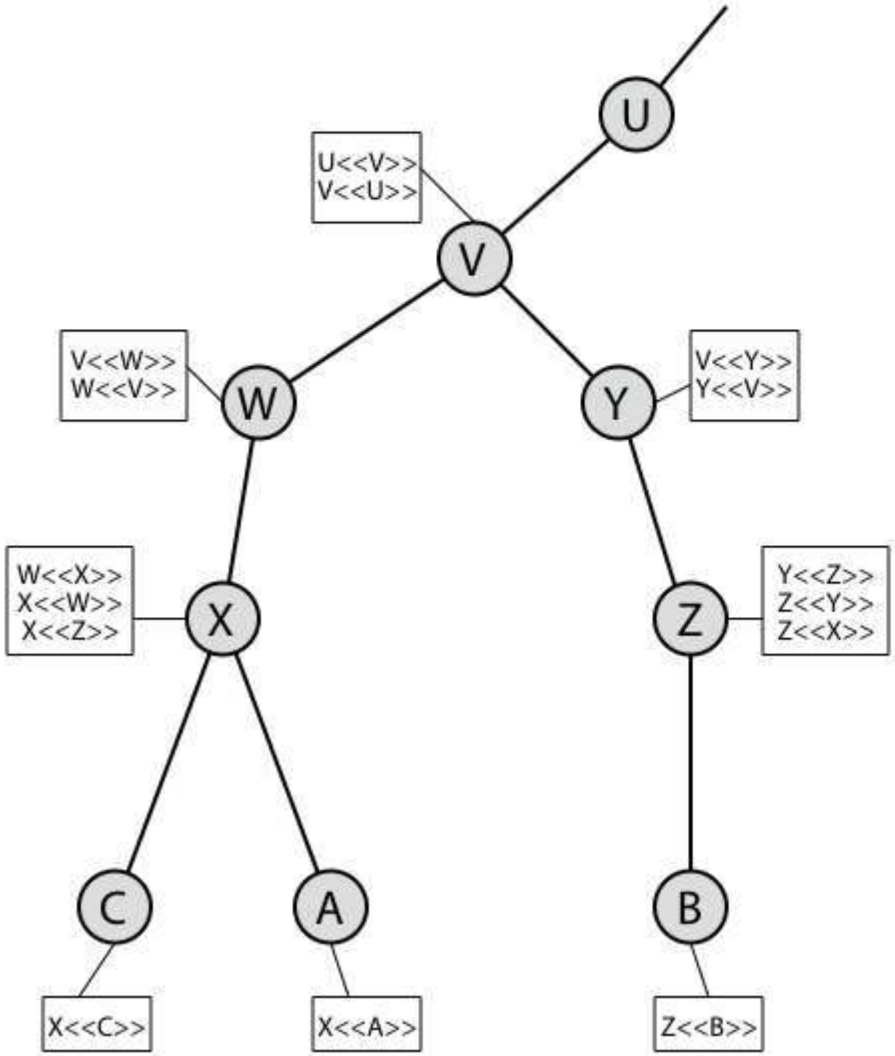
Obtaining a Certificate

- Any user with access to Certificate Authority(CA) can get any certificate from it.
- Only the CA can modify a certificate because it cannot be forged as certificates are 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
- We 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 and enable verification of any certificate from one CA by users of all other CAs in hierarchy

CA Hierarchy Use



Certificate Revocation

- Each certificates have a period of validity
- It may need to revoke before expiry,

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)
- Each users should check certificates with CA’s CRL

Authentication Procedures

- X.509 includes three alternative authentication procedures:
- One-Way Authentication
- Two-Way Authentication
- Three-Way Authentication

One-Way Authentication

- 1 message (A->B) used to establish
  - the identity of A and that message is from A
  - message was intended for B
  - integrity & originality of message
- message must include timestamp, nonce, B's identity and is signed by A
- may include additional info for B
  - eg session key

Two-Way Authentication

- 2 messages (A->B, B->A) which also establishes in addition:
  - the identity of B and that reply is from B
  - that reply is intended for A
  - integrity & originality of reply
- reply includes original nonce from A, also timestamp and nonce from B
- may include additional info for A

Three-Way Authentication

- 3 messages (A->B, B->A, A->B) which enables above authentication without synchronized clocks
- has reply from A back to B containing signed copy of nonce from B
- means that timestamps need not be checked or relied upon

Unit-4

Electronic Mail and IP Security

Electronic Mail and IP Security

- Pretty good privacy - S/MIME
- IPsec overview – architecture - Authentication Header and Encapsulating security pay load - combining security associates.

Pretty Good Privacy (PGP)

- Phil Zimmermann is the creator of PGP
- PGP provides confidentiality and authentication service used for electronic mail and file storage applications
- Available as free on variety of platforms
- Based on well known algorithms
- Wide range of applicability
- Not developed or controlled by governmental or standard organization

Operational Description

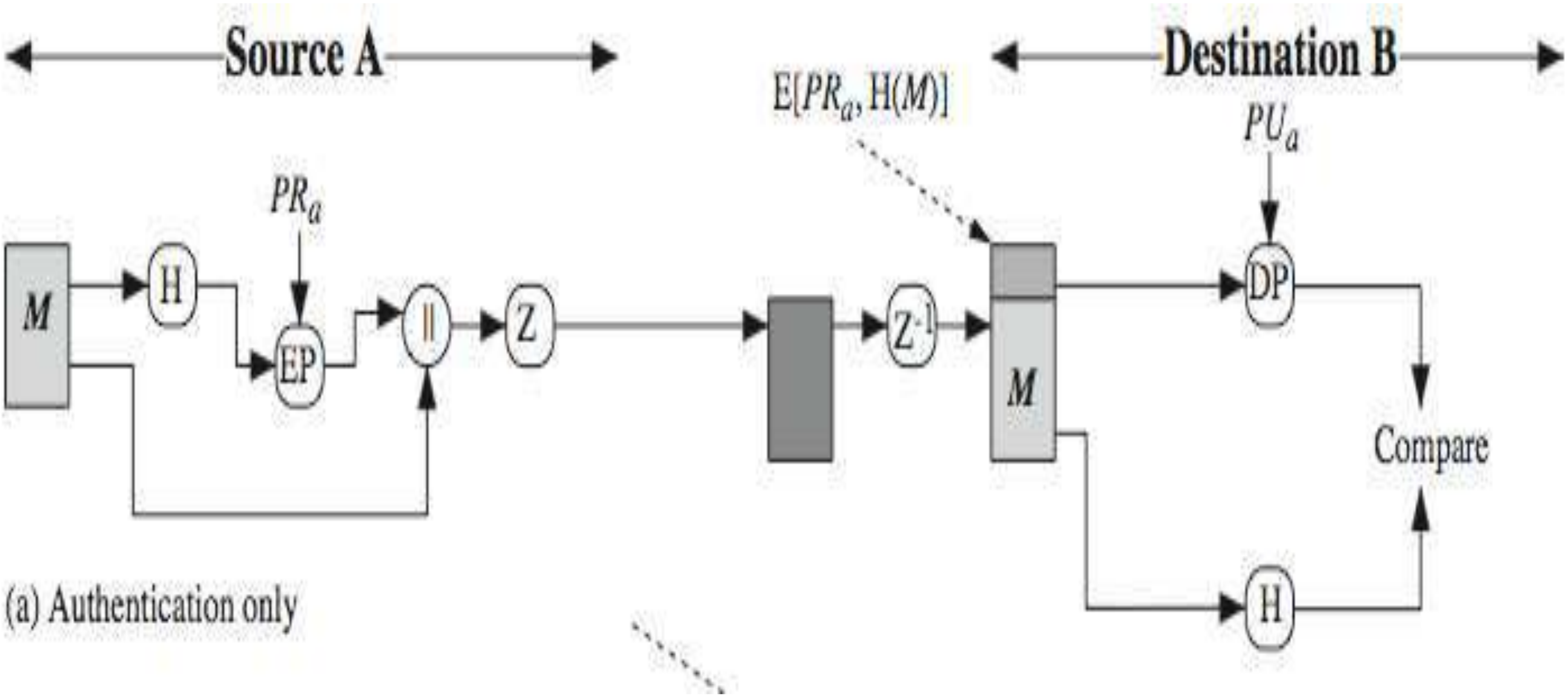
Consist of five services:

- ✓ Authentication
- ✓ Confidentiality
- ✓ Compression
- ✓ E-Mail compatibility
- ✓ Segmentation

PGP Operation – Authentication

1. sender creates message
2. use SHA-1 to generate 160-bit hash of message
3. signed hash with RSA using sender's private key, and is attached to message
4. receiver uses RSA with sender's public key to decrypt and recover hash code
5. receiver verifies received message using hash of it and compares with decrypted hash code

PGP Operation – Authentication

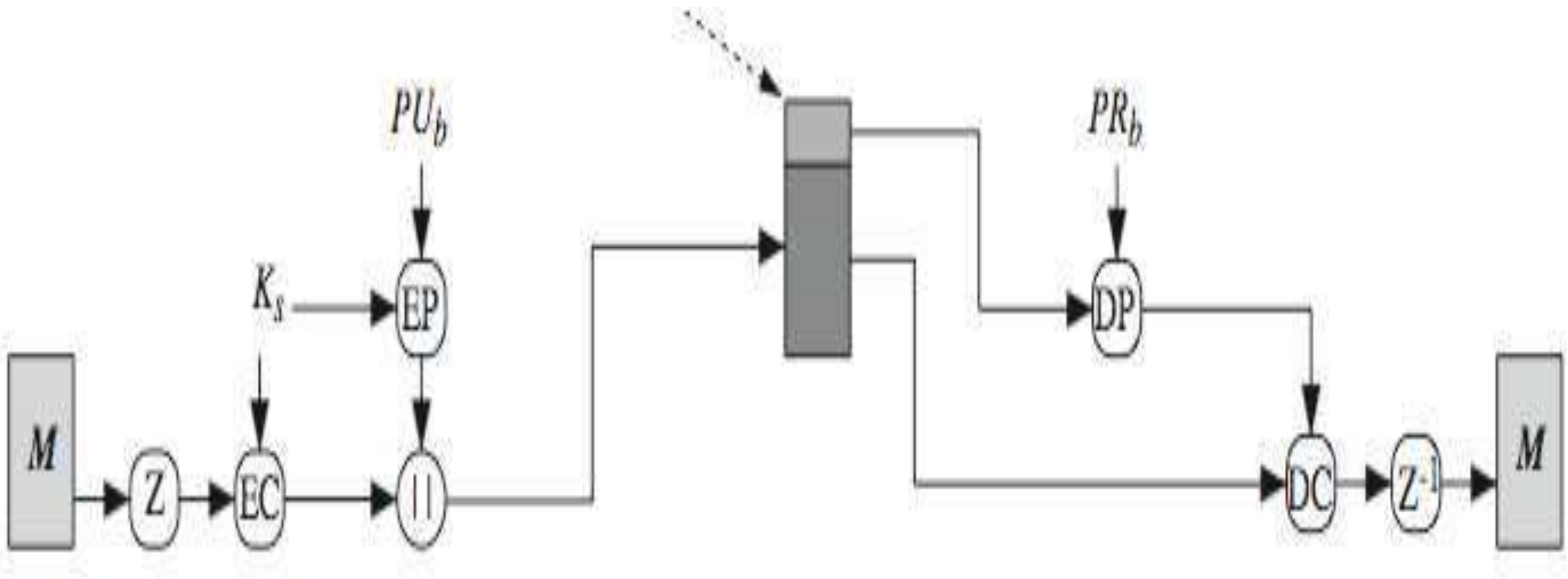


(a) Authentication only

PGP Operation – Confidentiality

1. sender generates message and encrypts with 128-bit random number as session key for it
2. encrypt message using CAST-128 / IDEA / 3DES in CBC mode with session key
3. session key encrypted using RSA with recipient's public key, & attached to msg
4. receiver uses RSA with private key to decrypt and recover session key
5. session key is used to decrypt message

PGP Operation – Confidentiality

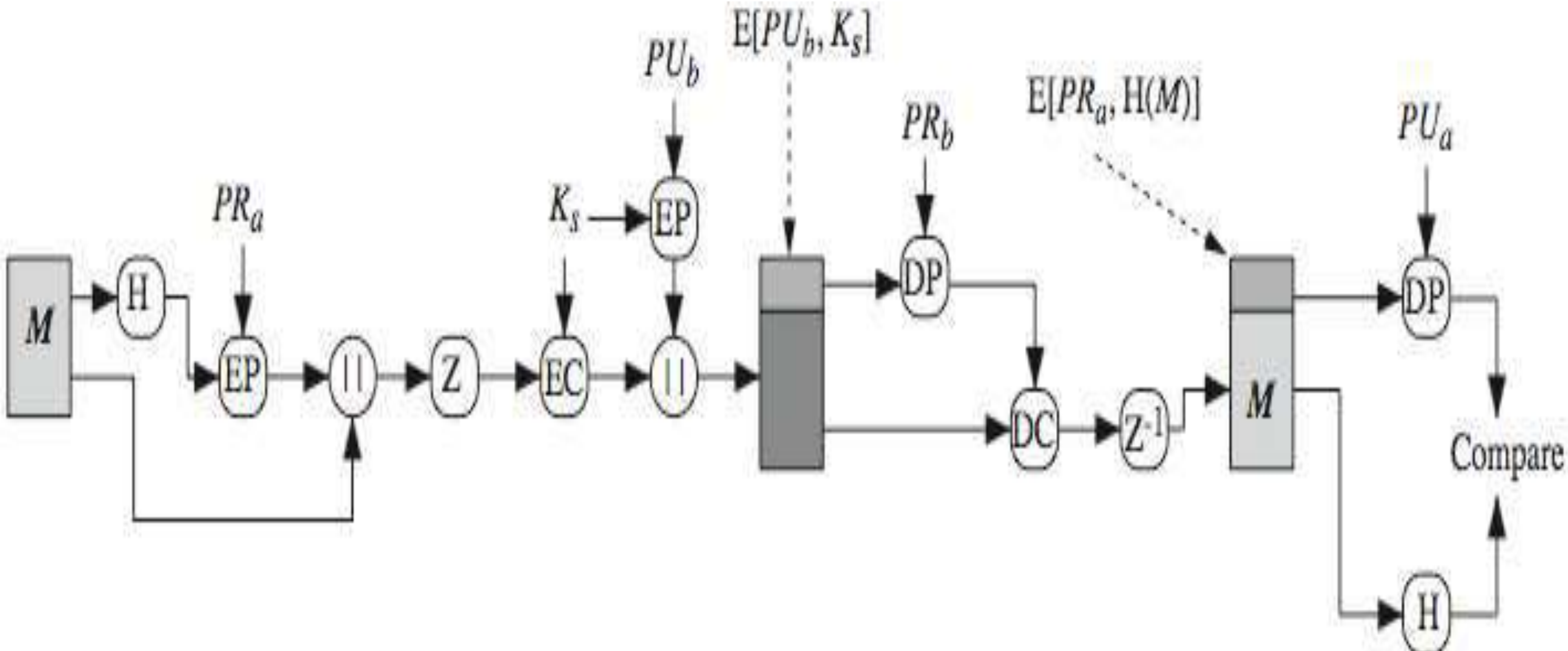


(b) Confidentiality only

PGP Operation – Confidentiality & Authentication

- can use both services on same message
  - create signature & attach to message
  - encrypt both message & signature
  - attach RSA encrypted session key

PGP Operation – Confidentiality & Authentication



(c) Confidentiality and authentication

PGP Operation – Compression

- by default PGP compresses message after signing but before encrypting
- uses ZIP compression algorithm

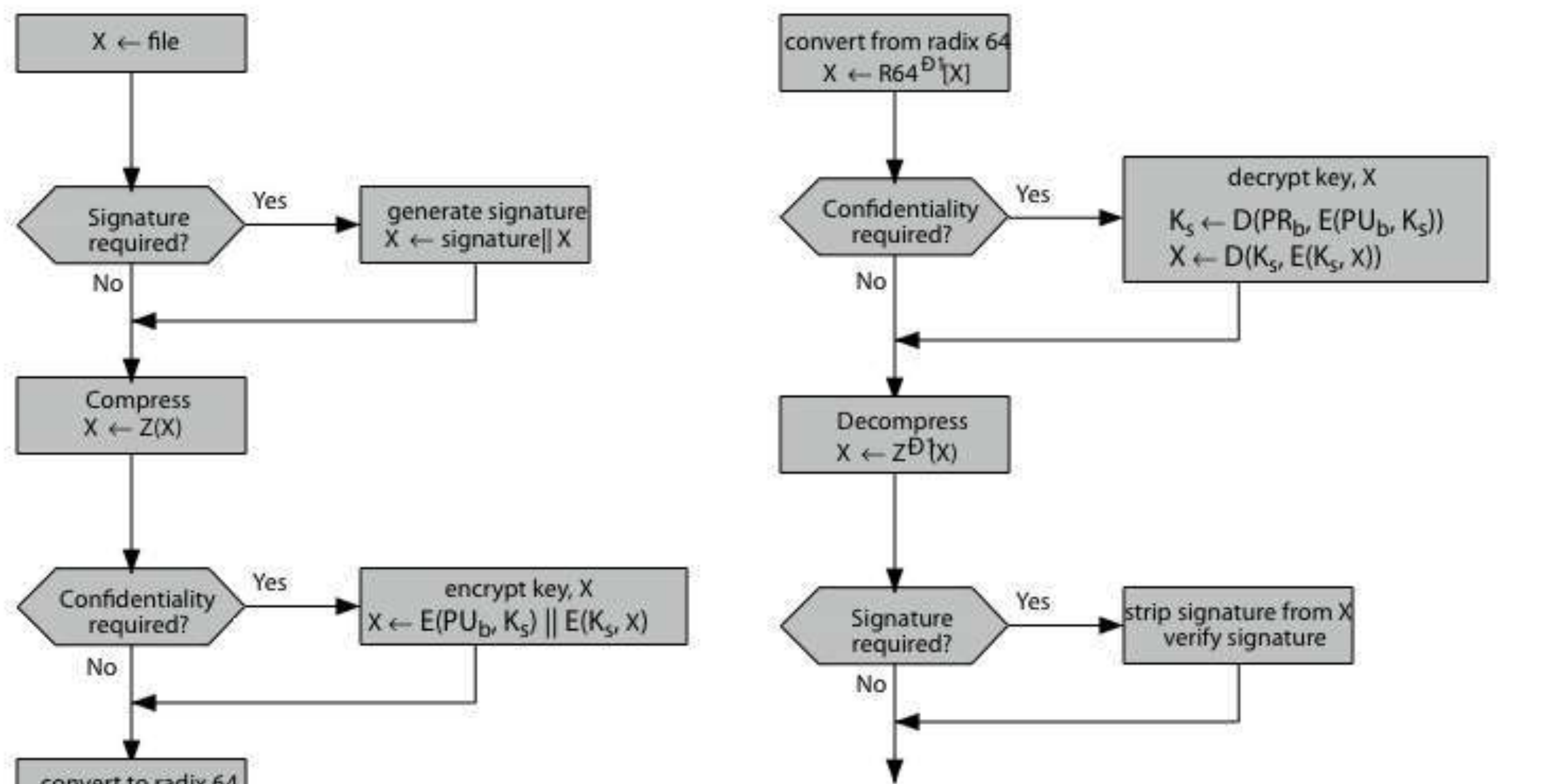
PGP Operation – Email Compatibility

- when using PGP will have binary data to send (encrypted message etc)
- however email was designed only for text
- hence PGP must encode raw binary data into printable ASCII characters
- uses radix-64 algorithm
  - maps 3 bytes to 4 printable characters of 6 bits each
  - R64 table consist of 0 maps to A,1 maps to B,2 maps to C, and so on,26 maps to a,27 maps to b and so on
  - After mappping, the characters are stored in 8 bit ASCII format that consist of total of 32 bits
- Use of Radix-64 expands the mesage by 33%

PGP Segmentation

- ✓ Restricted to maximum message length of 50,000 octets
- ✓ Longer message must be broken up into segments
- ✓ PGP automatically subdivides a message that is too large
- ✓ Receiver strip of all e-mail headers and reassemble the block.

PGP Operation – Summary



(a) Generic Transmission Diagram (from A)

(b) Generic Reception Diagram (to B)

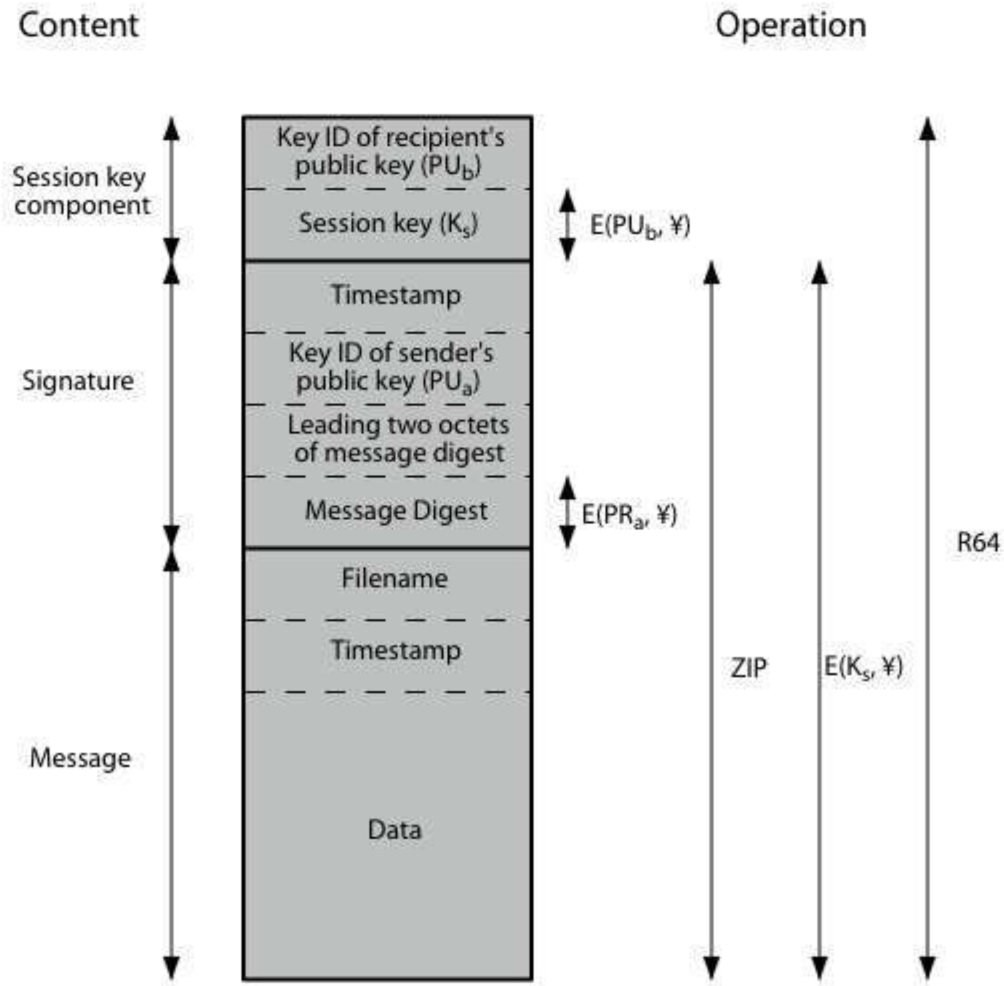
PGP Session Keys

- need a session key for each message
  - of varying sizes: 56-bit DES, 128-bit CAST or IDEA, 168-bit Triple-DES
- generated using ANSI X12.17 mode that uses triple DES alg
- uses random inputs taken from previous uses

PGP Public & Private Keys

- since many public/private keys may be in use, need to identify which is actually used to encrypt session key in a message
  - could send full public-key with every message
  - but this is inefficient and unneccesarily wasteful of space
- rather use a key identifier based on key
  - is least significant 64-bits of the key
  - will very likely be unique
- also use key ID in signatures

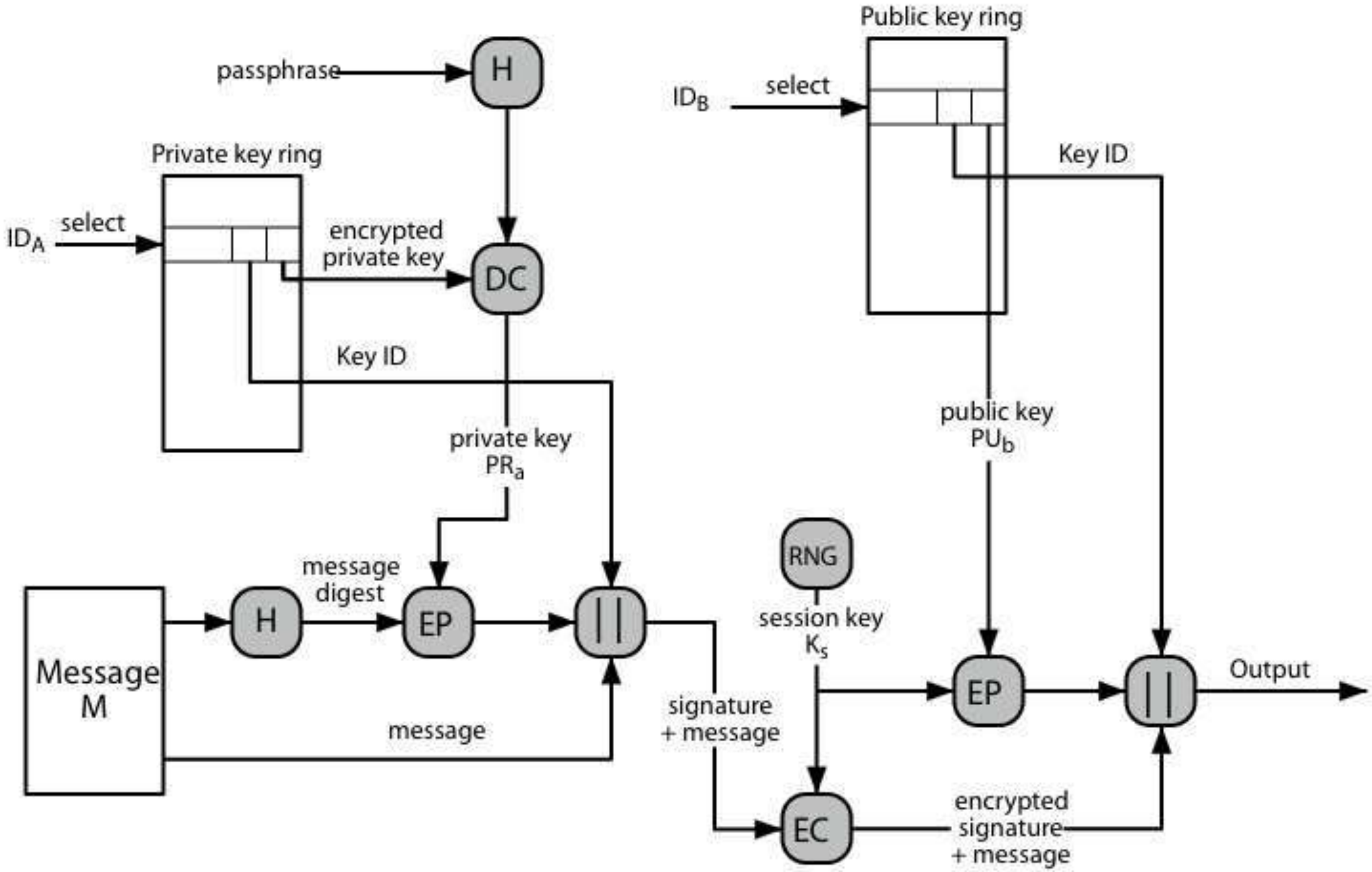
PGP Message Format



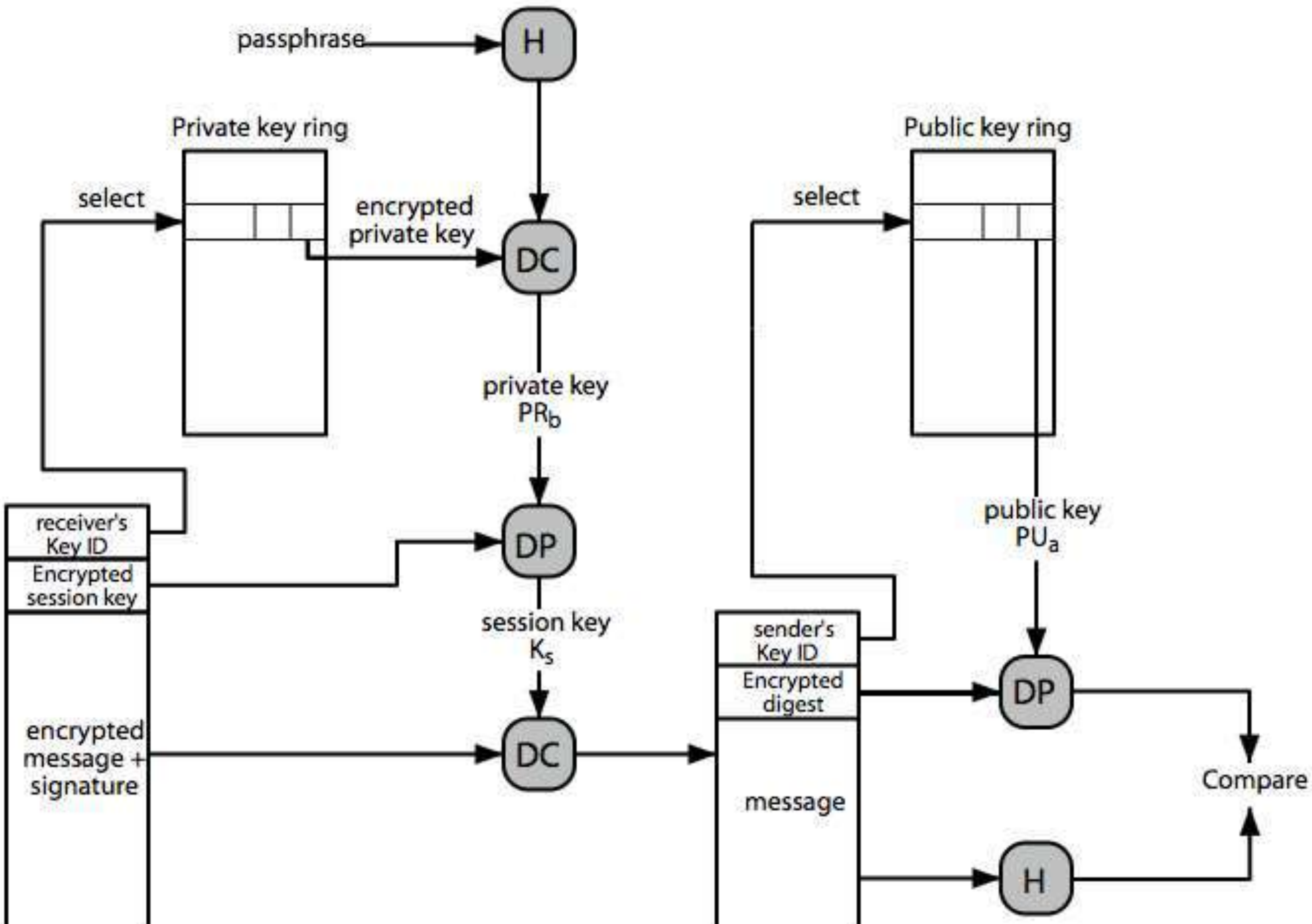
PGP Key Rings

- each PGP user has a pair of keyrings:
  - public-key ring contains all the public-keys of other PGP users known to this user, indexed by key ID
  - private-key ring contains the public/private key pair(s) for this user, indexed by key ID & encrypted keyed from a hashed passphrase
- security of private keys thus depends on the pass-phrase security

PGP Message Generation



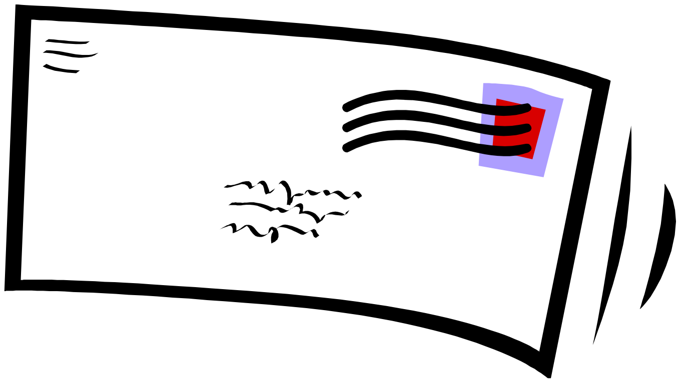
PGP Message Reception



PGP Key Management

- rather than relying on certificate authorities
- in PGP every user is own CA
  - can sign keys for users they know directly
- forms a “web of trust”
  - trust keys have signed
  - can trust keys others have signed if have a chain of signatures to them
- key ring includes trust indicators
- users can also revoke their keys

MIME

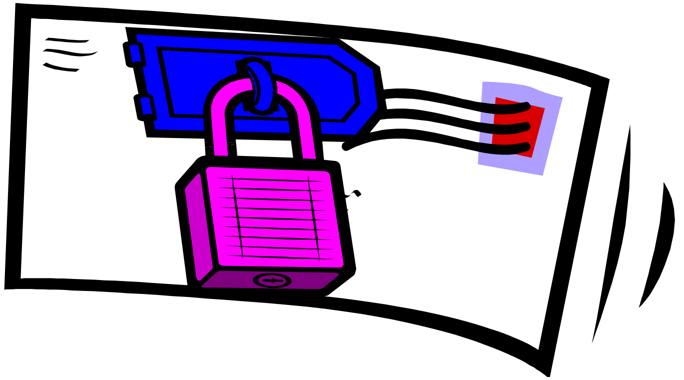


MIME

✓ **Multipurpose Internet Mail Extensions** (**MIME**) is an Internet standard that extends the format of email to support:

- ❖ Text in character sets other than ASCII
- ❖ Non-text attachments: audio, video, images, application programs etc.
- ❖ Message bodies with multiple parts
- ❖ Header information in non-ASCII character sets

S/MIME



S/MIME - Overview

- S/MIME **(Secure/Multipurpose Internet Mail Extension)** is a security enhancement to the MIME Internet e-mail format standard.
- S/MIME is ***not restricted to mail***; it can be used with any transport mechanism that transports MIME data, such as HTTP.
- S/MIME is ***likely to emerge as the industry standard*** for commercial and organizational use, while PGP will remain the choice for personal e-mail security for many.

S/MIME - Overview

- S/MIME provides the following cryptography security services:
  - ✓ Authentication.
  - ✓ Message Integrity.
  - ✓ Non-repudiation of origin.

  By using digital signing

  - ✓ Privacy and data security. By using encryption

- There are three versions of S/MIME:
  - S/MIME version **1 (1995)**- was specified and officially published in 1995 by RSA Security, Inc.
  - S/MIME version **2 (1998)**- was specified in a pair of informational RFC documents - RFC 2311 and RFC 2312 - in March1998.
  - The work was continued in the Internet Engineering Task Force IETF for S/MIME Mail Security (SMIME) and resulted in S/MIME version **3 (1999)** specified in RFCs 2630 to 2634 in June 1999.

MIME - Overview

- RFC 822 defines a format for text messages that are sent using electronic mail.

SMTP/RFC822 scheme limitations:

1. SMTP cannot transmit executable files or other binary files.
2. SMTP cannot transmit text data that includes national language characters because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
3. SMTP servers may reject mail message over a certain size.
4. SMTP gateways that translate between ASCII to EBCDIC suffer translation problems.
5. Some SMTP implementations do not adhere completely to the SMTP standard defined in RFC 822.

MIME (contd.)74

MIME specification includes the following elements:

Five new message header fields. These fields provide information about the body of the message.

1. **MIME-Version: Must have the parameter value 1.0. This field indicates that the message** conforms to RFCs 2045 and 2046.
2. **Content-Type: Describes the data contained in the body with sufficient detail that the receiving** user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner.
3. **Content-Transfer-Encoding: Indicates the type of transformation that has been used to** represent the body of the message in a way that is acceptable for mail transport.
4. **Content-ID: Used to identify MIME entities uniquely in multiple contexts.**
5. **Content-Description: A text description of the object with the body; this is useful when the** object is not readable (e.g., audio data)

MIME (contd.)

Here is a summary of the different MIME content types:

Type	Subtype	Description
Text	Plain Enriched	Unformatted text (ASCII or ISO 8859). Provides greater format flexibility.
Multipart	Mixed Parallel Alternative Digest	The different parts are independent but are to be transmitted together. Should be presented to the receiver in their original order. Differs from mixed only in that no order is defined. The different parts are alternative versions of the same information. Similar to Mixed but the default type/subtype of each part is message/rfc822.
Message	rfc822 Partial External body	The body is itself an encapsulated message that conforms to RFC822. Used to allow fragmentation in a transparent way to the recipient. Contains a pointer to an object exists else where.

MIME (contd.)

Type	Subtype	Description
Image	Jpeg gif	The image is in JPEG format. The image is in GIF format.
Video	Mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8kHz
Application	Postscript Octet-stream	Adobe Postscirpt. General binary data consisting of 8-bit bytes.

S/MIME - Functions

- **S/MIME is based on the Cryptographic Message Syntax (CMS) specified in RFC 2630.**
- ***<u>Enveloped data:</u>*** *(*encrypted content and associated keys)

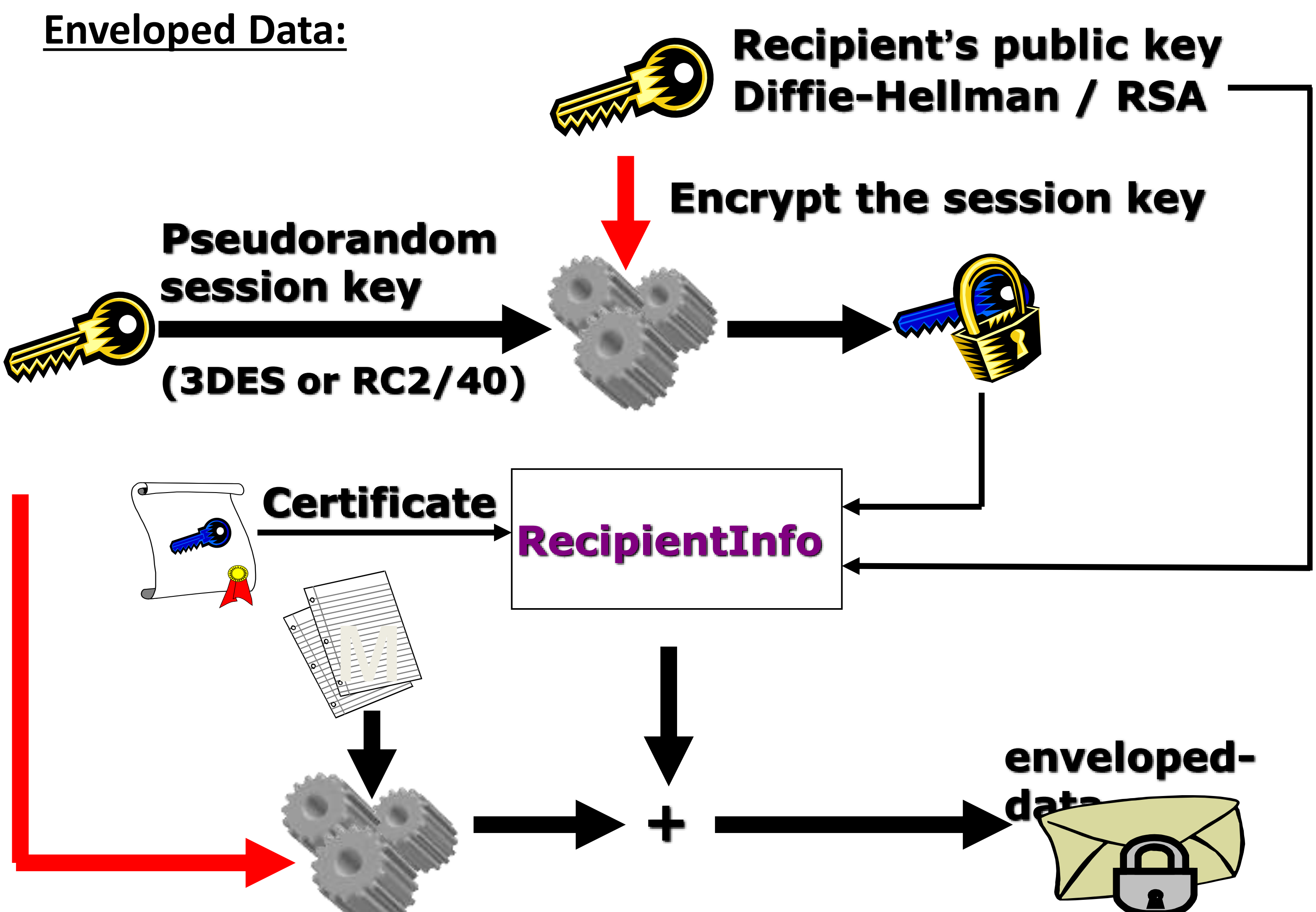
  This consists of encrypted content of any type and encrypted contents encryption keys for one or more users. This functions provides privacy and data security.

- ***<u>Signed data</u>***: (encoded message + signed digest)

  A digital signature is formed by signing the message digest and then encrypting that with the signer private key. The content and the signature are then encoded using base64 encoding.

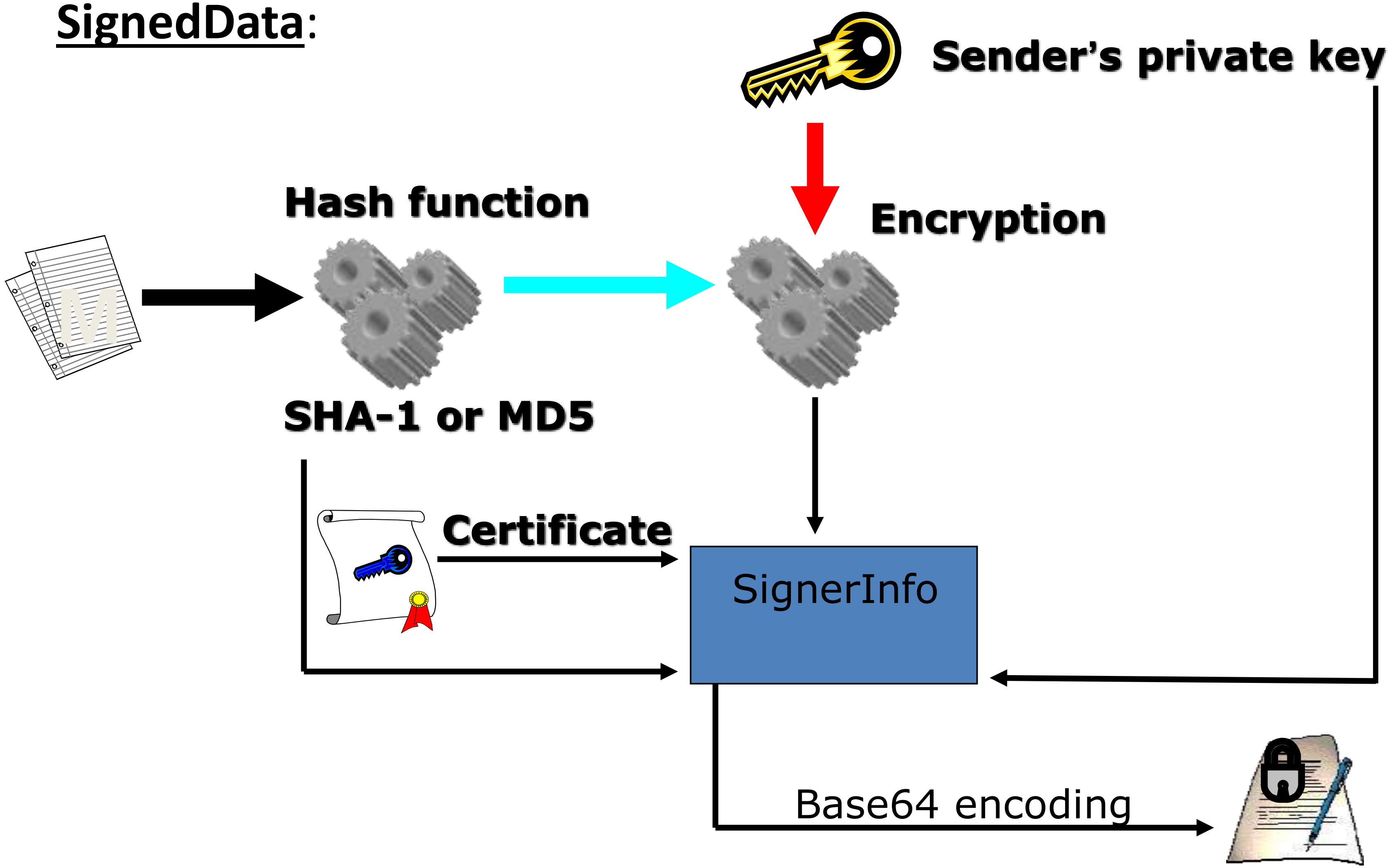
  This function provides authenticity, message integrity and non-repudiation of origin.

S/MIME - Message



S/MIME - Message

SignedData:



S/MIME - Functions

- **<u>SignerInfo:</u>** allows the inclusion of unsigned and signed attributes to be included along with a signature.
  - Signing Time
  - sMIME Capabilities
  - sMIME Encryption Key Preference

S/MIME - Functions

- ***Clear signed data:*** *(*clear text message + encoded signed digest)

  In this case a digital signature of the content is formed, However only the signature is encoded with base64.

- ***Signed and enveloped data:*** (nesting of signed & encrypted entities)

  Because of S/MIME encapsulating capability (multipart type), signed only and encrypted only entities may be nested, so that encrypted data may be signed and signed data may be encrypted.

IP Sec – Overview, AH & ESP

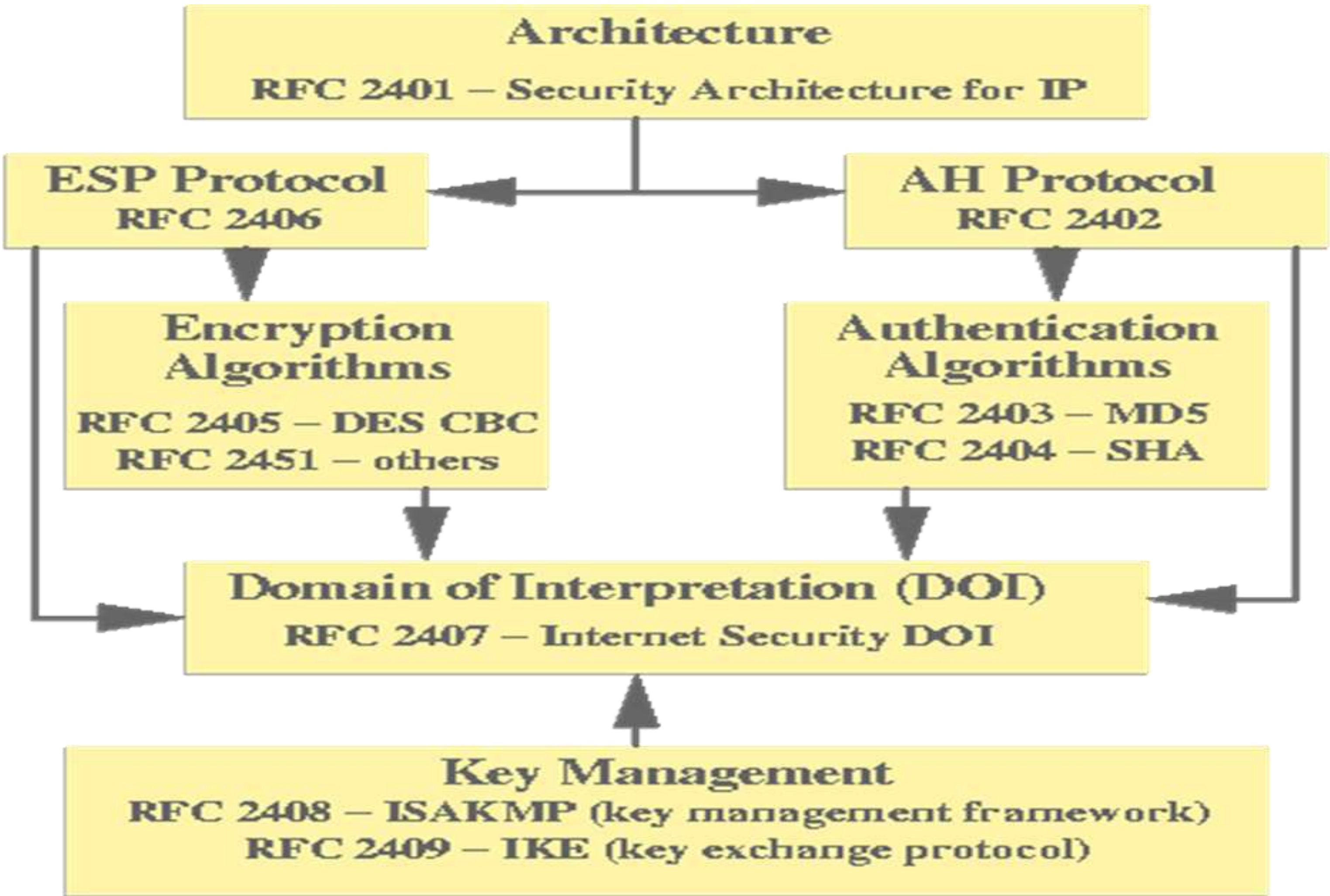
IP Security: Overview

- ✓ IPSec is an abbreviation for IP security, which is used to transfer data securely over unprotected networks like “Internet”.
- ✓ It acts at the networks layer and is part of IPv4,IPv6.
- ✓ It provides :
  - ✓ Sender encrypts packets before sending them on the network.
  - ✓ Receiver authenticates packets.
  - ✓ Anti replay checks to reject duplicate packets preventing DOS attack.
  - ✓ IKE is the key exchange mechanism to securely exchange keys

IPSec Services

- ✓ Access control
- ✓ Connectionless integrity
- ✓ Data origin authentication
- ✓ Rejection of replayed packets
- ✓ Confidentiality (encryption)
- ✓ Limited traffic flow confidentiality

IPSec Architecture – Documents



Authentication Header (AH)

- ✓ The Authentication Header provides support for data integrity and authentication of IP packets.

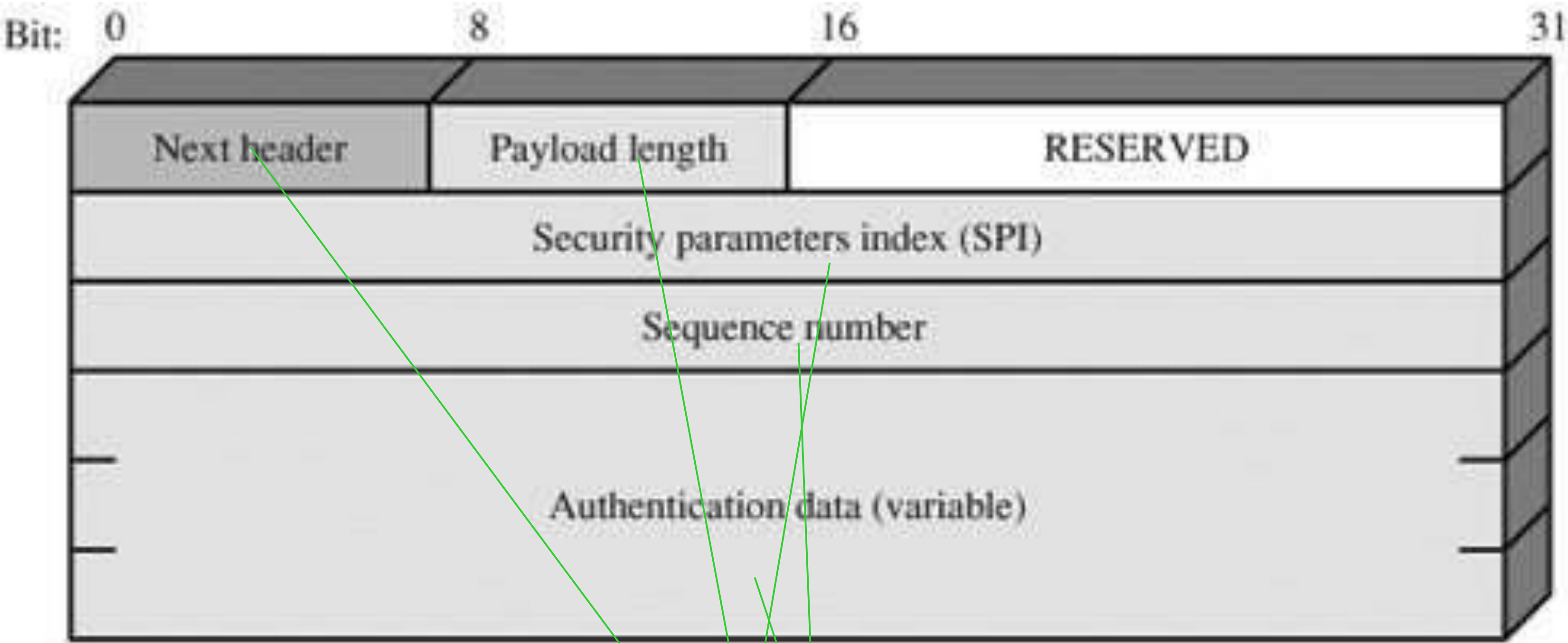
  - ✓ The data integrity feature ensures that undetected modification to a packet's content in transit is not possible.

  - ✓ The authentication feature enables an end system or network device to authenticate the user or application and filter traffic accordingly.

- ✓ The AH also guards against the replay attack.

- ✓ Authentication is based on the use of a message authentication code (MAC)

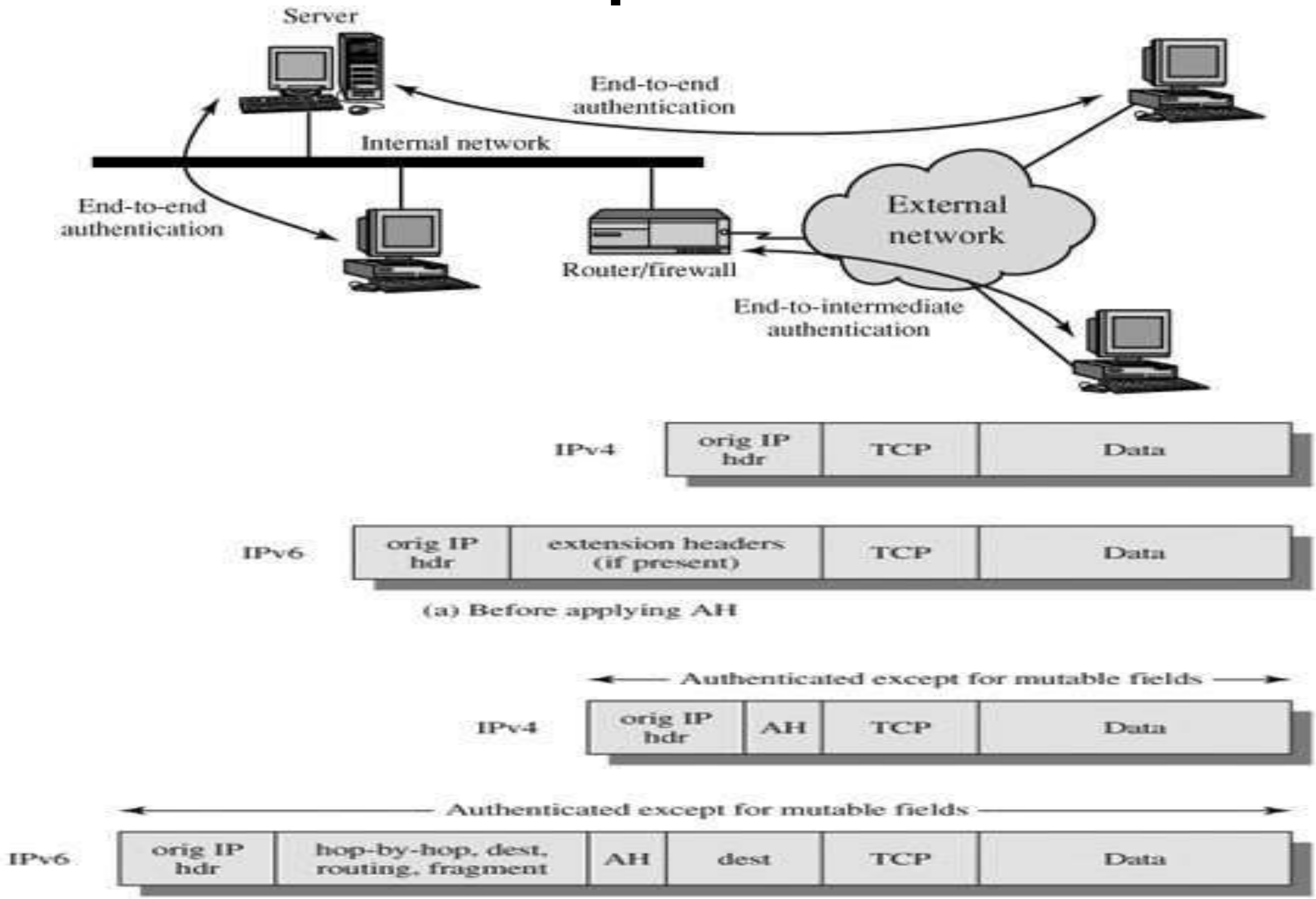
Authentication Header format



Contains the Integrity Check Value (ICV) that is used to verify the integrity of the message. The receiver calculates the hash value and checks it against this value (calculated by the sender) to verify integrity.

packet for a security association with this number has not been received already. If one has been received, the packet is rejected.

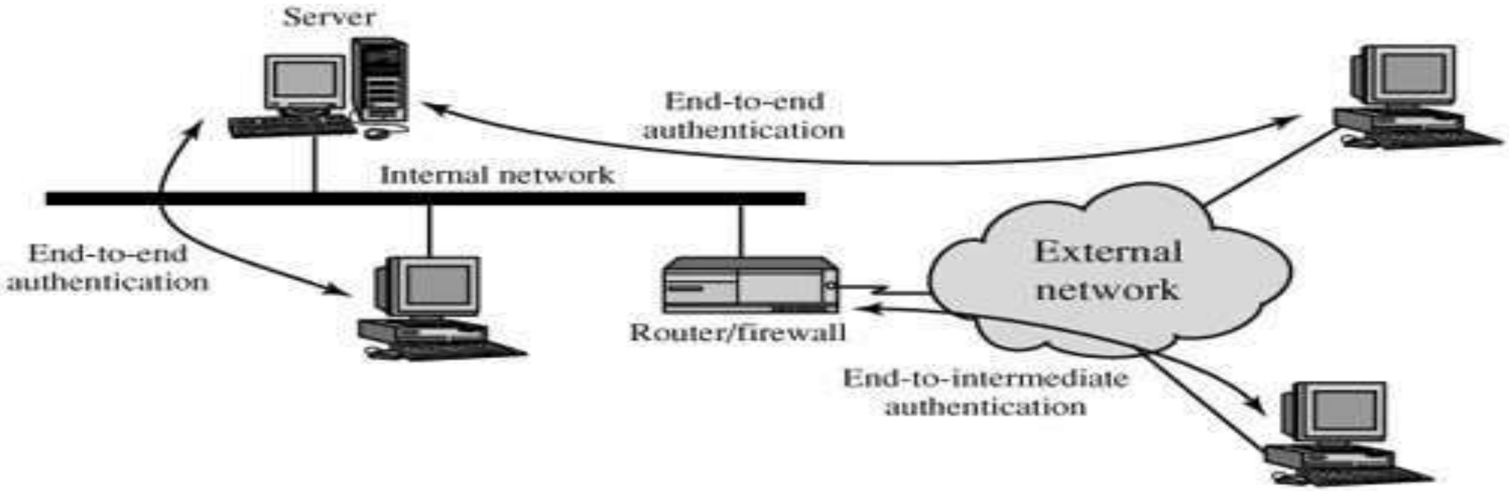
Transport Mode

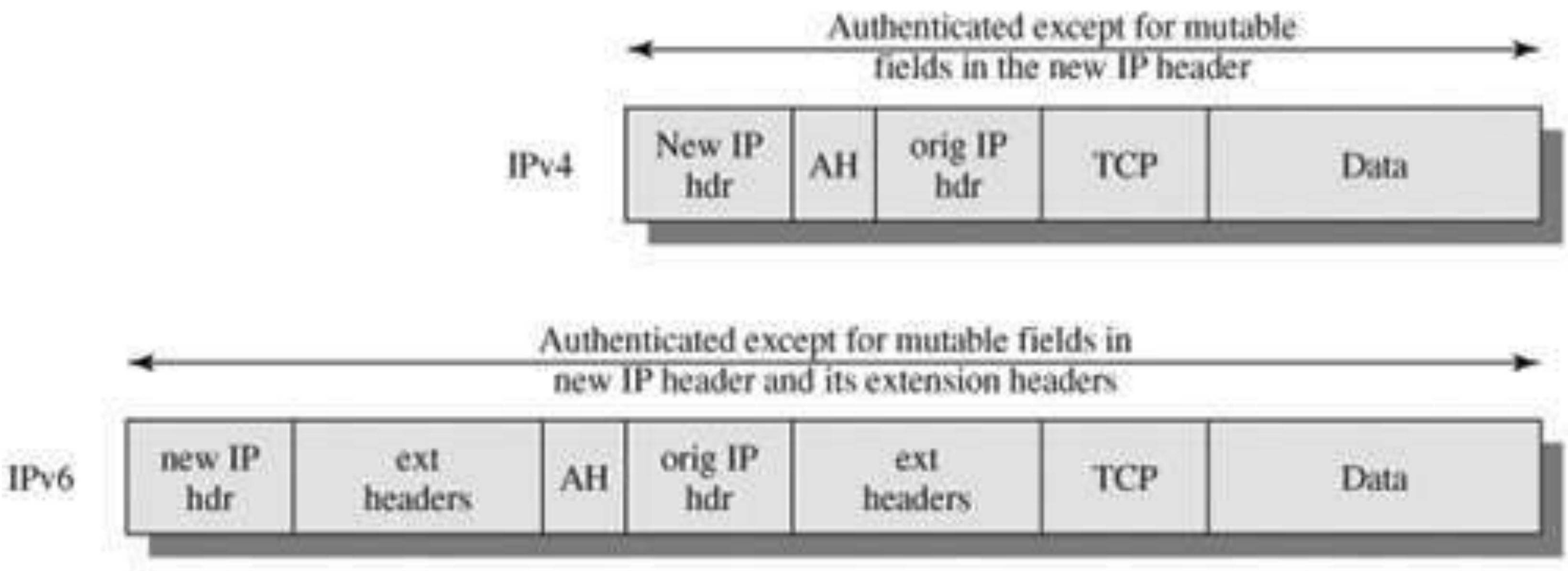


(a) Before applying AH

(b) Transport mode

Tunnel Mode


Server
End-to-end
authentication
Internal network
End-to-end
authentication
External
network
Router/firewall
End-to-intermediate
authentication


Authenticated except for mutable
fields in the new IP header
IPv4
New IP
hdr
AH
orig IP
hdr
TCP
Data
Authenticated except for mutable fields in
new IP header and its extension headers
IPv6
new IP
hdr
ext
headers
AH
orig IP
hdr
ext
headers
TCP
Data

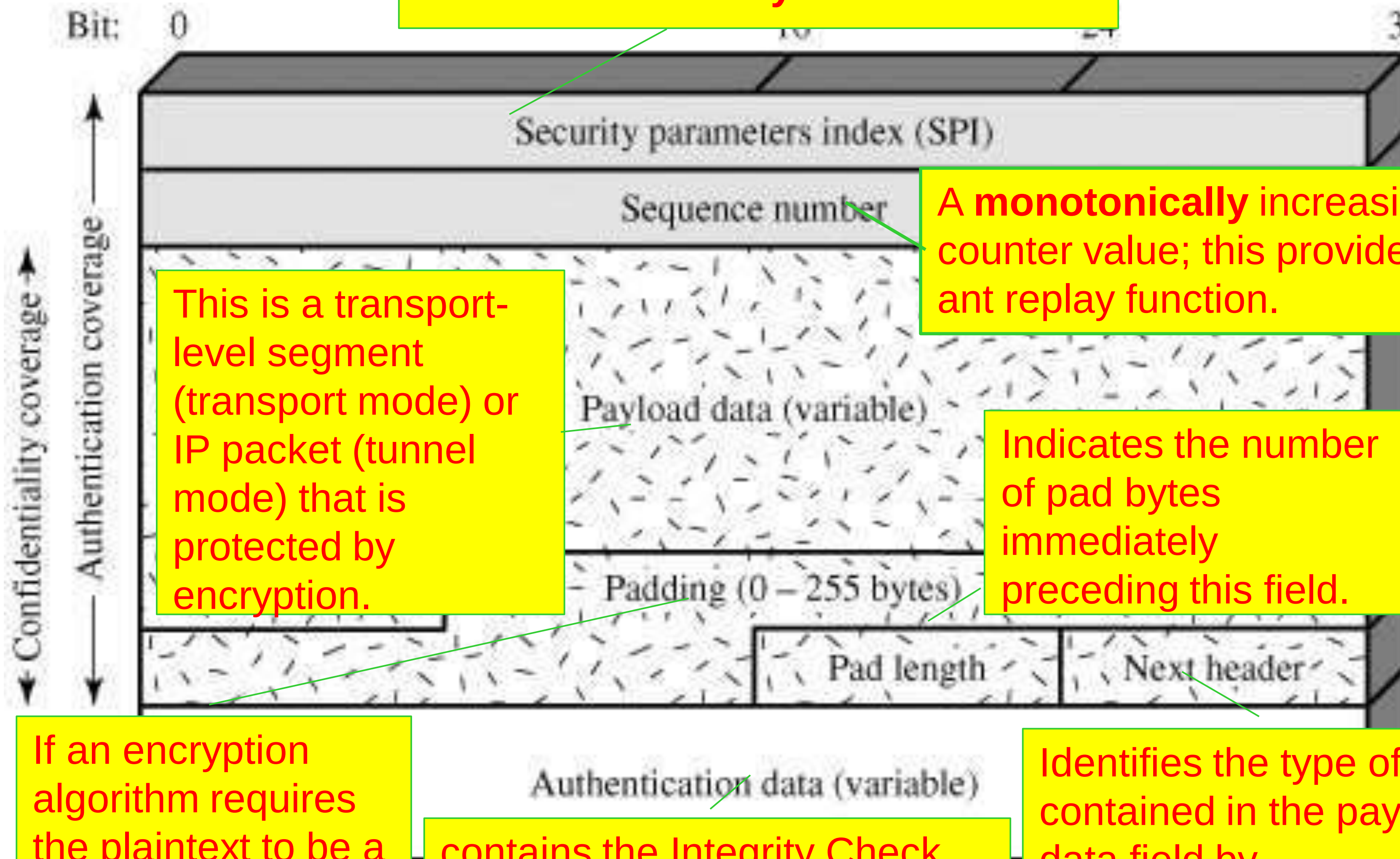
Encapsulation Security Payload(ESP)

- The Encapsulating Security Payload provides

- confidentiality services, including confidentiality of message contents and limited traffic flow confidentiality.

- As an optional feature, ESP can also provide an authentication service.

ESP Format



Identifies a security association.

A **monotonically** increasing counter value; this provides an ant replay function.

This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.

Indicates the number of pad bytes immediately preceding this field.

If an encryption algorithm requires the plaintext to be a multiple of some number of bytes

contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.

Identifies the type of data contained in the payload data field by identifying the first header in that payload

Encryption and Authentication Algorithms

- The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service.

- If the algorithm used to encrypt the payload requires cryptographic synchronization data, such as an initialization vector (IV), then these data may be carried explicitly at the beginning of the Payload Data field.
- If included, an IV is usually not encrypted, although it is often referred to as being part of the ciphertext.

Algorithms

Three-key triple DES

- RC5
- IDEA
- Three-key triple IDEA
- CAST
- Blowfish

Encryption and Authentication Algorithms

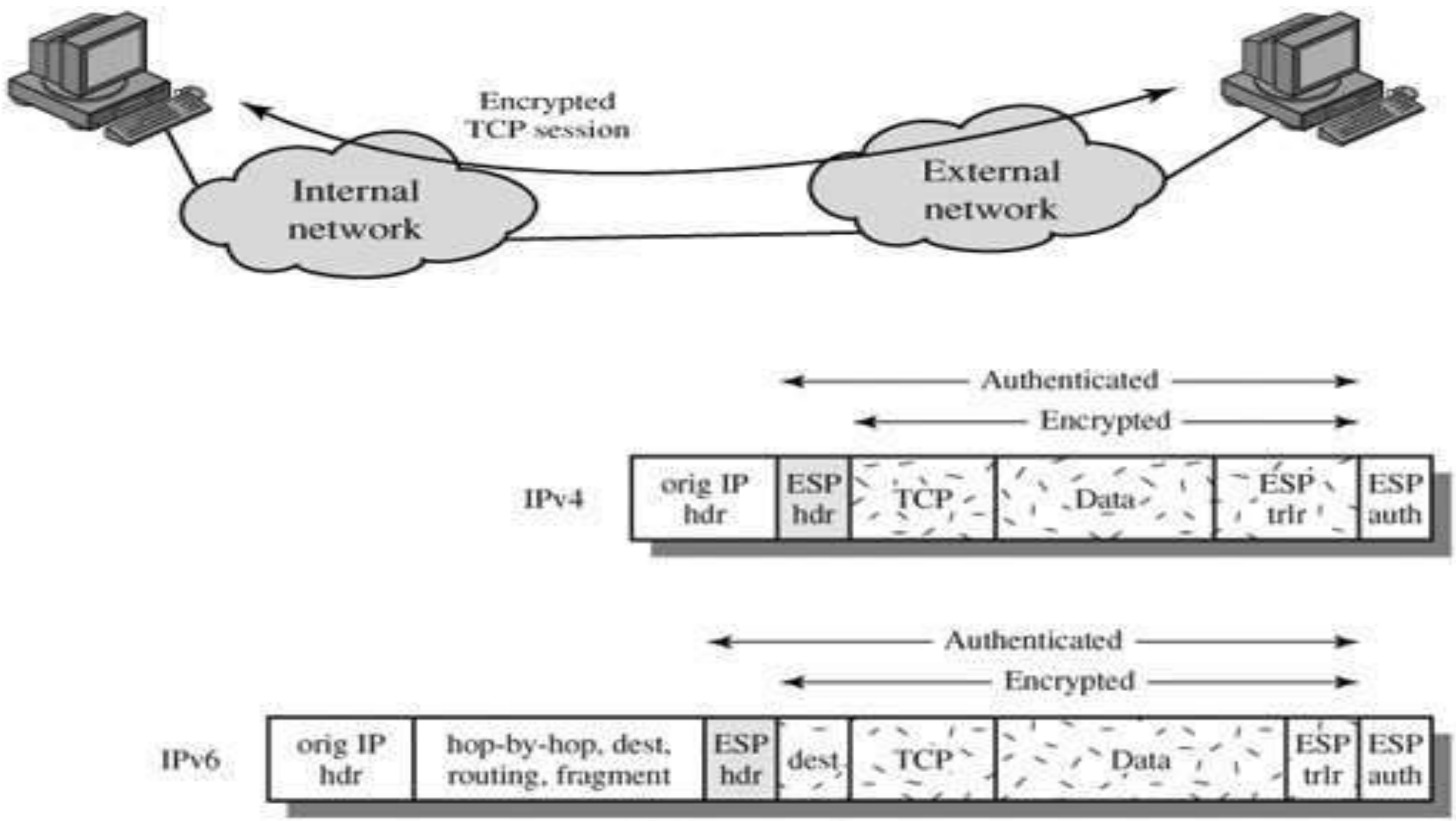
- The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service.
- If the algorithm used to encrypt the payload requires cryptographic synchronization data, such as an initialization vector (IV), then these data may be carried explicitly at the beginning of the Payload Data field.
- If included, an IV is usually not encrypted, although it is often referred to as being part of the ciphertext.

Algorithms

Three-key triple DES

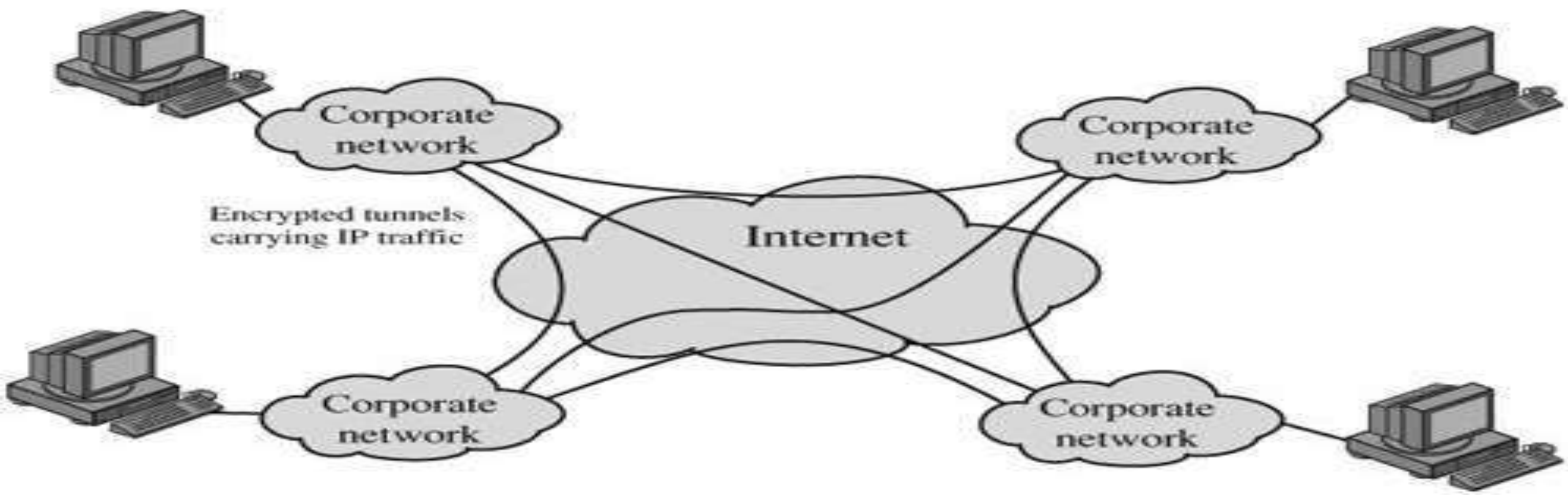
- RC5
- IDEA
- Three-key triple IDEA
- CAST
- Blowfish

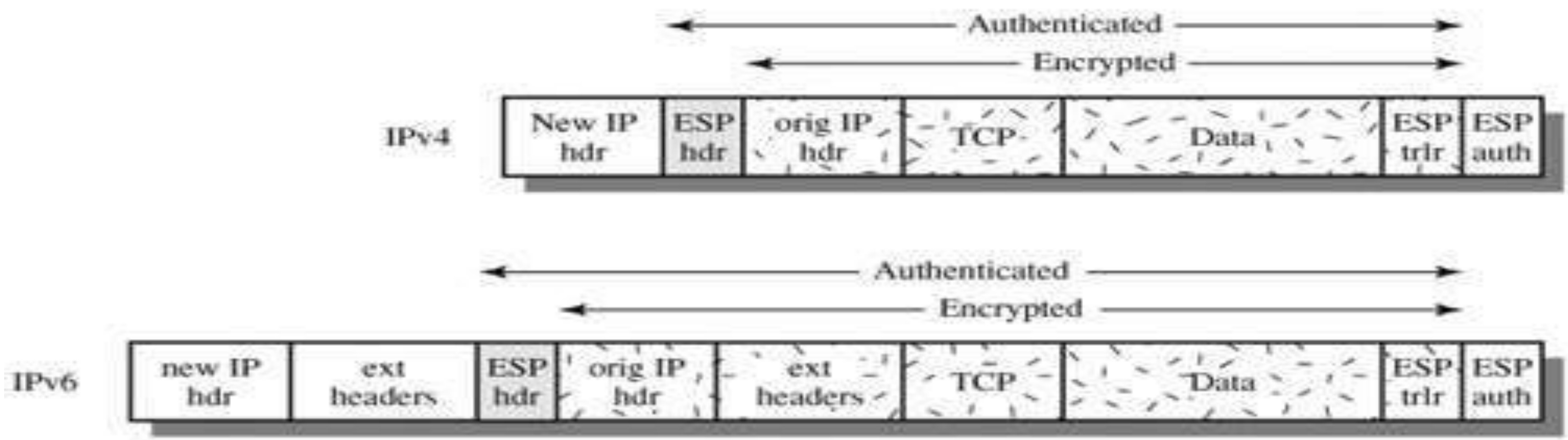
Transport and Tunnel Modes



(a) Transport mode

Transport and Tunnel Modes





(b) Tunnel mode

Unit VIII

Intruders:

→ It is one of the important threat for information security and three important classification of intruders are

1. masquerader
2. Misfeasor
3. clandestine user.

Masquerader:

→ An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit valid user's account

→ Generally, masquerader is an outsider.

Misfeasor:

→ A valid user who accesses data, programs or resources for which such access is not authorized.

→ A valid user who is authorized to access the data or programs but misuses his or her privileges

→ Generally, misfeasor is an insider

clandestine user:

→ A person who captures administrative rights of a system is called as clandestine user.

→ After that he uses them to control other sources

→ A clandestine user can be either an outsider or an insider.

→ Intruders attack range from benign to the serious.

→ Some users of internet just for time pass purpose explore the internet for watching other websites with or without proper permission.

→ Technically speaking these people test the security of a web site. They are known as benign intruders.

→ On the other hand some intruders explore the internet intentionally to modify or disrupt the data on the internet that causes disturbances to the websites. They are called as serious intruders

Intrusion Techniques:

→ The objective of the intruder is to gain access to a system or to obtain more privilages.

→ Generally passwords of all users are stored in some password file. if it is not secured file then automatically people will access it and some times they may change it too.

→ Once intruder obtained permission with one password, definitely they search for passwords of valid users.

→ Once the password file is accessible or data is visible the intruders will modify the contents.

→ Password file can be protected in one of two ways.

① One way function: → OS stores all password in encrypted form.

→ When a user presents the password, the

System transforms that password by encrypting and compares it with the stored encrypted password value.

→ In practice, the system usually performs a one way transformation in which the password is used to generate a key for the one way function and in which a fixed length output is produced.

② Access control:

Access to the password file is limited to one or very few accounts

Some of techniques used by intruder to get the password are

① Try default passwords used with standard account

Ex: Exam login account administrator puts exam as password

Internet login account administrator puts internet as password.

② Try all short passwords (those of one of 3 characters,

③ Try to get personel information like spouse name, children names and try them in password entry.

④ Try all popular movie names

⑤ Try all the words in electronic dictionary words

⑥ Try users phone numbers, room numbers and personel identification numbers.

→ The main key goal often is to acquire the password and so then exercise to access rights of the owner.

Intrusion detection:

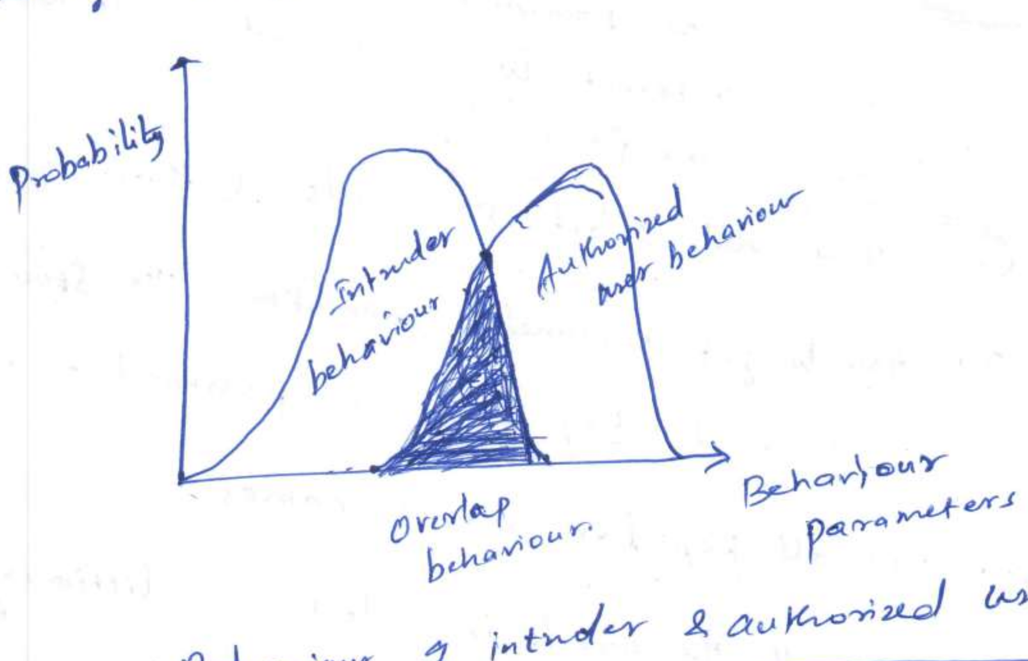
→ For every system there is a ~~change~~ chance of attack by intruder, so intrusion prevention is difficult task.

→ Intrusion detection is very important because.

1. Quickly detect intrusion and next step is to eliminate the intruders.
2. It can serve as anti threatening system so automatically prevents from further intrusion
3. Strengthens the intrusion prevention facility.

→ By observing the behaviour of intruders and authorized users, Intrusion detection system has been designed.

→ Behaviour of these two kinds are given in the following diagram



Behaviour of intruder & authorized user

→ There are several approaches for intrusion detection

Some of them are

① Statistical Anomaly detection
② Rule based detection.

① Statistical anomaly detection:

→ This is based on Collection of data relating to the behavior of valid user Over a Specified period

→ Then the statistical tests applied on this data to obtain average behavior to find the difference with authorized behavior

→ Identify number of events that occurs frequently and detect who are unauthorized user. This kind of detection is known as thershold detection

→ Observe the regular behavior of the users and find individual behavior changes. This kind of detection is known as profile based detection.

② Rule based detection:

→ Involves an attempt to define a set of rules that can be used to decide that a given behavior is authorized user behavior or intruder behavior

→ Rules are developed to detect deviation from previous usage patterns, it is known as anamoly detection

→ An intelligent or expert System approach that automates the process of finding unauthorized user behaviors and it is known as perpetration identification

Audit records:

→ It is one of the tool for intrusion detection. There are two kinds of plans that are based on audit records used for intrusion detection.

1. Native audit records
2. Detection specific audit records

1. Native audit records:

→ An audit record that is included in operating system itself is called as native audit record

→ All multiuser OS includes native audit records in form of accounting software that collects information on user activity.

→ Main purpose of this accounting software is that it does not require any additional software for collecting details.

→ The problem with native audit record is that it is not in convenient form and may not contain all the needed information to user.

2. Detection specific audit records:

→ The required information alone is collected in detection specific audit record

→ It gives information that is only required to the intrusion detection system.

→ Advantage is that it contains Vendor Specific information and disadvantage is that it takes additional overhead for collecting information.

→ One such detection specific format is as follows.

Subject	Action	Object	Exception condition	Resource usage	Timestamp

→ The purpose of field is as follows

Field name	Purpose
① Subject :	Indicates initiators of action such as terminal user
② Action :	Required operation performed by subject.
③ Object :	Indicates receptors of action such as programs messages, files, records etc
④ Exception Condition :	Indicates exception, if any that is raised on return
⑤ Resource usage :	Resource that are used by each element
⑥ Time stamp :	To identify when the action took place.

Password Management :

→ All multiuser Systems requires that user must have username and password to login.

→ The password protects user not to access by other users.

→ The user name must be unique and it tells that who is authorized user and who is unauthorized user.

→ All the user names along with passwords are stored in a password file.

→ When any user gives his user name and password, the system first verifies whether there is record for user name or not.

→ If there is no record the given user name is an unauthorized user, so immediately system replies Invalid user.

→ The user name also tells privileges granted to user, whether he is normal user, group user, guest or administrator.

Unix password Scheme:

→ In unix passwords are stored in encrypted form using DES algorithm in password file

→ When user is created for the first time by an administrator, it takes password along with some salt value

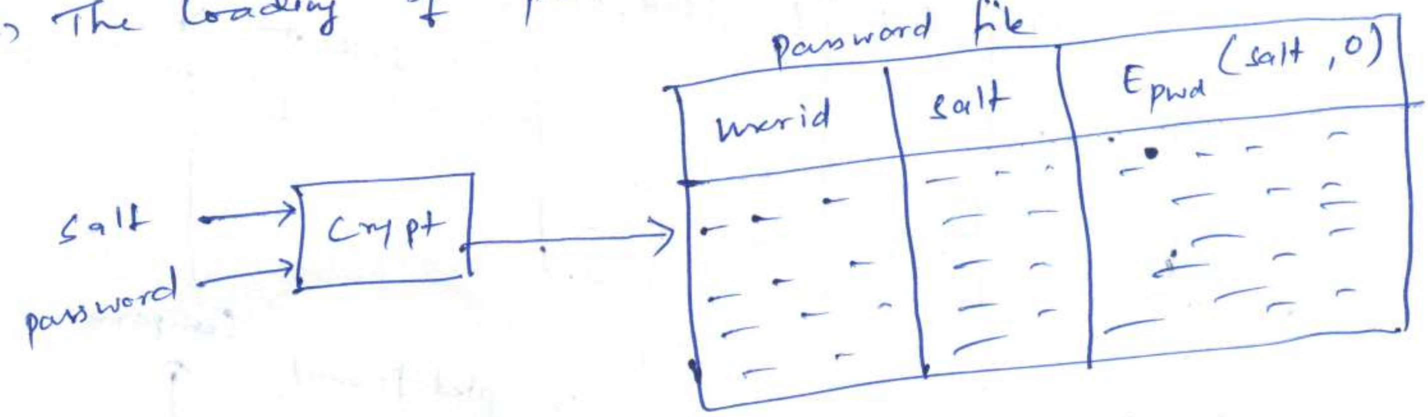
→ Salt value is in general time when he is created.

→ Generally user name taken as key so converted into a 56 bit value by using 7 bit ASCII and salt value used for increasing security aspects.

→ Result of encryption after several encryption is a 11 character password.

→ Advantage of using salt value is to prevent duplicate password, increase the length of password and also prevent the use of a hardware implementation of DES.

→ The loading of password is shown below



New password loading

→ There are three field in the password file, first one is user id, it contains loginname, second one is salt that contains time when the user was created in 12 bit length

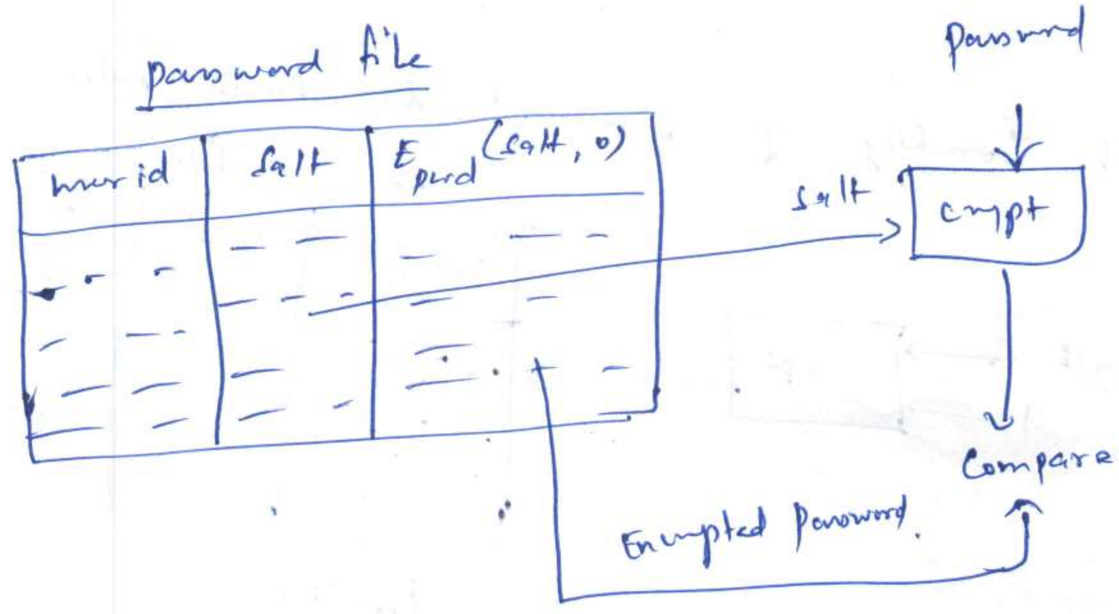
→ But DES requires 64 bit data block so for these 12 bits other 52 bits are added in the form of zero's.

→ Third field is encrypted password that encrypts the salt value along with padding of zero's with the help of password as key. This kind of alg refered in UNIX terminology as "crypt(3)"

→ when a user login into the account, system takes userid and searches for the user id in password file, if user id is not found results as invalid account.

→ In other case when userid is found then it takes existing salt value store in password file and encrypts it with the help of given password.

→ Result of encrypted password and value stored in the third field are compared, if both are same then given user id and password are valid otherwise it is invalid.



Password verification

Password Selection Strategies:

There are four basic techniques for password selection. They are

1. using Computer generated password
2. Educating users in password selection
3. Reactive password checking
4. Proactive password checking

① using Computer generated Password:

→ when we are allowed to create a password in UNIX OS, it provides some predefined passwords which is mixture of alphabets and numerals.

→ User can select one of the passwords from the list.

→ The Computer generated password are random in nature So user cannot remember it.

Ex: If a password like A5OPQRTY CXCPPRAT. it is not possible by the user to remember it.

→ Many users are not in a position to accept the Computer generated passwords, they give their own.

→ FIPS PUB is one of the automated password generators and it provides pronounceable passwords as well as Contatenates some words together.

② Educating users in password selection:

→ In many organizations there is a policy for giving passwords, but unfortunately people not follow these policies.

→ This policy educates users in password selection, which one is better password and which one is not a good password

→ Some people are not in a position to decide a strong password. To over come this problems user can reverse the password and make first and last characters of lower case or uppercase letters and make not to guess.

→ One of the regular phenomenons in password selection is Replacing of character 'o' with the letter zero

③ Reactive password checking:

→ A password strategy that runs its own password cracker time to time to find guessable passwords is Reactive password checking

→ System cancels any passwords that are entered by user is guessed and same thing is notified to the user.

→ This approach has drawback that intruder tries all passwords not only guessable or pronounceable passwords but they spent their Complete time to break the security

4. Proactive password checking:

→ In this scheme user allowed to select password and proactive password checker verifies that password entered is feasible or not.

→ It means that whether the given password is a guessable or not.

→ Depends on this strategy, system accepts or rejects the password

Firewalls :

→ We are connecting one computer to another computer to share information in a network.

→ sharing information among computer is a good thing but leads to lot of problems like a non authenticated user may use our services or he may add new data to our database that may corrupt our secured data.

→ We are classifying the attacks into two types

1. Leaking of information
2. Adding outside elements.

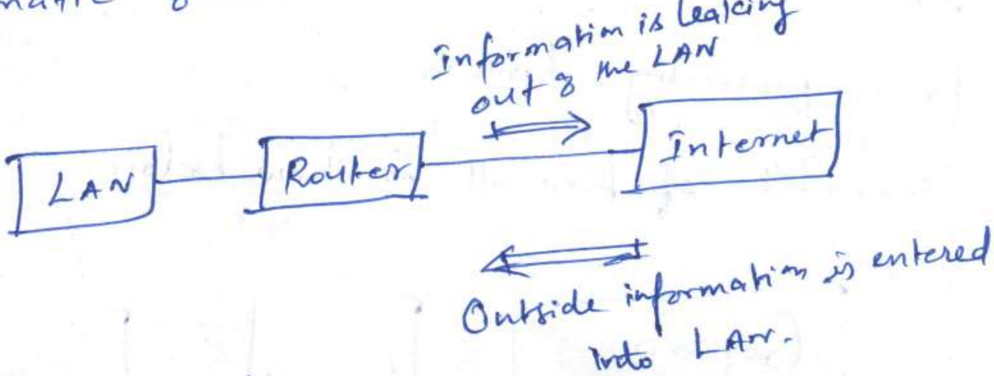
① Leaking of information:

→ Most of the corporation are having large amount of valuable information and confidential data in their network. In this type of attack people may leak the critical information to the competitors.

② Adding Outside elements:

→ Adding the outside malpractice programs Such as viruses and worms may lead to Corrupt our secured data

→ Schematic of the two attacks are shown below



Kinds of attacks

→ To avoid this attacks, we must have mechanisms which can ensure that important information remains as it is and also prevents the outside attackers to enter into LAN.

→ One way to achieve is encryption where the sender encrypts the information which was not understable by the outsiders

→ Encryption does not work in all cases, better scheme are desired to ~~achieve~~ avoid outside attack on the Corporate N/w is firewall.

→ The Working principle of firewall is as Security guard standing outside an important persons house.

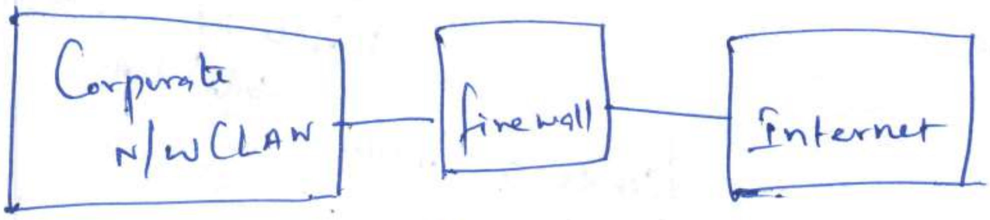
→ Generally firewall is placed between Corporate network and the internet

→ firewall guards Corporate N/w by standing between the ~~internet~~ network and outside network.

→ Whenever data is Coming into Corporate N/w it must pass through firewall and when data is going out of Corporate N/w it must also pass through firewalls.

→ firewall will decide whether to allow the flow or to stop from proceeding further.

→ Schematic of firewall is shown below



firewall.

Firewall characteristics:

1. All the traffic from inside to outside and vice versa must pass through the firewall. This is achieved by physically blocking all access to LAN except via Firewall

2. Only authorized traffic as defined by Local Security policy will be allowed to pass

3. firewall itself is immune (protected) to penetrate. This implies the use of trusted System with a Secured Operating Systems.

4. firewall uses various Control access method to enforce the Security activity

5. Service Control:

It determines type of internet service it is accessing

6. Direction Control:

Determines the direction ~~of~~ flow, in which direction the service request is initiated

7. User Control:

firewall is used to control the user access. In this firewall allows the user to access ~~based~~ the data based on the permission given

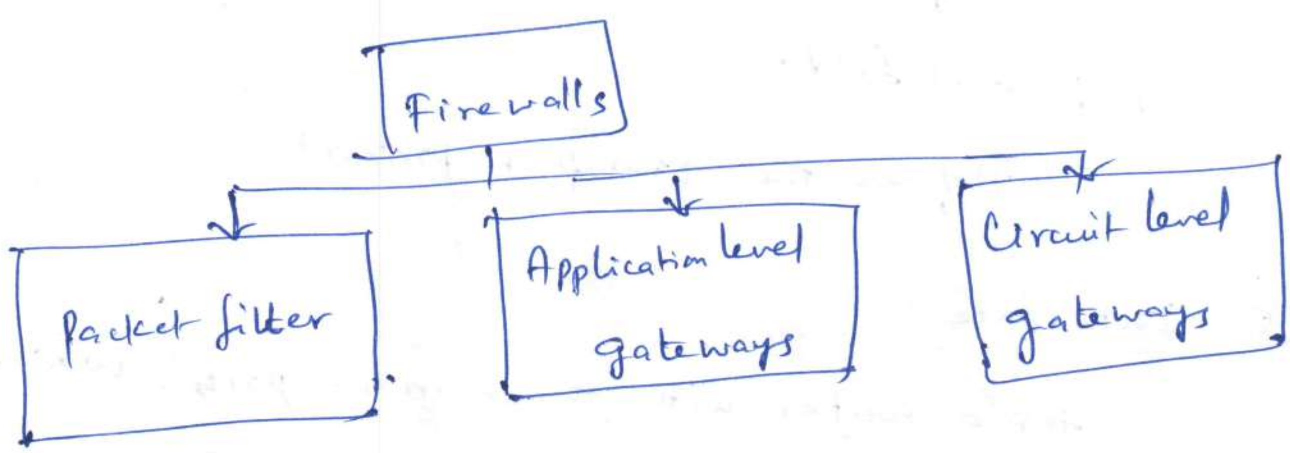
8. It controls the user from doing malpractices on the secured data.

9. firewall itself must be strong enough, so as to render attacks on it useless.

Types of firewalls:

There are 3 types of firewalls. They are

1. packet filter
2. Application level gateways
3. Circuit level gateways.



Packet filtering router:

→ A packet filtering router applies set of rules to each incoming and outgoing IP packets. and then forwards or discards the packets.

→ The router is typically configured to filter packets going in both direction.

→ filtering rules are based on information contained in a network packet:

1. Source IP address:
   The IP address of the system that originated the IP packet

2. Destination IP address:
   The IP address of the system the IP packet is trying to reach.

3. Source and destination transport level address:
   The transport level (eg: TCP/UDP) port number which defines applications such as SNMP or TELNET

4. IP protocol field:
   Defines the transport protocol

5. Interface:
   for a router with 3 or more ports, which interface of the router the came from or which interface of the router the packet is destined for.

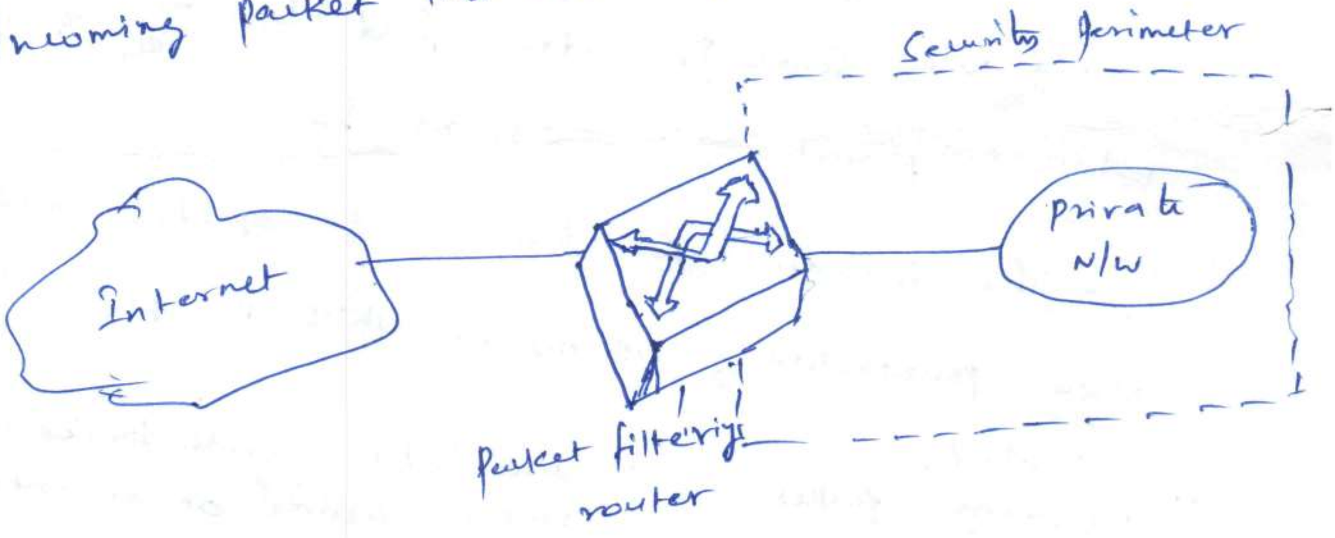
→ The packet filter is applying the listed rules on the incoming IP packet, if any rule matches, that rule is invoked to determine whether to transfer or to discard the packet.

→ If any rule is not matched then the default action is taken.

→ There are two default action policies are possible

① Default action = discard: This will not allow all the incoming packet into Corporate N/w

② Default action = forward: This will allow all the incoming packet into the Corporate N/w.



packet filtering router

Advantages:

① Simplicity: user need not to be aware of a packet filter at all

② Very fast: In operating speed point of view, it is very fast

③ Transparency: Packet filter are transparent to the user.

Disadvantages

1. Lack of support for authentication
2. Difficult in setting up the packet filter rules correctly.

Attacks on packet filtering and the Countermeasures:

The attacks and counter measures on the packet filtering router are

1. **IP address spoofing:**

Attack:

→ The intruder transmits the packet from the outside world with Source IP address field consisting of an address which is present in the internal host

→ The intruder hope that use of spoofed address will allow penetration of systems in which the packets are accepted.

Countermeasure: packet is to be discarded with inside source address if the packet arrives ~~on~~ an external interface

2. **Source routing attack:**

Attack:

→ Attackers specifies the route the packet should move along the internet.

→ ~~By~~ The attacker hopes that by specifying this option the packet filter can ~~bypass~~ bypass its normal check.

Countermeasure:

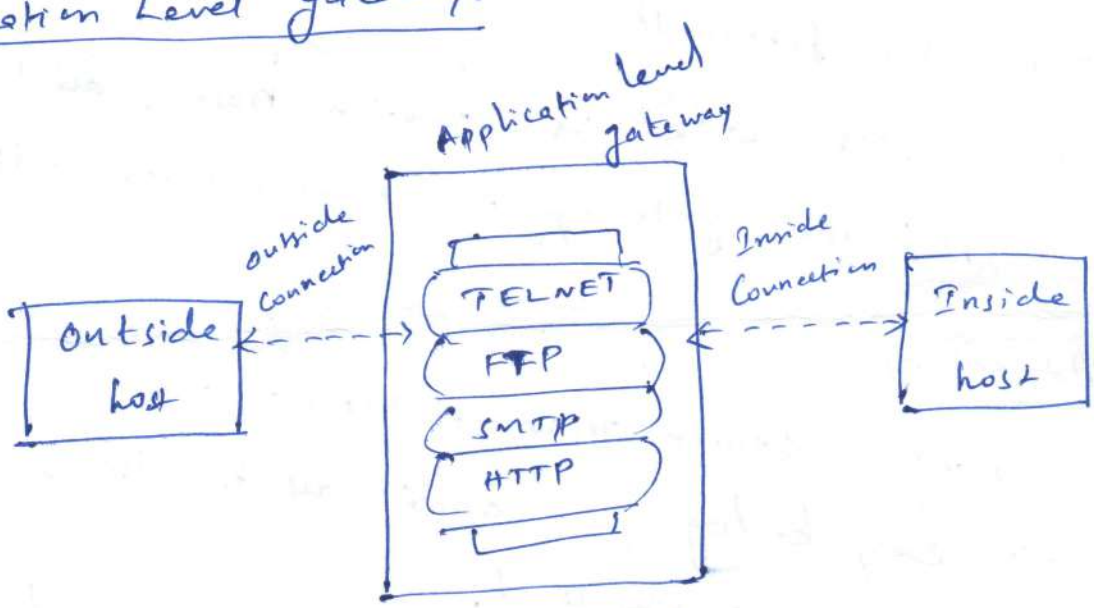
packet filter will discard all the packets that uses this option.

③ Tiny fragment attack:

→ Intruder uses the IP fragment option to create extremely small fragments and place the TCP header information into seperate ~~packets~~ fragment

→ All the N/w are having a predefined maximum frame size. When the data frame size is more than that then sender will fragment the data into small fragments.

Application Level gateway:



Application level gateway

→ It is also called as the proxy server.

→ In this the user contacts the gateway for using an application

→ when the request is coming from user the application gateway will ask for the user ID and the authentication

→ when both are correct the gateway will provide the requested service to the user.

Ex: When the user is willing to use TCP/IP application like telnet, FTP or HTTP then user contacts the application gateway

→ The gateway asks for the name of the remote host to be accessed.

→ The user responds to gateway by giving the valid user ID and authentication information.

→ The gateway contacts the application on the remote host and relays TCP segments containing application data between two end points

→ When the application requested by user is not present, the service is not supported and cannot forward across the firewall

→ This is also called as bastion host. and the structure of application gateway is shown in above diagram.

Advantages:

→ More secure than packet filter

→ easy to log and audit all the incoming traffic at application level.

→ Allows few applications rather than trying to deal with numerous possible combinations.

→ Simply detects whether a user is allowed to use TCP/IP applications.

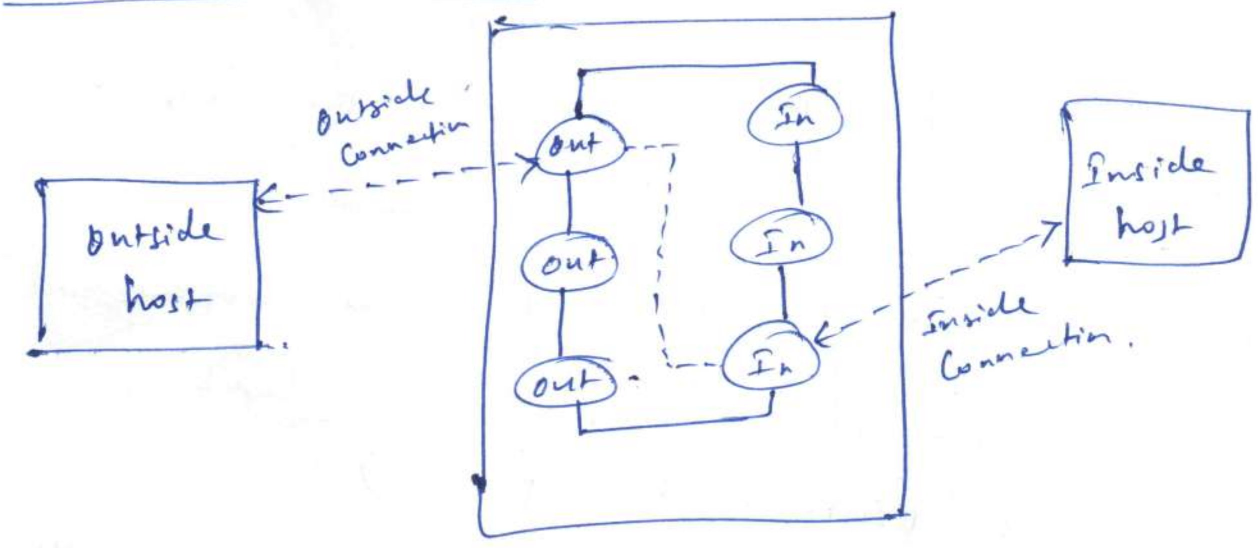
Disadvantages:

→ We find only two sets of communication: one is b/w user and application gateway and other is b/w the application gateway and the remote host.

→ Overhead in terms of connections.

→ Gateway must examine and forward all traffic in both directions.

Circuit level gateway:



Circuit level gateway

→ Circuit gateway will not provide direct end to end access b/w the user and remote host.

→ In order to provide the Connection the circuit level gateway set up two Connection one is b/w user and circuit gate and second Connection b/w the Circuit gateway and remote host

→ Once the two Connections are established, the gateway relays the segments from one connection to another without examining the Contents.

→ Arrangement of Circuit level gateway is shown in above diagram.

Advantages:

1. System administrator trusts the internal user.
2. Gateway can incur the processing overhead of examining the incoming data for forbidden functions

Disadvantages:

Circuit gateway does not look for the overhead on the outgoing data

Limitations of firewalls:

① firewall Cannot protect the Corporate n/w against the attack those bypass the firewall.

②. firewall will not protect the Corporate n/w when the Corporate n/w is having dial out Capability to Connect to ISP. (Internet service provider)

③ firewall doesnot protect against the attacks when an internal LAN may support modem pool that provides dial in Capability for traveling employees and telecommuters

④ firewall does not protect against internal threats such as disgruntled (unsatisfied) employee (or) employee who unwittingly (not aware of full facts) Co-operates with an external attackers.

⑤ firewall Cannot protect against the transfer of virus infected programs or files.

Socks:

→ Socks package is an example of a circuit level gateway.

→ ~~Socks package~~

Trusted System:

→ The applicable requirement to protect the data or resources is on the basis of levels of security

→ This is Commonly found in the millitary where information is categorized as unclassified (U), Confidential (C), Secret (S) top secret (TS) or beyond

→ In the sameway different organization also store their informations in different Categories and users are allowed to access Certain Categories of data.

→ We are Storing the information in different levels, so that this requirement is referred as multilevel security.

→ In multilevel the data in the highest level will not be given to the sublevels.

→ The highest level Convey the information to the sublevel only when there is an authorized user for accessing the data.

→ A multilevel System must enforce the following rules.

1. No read up:

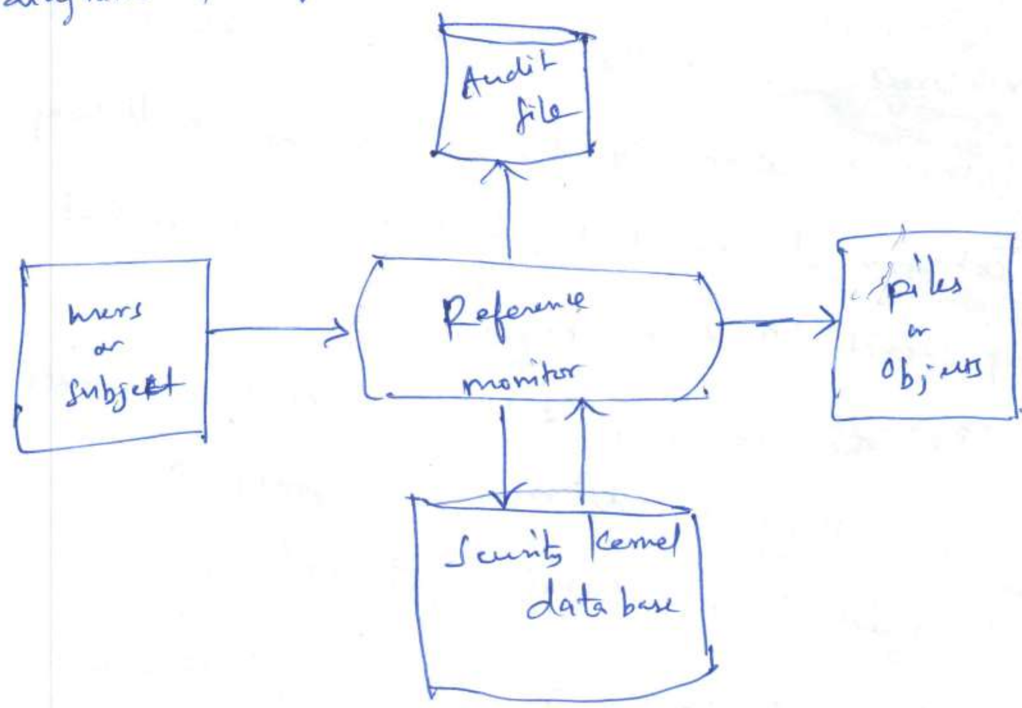
A subject can only read an object of less or equal security level. That means the user can read the files of lesser than or equal to security level. This is referred as Simple security property

2. No written down:

A subject can only write into an object greater or equal security level. That means user can write content on to the files which is having greater or equal to security level. This is referred as *-Property (star property)

→ When we enforce these two rules then the multilevel security is provided.

→ The diagram of reference monitoring system is shown below



Reference monitoring Concept

→ The reference monitoring system mainly consist of five blocks

They are
① Reference monitor
② Security kernal database
③ Audit files
④ Subject
⑤ Objects.

① Reference monitor:

→ The reference monitor is a controlling element in the H/w and OS of a computer

→. It controls all the access of subject to objects on the basis of security parameters of the subject and object.

→ When the subject sends a request for accessing an object the reference monitor refers the security kernel database for the access permission of both subject and object.

→ The reference monitor enforce the security rules (no read up, no write down) and ~~the~~ has the following properties.

① Complete mediation:

The security rules are enforced on every access, not just for one time.

Ex: When a file is opened for five times then reference monitor will apply the security rules for five times

② Isolation:

Reference monitor and security kernel database are protected from unauthorized modification

③ Verifiability:

Reference monitor correctness must be provable. That is it must be possible to demonstrate mathematically the reference monitor enforce the security rules and provides complete mediation and isolation.

② Security kernel database:

→ Security kernel database is a data store consisting of the access matrix that contains all the accessing permission on the existing objects or files

→ The permission given to the users on the files are read, write and execute

→ When user sends a request to reference monitor, it refers security kernel for access permission.

→ Based on that reference monitor will decide whether to allow the user to access the file or not.

	Program 1	....	Segment A	Segment B
Process 1	Read Execute		Read write	Read.
Process 2				
⋮				

Access matrix.

③ **Subject:**

Subject represents the users who are using the system

④ **Object:**

The object represents files on the system

⑤ **Audit file:**

→ All important security events such as security violation and authorized changes to security kernal databases are stored in audit file

→ Whenever reference monitor finds any unauthorized access, security rule violation done by the user, immediately that is to be stored on the audit file.

→ Audit file is data store consists of violation rules made by the user or files or objects.

Important Questions

① a) What are the limitations of firewall?

b) Write short notes on intruders

② a) Explain the intrusion detection tool audit records

b) What are the services provided by firewall?

③ a) How does a trusted system defend from Trojan horse attack?

b) Explain about packet filtering firewall.

④ a) List the characteristics of a good firewall implementation

b) Define reference monitor. What is the difference between subject and object of access control.

⑤ a) Discuss in detail about firewalls

b) What is meant by password management?

[illegible]

(a) What are the [illegible]

[illegible]

Explain the [illegible]

(b) What are the [illegible]

[illegible]

(c) [illegible]

Explain [illegible]

[illegible]