NETWORK SECURITY

- **UNIT 1: Introduction and symmetric Ciphers:** Pg 1
  - Introduction
  - Services
  - Mechanisms
  - Attacks
  - OSI security architecture
  - Model for network security
  - Symmetric cipher model
  - Substitution Techniques
  - Transposition Techniques
  - SLE: Steganography and program on multiplicative inverse of modulus

- **UNIT 2: Block Cipher and Encryption standards:** Pg 41
  - Simplified DES
  - Data Encryption Standard (DES)
  - Strength of DES
  - Block cipher design principles and modes of operation
  - The AES cipher (overview)
  - SLE: Block cipher principles, finite fields

- **UNIT 3: Public-key Encryption and Hash Functions:** Pg ??
  - Principles of Public-key cryptosystems
  - The RSA algorithm
  - Key Management:
    - Symmetric Key Distribution Using Asymmetric Encryption
    - Distribution of Public keys
    - Diffie-Hellman key Exchange
    - Elliptic curve cryptography (ECC)

Applications of cryptographic Hash Functions
Message Authentication Functions
SLE: program on a^b modulus n.

- UNIT 4: Digital Signatures and Authentication Protocols:
  Digital Signatures
  Elgamal digital signature scheme
  Digital Signature Standard
  Web security consideration
  Security Socket Layer (SSL) and transport layer security
  Secure electronic transaction
  Wireless Network threats
  Wireless Security Measures
  Mobile Device Security threats
  Mobile Device Security strategy.
  SLE: Cloud Security Risks and countermeasures

- UNIT 5: Intruders and Malicious Software:
  Intruders
  Intrusion Detection
  Password Management
  Types of malicious software
  Viruses
  Virus countermeasures
  SLE: Distributed intrusion detection, Behaviour-Blocking software

classmate
Date
Page 6

- **UNIT 6: Fire Walls:**

  The need for Firewalls
  Firewall characteristics
  Types of Firewalls
  SLE: Firewall configuration.

TEXTBOOK:

William stalling, "Cryptography and Network Security", Pearson Education, 4th Edition, 2011.

UNIT - 1

Introduction and Symmetric Ciphers

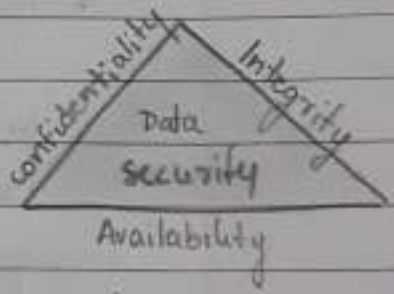
- **Introduction:**

★ **Computer security**

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability and confidentiality of information system resources (includes hardware, software, firmware, information/data and telecommunication).

★ **CIA triad**

The three concepts embody the fundamental security objectives for both data & information and computing services.



- **Confidentiality:**

1. Data confidentiality: Assures the private or confidential information is not made available or disclosed to unauthorized individuals.
2. Privacy: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information can be disclosed.

- **Integrity:**

1. Data Integrity: Assures that information and programs are changed only in a specific and authorized manner.
2. System Integrity: Assures that a system performs its intended functions in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

- Availability:
  Assures that systems work promptly and services is not denied to authorized users.

Additional Security Objectives are:

- Authenticity:
  The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message or message originator. This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source.
- Accountability:
  The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes.

- ★ We use three levels of impact on organizations or individuals when there is breach of security:
  - Low: limited adverse effect on organizational operations and assets of individuals.
  - Medium: serious adverse effect on organizational operations and assets or individuals.
  - High: severe or catastrophic adverse effect on organizational operations and assets or individuals.

The OSI Security Architecture

The OSI security architecture focuses on:

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

NOTE:

Threat: It is a possible danger that might exploit a vulnerability.
Attack: A deliberate attempt to evade security services and violate the security policy of a system.

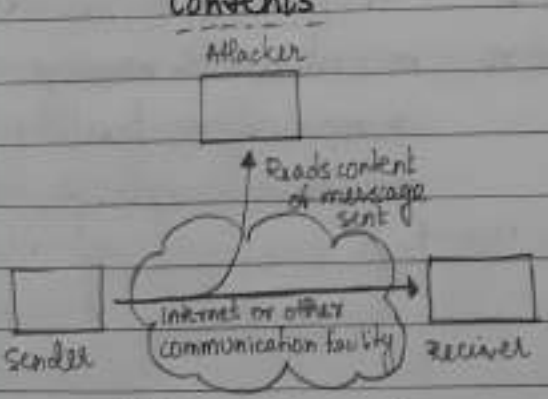
Security Attacks:

The security attacks are classified into two types:

1. Passive Attacks:

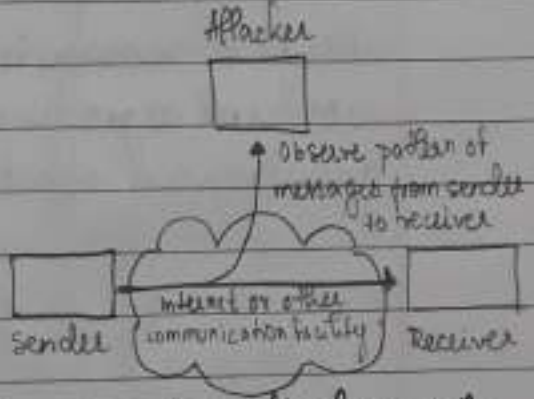
A passive attack attempts to learn or make use of information from the system but does not affect system resources. The goal of the opponent is to obtain information that is being transmitted. The two types of passive attacks are:

a. Release of Message Contents



We would like to prevent an opponent from learning the contents of transmissions like email, convo etc.

b. Traffic Analysis



An opponent might observe the pattern of messages like frequency and length of messages.

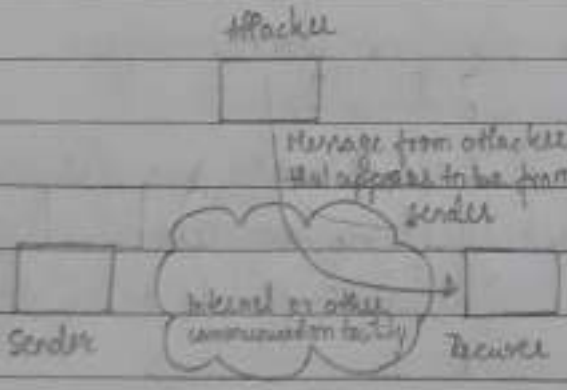
Passive attacks are very difficult to detect as it does not involve any alteration of data. It is feasible to prevent the success of these attacks by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

2. Active Attacks:

An active attack attempts to alter system resources or affect their operation. Thus it involves some modification of the data stream or the creation of a false stream. They are classified as:

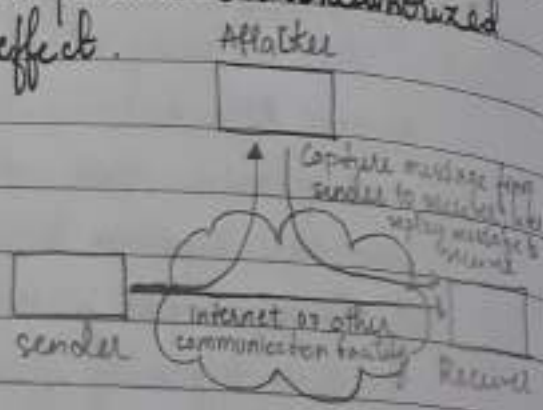
a. Masquerade

It takes place when one entity pretends to be a different entity.



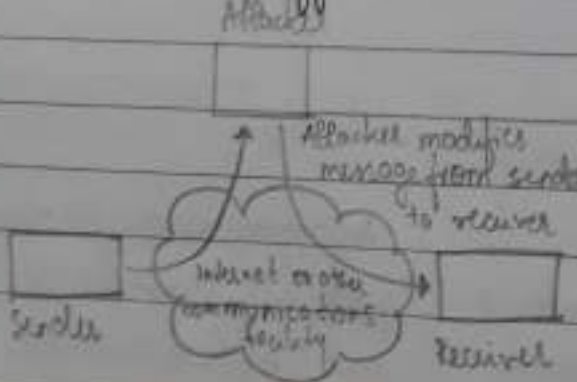
b. Replay

It involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.



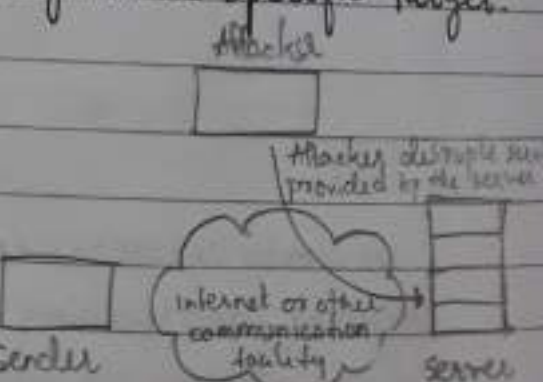
c. Modification of Messages

Some portion of the message is altered or message is delayed or reordered to produce an unauthorized effect.



d. Denial of Service

It prevents or inhibits the normal use or management of communications facility. This attack may have a specific target.



Active attacks can be detected but quite difficult to prevent them absolutely because of wide variety of potential physical, software and network vulnerabilities. The goal is to detect and to recover from any disruption or delays caused by them.

Passive Attacks	Active Attacks
• It tries to read or make use of information from the system • Modification in the information does not occur. • Doesnot cause any harm to the system. • Threat to confidentiality. • The entity is unaware of the attack. • Detection is difficult and prevention can be done by encryption. Thus emphasis on prevention.	• It tries to change the system resources or affect their operation • Modification in the information occurs. • Always causes damage to the system. • Threat to integrity and availability • The entity is aware of the attack. • Detection is possible but prevention is quite difficult. Thus emphasis on detection. which might contribute to prevention.

- Security Services:

The security services are divided into five categories and fourteen specific services:

1. Authentication:

The assurance that the communicating entity is the one that it claims to be. The service must assure that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties for the purposes of unauthorized transmission or reception.

Two specific authentication services are defined:

a. Peer entity authentication:

Used in association with a logical connection to provide confidence in the identity of the entities connected.

b. Data Origin authentication:

In a connectionless transfer, provides assurance that the source of received data is as claimed.

2. Access control:

The prevention of unauthorized use of a resource ie, this service controls who can have access to a resource, under what conditions access can occur and what those accessing the resource are allowed to do.

3. Data Confidentiality:

The protection of data from unauthorized disclosure.

a. Connection Confidentiality:

The protection of all user data on a connection

b. Connectionless Confidentiality:

The protection of all user data in a single data block.

c. Selective-Field Confidentiality:

The confidentiality of selected fields within the user data on a connection or in a single data block.

d. Traffic-Flow confidentiality:

The protection of the information that might be derived from observation of traffic flows.

4. Data Integrity:

The assurance that data received are exactly as sent by an authorized entity ie., contain no modification, insertion, deletion, or replay.

a. Connection Integrity with Recovery:

Provides the integrity of all user data on a

connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.

b. Connection Integrity without Recovery:

Provides the integrity of all user data on a connection and detects any modification, insertion, deletion or replay of any data within an entire data sequence, but provides only detection without recovery.

c. Selective-Field Connection Integrity:

Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted or replayed.

d. Connectionless Integrity:

Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.

e. Selective-Field Connectionless Integrity:

Provides the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.

5. Non-Repudiation:

Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication

a. Nonrepudiation, Origin

Proof that the message was sent by the specified party.

b. Nonrepudiation, Destination:
Proof that the message was received by the specified party.

- Security Mechanisms:
The mechanisms are divided into those that are implemented in a specific protocol layer such as TCP or an application-layer protocol and those that are not specific to any particular protocol layer or security service.

1. Specific Security Mechanisms:
May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

a. Encipherment:
The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.

b. Digital Signature:
Data appended to or a cryptographic transformation of a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protection against forgery.

c. Access Control:
A variety of mechanisms that enforce access rights to resources.

d. Data Integrity:
A variety of mechanisms used to assure the integrity of a data unit or stream of data units.

e. Authentication Exchange:
A mechanism intended to ensure the identity of an entity by means of information exchange

f. Traffic Padding:

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

g. Routing control:

Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.

h. Notarization:

The use of a trusted third party to assure certain properties of a data exchange.

2. Pervasive Security Mechanisms:

Mechanisms that are not specific to any particular OSI security service or protocol layer.

a. Trusted Functionality:

That which is perceived to be correct with respect to some criteria.

b. Security Label:

The marking bound to a resource that names or designates the security attributes of that resource.

c. Event Detection:

Detection of security-relevant events.

d. Security Audit Trail:

Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.

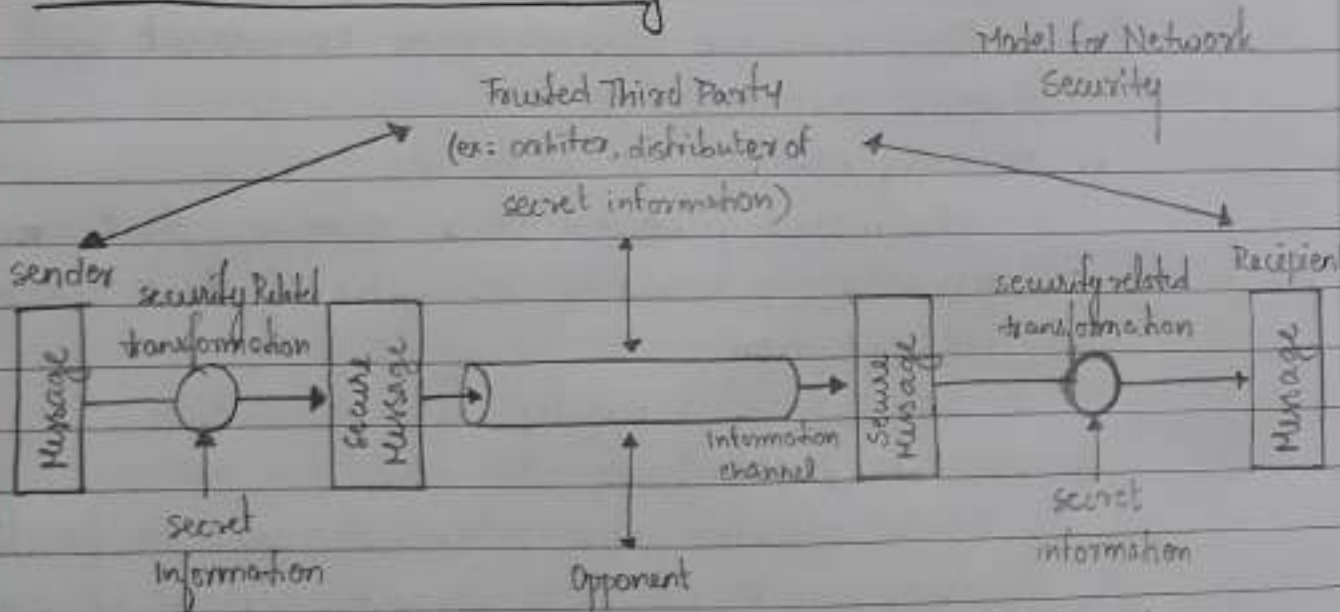
e. Security Recovery:

Deals with requests from mechanisms, such as event handling and management functions and takes recovery actions.

- Relationship between security services and Mechanisms

SERVICE	MECHANISM							
	Encipherment	Digital signature	Access control	Data Integrity	Authentication Exchange	Traffic Padding	Routing control	notarization
Peer Entity Authentication	✓	✓			✓			
Data Origin Authentication	✓	✓						
Access Control			✓					
Confidentiality	✓						✓	
Traffic flow confidentiality	✓					✓	✓	
Data Integrity	✓	✓		✓				
Non-repudiation		✓		✓				✓
Availability				✓	✓			

- **A Model for Network security:**



- A message is transferred from one party to another across some sort of internet service.
- The two parties of the transaction must cooperate for the exchange to take place.
- A logical information channel is established by defining a route through the internet from source to destination and by the cooperative use of communication protocols.

- Security is necessary or desirable to protect the information transmission from an opponent. It requires two components:
  - A security related transformation on the information to be sent. Ex: Encryption of the message
  - Some secret information shared by the two and is hoped to be unknown to the opponent. Ex: encryption key to unscramble the message at reception.
- A trusted third party may be needed to achieve secure transmission. Ex: They may be responsible for the distribution of secret information to the two principals while keeping it from any opponent. Also to arbitrate disputes between the two principals concerning the authenticity of a message transmission.

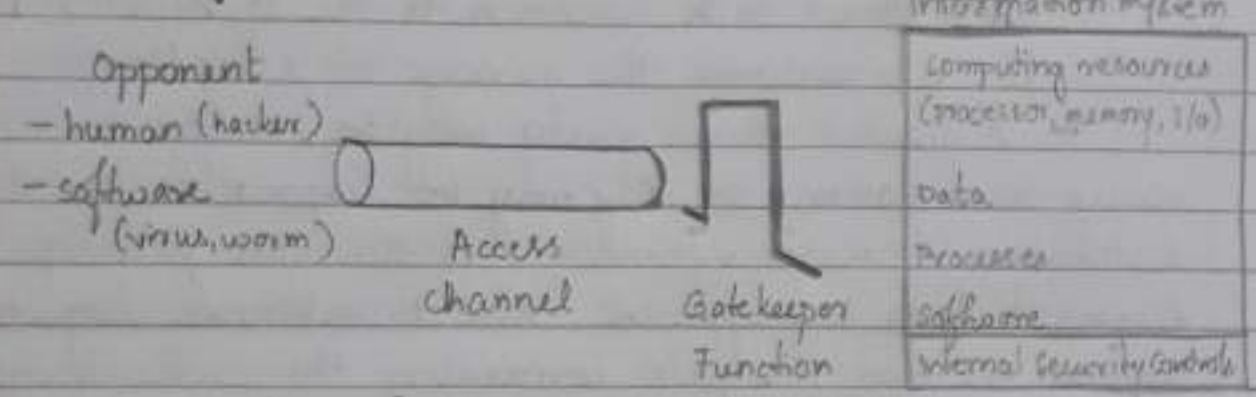
This model shows four basic tasks in designing a particular security system:

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

Another type of unwanted access that exploits vulnerabilities in the system and can affect application programs as well as utility programs (such as editors and compilers) are threats. The two kinds of threats are:

a. Information access threats:
Intercept or modify data on behalf of users who should not have access to that data.

b. Service threats:
Exploit service flaws in computers to inhibit use by legitimate users.



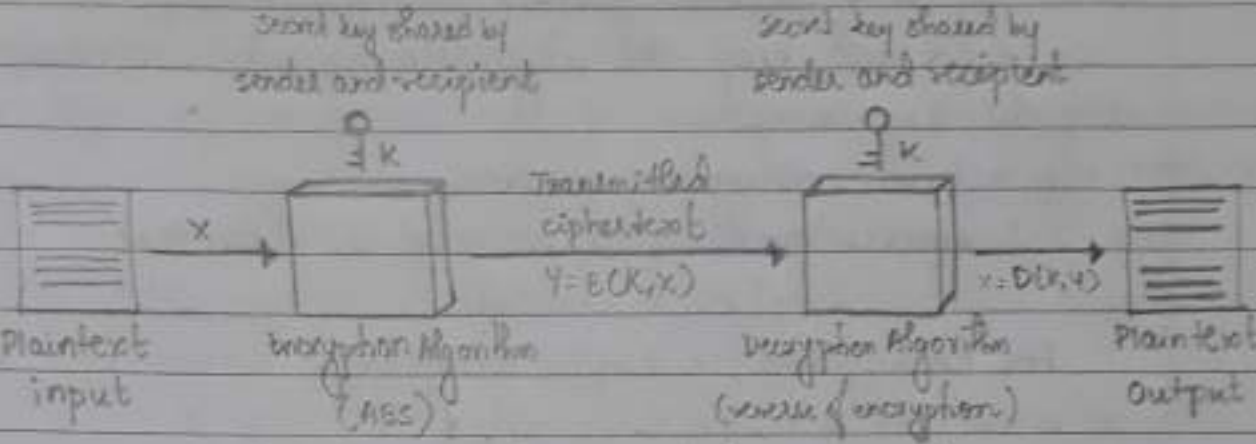
Network Access Security Model

Software attacks such as viruses and worms can be introduced to the system by means of a disk that contains the unwanted logic concealed in useful software. They can also be inserted into a system across a network.

The network security mechanism needed to cope with unwanted access are of two broad categories:

1. Gatekeeper Function which includes password based login procedures that are designed to deny access to all but authorized users and screening logic that is designed to detect and reject worms, viruses and other similar attacks.
2. Second line of defense consists of a variety of internal controls that monitor activity and analyze stored information in an attempt to detect the presence of unwanted intruders.

Symmetric Cipher Model



Simplified Model of Symmetric Encryption

- A symmetric encryption scheme has five ingredients:
  - Plaintext: This is the original intelligible message or data that is fed into the algorithm as input.
  - Encryption Algorithm: The encryption algorithm performs various substitution and transformations on the plaintext.
  - secret key: The secret key is also an input to the encryption algorithm. It is a value independent of the plaintext and of the algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.
  - Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the secret key. The ciphertext is an apparently random stream of data and is unintelligible.
  - Decryption Algorithm: This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.
- There are two requirements for secure use of conventional encryption:
  1. We need a strong encryption algorithm (The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext)

2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. We need not keep the algorithm secret, we need to keep only the key secret.

- Symmetric encryption scheme.

A source produces a message in plain text

$$X = [X_1, X_2 \cdots X_M]$$

For encryption, a key is generated

$$K = [K_1, K_2 \cdots K_J]$$

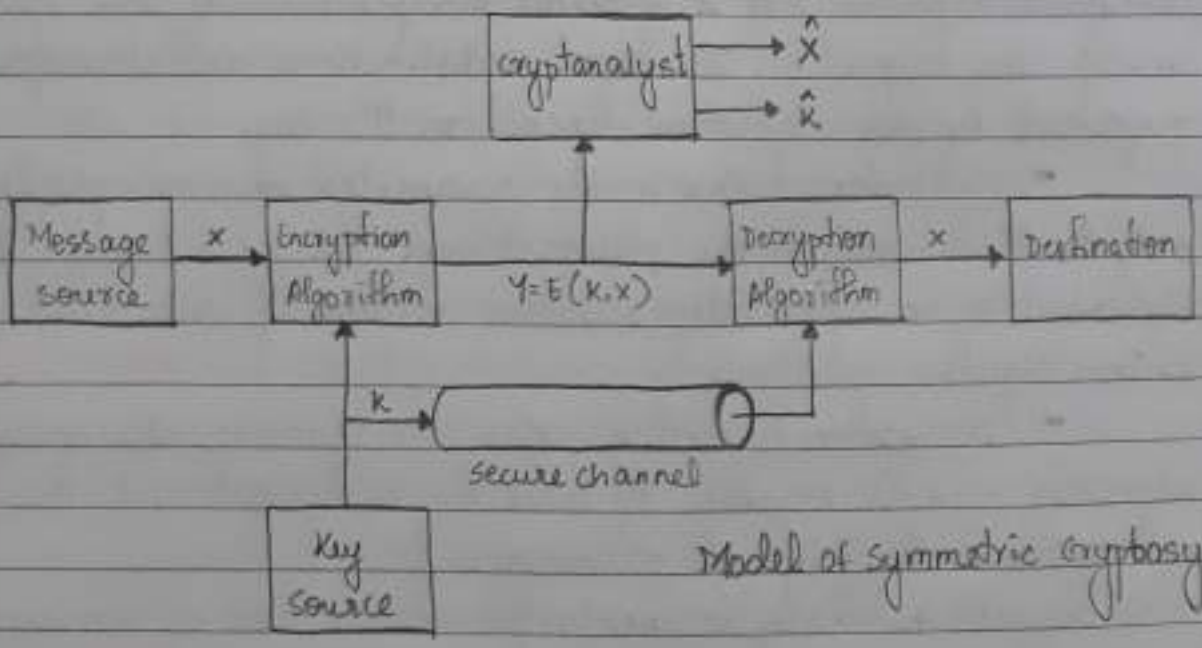
with the message X and the encryption key K as input, the encryption algorithm forms the ciphertext

$$Y = [Y_1, Y_2 \cdots Y_N]$$

i.e., $Y = E(K, X)$

The intended receiver, in possession of the key, is able to invert the transformation:

$$X = D(K, Y)$$



Model of symmetric cryptosystem

- Cryptography:

Cryptographic systems are characterized along three independent dimensions:

1. Types of operations used for transforming plaintext to ciphertext.
   - Substitution: each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element.
   - Transposition: 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 as symmetric, single-key, secret-key or conventional encryption.
   - If the sender and receiver use different keys, the system is referred as asymmetric, two-key or public-key encryption.
3. The way in which the plain text is processed:
   - Block cipher: Processes the input one block of elements at a time producing an output block for each input block.
   - Stream cipher: Processes the input elements continuously producing output one element at a time.

- Cryptanalysis and Brute-Force Attack:

The objective of attacking an encryption system is to recover the key in use rather than simply to recover the plaintext of a single ciphertext. The two general approaches to attacking a conventional encryption scheme are:

- Cryptanalysis: It relies on the nature of the algorithm and some knowledge of the plaintext or sample plaintext-ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.

- Brute-Force Attack: The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

— Types of attacks on Encrypted Messages:

Type of Attack	Known to Cryptanalyst
Ciphertext only	• Encryption Algorithm • Ciphertext
Known Plaintext	• Encryption Algorithm • Ciphertext • One or more ciphertext-plaintext pairs formed with the secret key
Chosen Plaintext	• Encryption Algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	• Encryption Algorithm • Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen text	• Encryption Algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key. • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key.

- There is no encryption algorithm that is unconditionally secure. Therefore all the users of an encryption algorithm can strive for is an algorithm that meets one or both of the following criteria: (Encryption scheme is said to be computationally secure if :)
  - The cost of breaking the cipher exceeds the value of the encrypted information.
  - The time required to break the cipher exceeds the useful life time of the information.

- **Substitution Techniques:**

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

1. **Caesar cipher** (Additive cipher)

The caesar cipher involves replacing each letter of the alphabet with the letter standing three places further down the alphabet

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

In general: $C = E(k, p) = (p+k) \bmod 26$

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

C: cipher text
p: plain text
k: key (1 to 25)

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

For caesar cipher a brute-force cryptanalysis can be easily performed by simply trying all the 25 possible keys.

The following three important characteristics of this problem enabled us to use a brute-force cryptanalysis:

1. The encryption and decryption algorithms are known
2. There are only 25 keys to try
3. The language of the plaintext is known and easily recognizable.

Caesar Cipher:

Q: Obtain the plaintext for the following cipher text:

1. Ciphertext: PHHW
   key: 13
   — wkt: $p = D(k,C) = (C-k) \bmod 26$
   here $C = \underset{15}{P}\ \underset{7}{H}\ \underset{7}{H}\ \underset{22}{W}$
   P: $p = (15-13) \bmod 26 = 2 : C$
   H: $p = (7-13) \bmod 26 = -6+26 = 20 : U$
   W: $p = (22-13) \bmod 26 = 9 : j$
   Therefore plaintext p: cuuj

2. Ciphertext: PHHW
   key: 18
   — wkt: $p = D(k,C) = (C-k) \bmod 26$
   here $C = \underset{15}{P}\ \underset{7}{H}\ \underset{7}{H}\ \underset{22}{W}$
   P: $p = (15-18) \bmod 26 = -3+26 = 23 : x$
   H: $p = (7-18) \bmod 26 = -11+26 = 15 : p$
   W: $p = (22-18) \bmod 26 = 4 : e$
   Therefore plaintext p: xppe

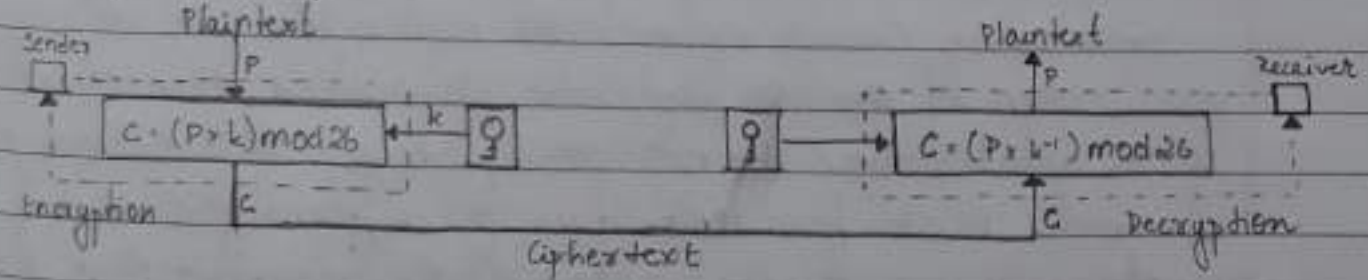
Q: Obtain the cipher text for the following plaintext:

1. Plaintext: ytrf
   key: 24
   — wkt: $C = E(k,p) = (p+k) \bmod 26$
   here $p: \underset{24}{y}\ \underset{19}{t}\ \underset{11}{l}\ \underset{5}{f}$
   y: $C = (24+24) \bmod 26 : 48-26 = 22 : W$
   t: $C = (19+24) \bmod 26 : 43-26 = 17 : R$
   l: $C = (11+24) \bmod 26 : 35-26 = 9 : J$
   f: $C = (5+24) \bmod 26 : 29-26 = 3 : D$
   Therefore ciphertext C = WRJD

2. Plaintext : cxpj
key : 20
— wkt $C = E(k,p) = (p+k) \bmod 26$
here $p = \overset{2}{c}\ \overset{23}{x}\ \overset{15}{p}\ \overset{9}{j}$
c : $C = (2+20) \bmod 26 = 22 : W$
x : $C = (23+20) \bmod 26 = 43 - 26 = 17 : R$
p : $C = (15+20) \bmod 26 = 35 - 26 = 9 : J$
j : $C = (9+20) \bmod 26 = 29 - 26 = 3 : D$
Therefore ciphertext C = WRJD

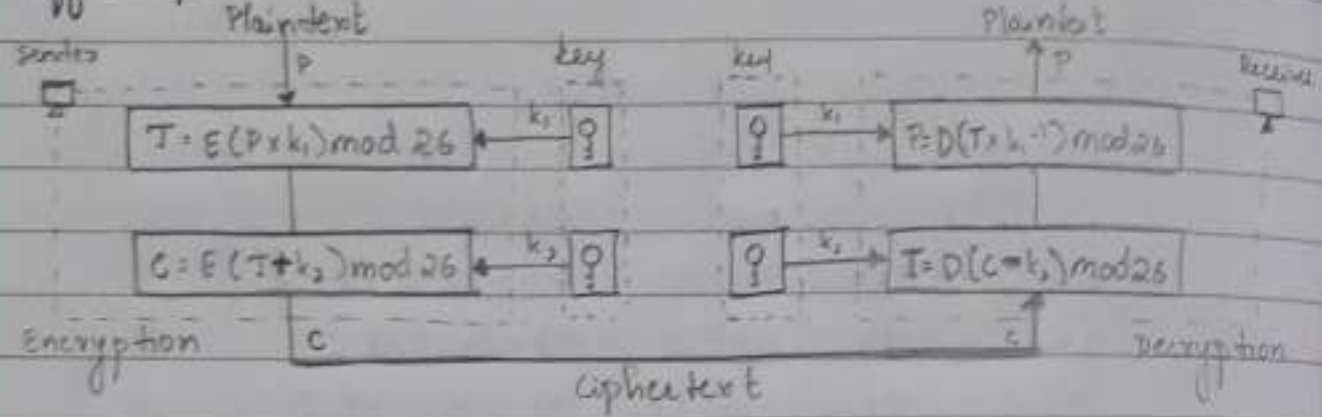
3. Plaintext: work is worship
— wkt $C = E(k,p) = (p+k) \bmod 26$
here key : k = 3
$p = \overset{22\ 14\ 17\ 10}{work}\ \overset{8\ 18}{is}\ \overset{22\ 14\ 17\ 7\ 8\ 15}{worship}$
w: $C = (22+3) \bmod 26 = 25 : Z$
o: $C = (14+3) \bmod 26 = 17 : R$
r: $C = (17+3) \bmod 26 = 20 : U$
k: $C = (10+3) \bmod 26 = 13 : N$
i: $C = (8+3) \bmod 26 = 11 : L$
s: $C = (18+3) \bmod 26 = 21 : V$
h: $C = (7+3) \bmod 26 = 10 : K$
p: $C = (15+3) \bmod 26 = 18 : S$
Therefore ciphertext C: ZRUN LV ZRUVKLS

Multiplicative ciphers:



Algorithm: Encryption : $C = E(P \times k) \bmod 26$
Decryption : $P = D(P \times k^{-1}) \bmod 26$

Affine ciphers:



Algorithm: Encryption: $C = E(P \times k_1 + k_2) \bmod 26$

Decryption: $P = D((C - k_2) \times k_1^{-1}) \bmod 26$

where k_1^{-1} is the multiplicative inverse of k_1 and $-k_2$ is the additive inverse of k_2

Because additive, multiplicative and affine ciphers have small key domains, they are very vulnerable to brute-force attack

2. Monoalphabetic Ciphers:

Caesar cipher is far from secure with only 25 possible keys A dramatic increase in the key space can be achieved by allowing an arbitrary substitution. A permutation of a finite set of elements S is an ordered sequence of all the elements of S, with each element appearing exactly once.

Ex: $S = \{a, b, c\}$

permutation: abc, acb, bac, bca, cab, cba (3!)

In general, there are n! permutations of a set of n elements considering Caesar cipher, the cipher line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4×10^{26} possible keys. This seems to eliminate brute-force techniques for cryptanalysis. But if the cryptanalyst knows the nature of the plaintext then the analyst can exploit the regularities of the language.

It is called as monoalphabetic substitution cipher because a single cipher alphabet (mapping from plain alphabet to cipher alphabet) is used per message.

Monoalphabetic ciphers are easy to break because the reflect the frequency data of the original alphabet.

A powerful tool is to look at frequency of letter combinations.

diagrams: two-letter combinations

trigrams: three-letter combinations.

A countermeasure is to provide multiple substitutes known as homophones, for a single letter.

Homophones

Homophonic substitution cipher is where a single plaintext letter can be replaced by any of several different ciphertext letters. They are generally much more difficult to break than standard substitution methods.

Ex: Plaintext: 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

ciphertext: D X S F Z E H C V 1 T P G A Q L K J R U O W M Y B N

9 7 3 5 0 4 6

2

1

To Encrypt the message: DEFEND THE WEST WALL

here E can be Z, 7, 2 or 1 at random

N can be A or 5 at random

T can be U or 6 at random

S can be R or 4 at random

A can be D or 9 at random

Plaintext: DEFEND THE WEST WALL

ciphertext: F7EZ5F UC2 M1R6 M9PP

The number of ciphertext letters assigned to each plaintext letter is chosen to flatten the frequency distribution as much as possible.

3. Playfair cipher:

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

The playfair algorithm is based on the use of 5×5 matrix of letters constructed using a keyword. The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom and then filling in the remainder of the matrix with the remaining letters in alphabetic order. The letters I and J count as one letter.

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

If a key word has repeating letters, use those letters only once.

Ex: keyword: "better"

⇒ betr

The plaintext is encrypted two letters at a time, according to the following rules:

1. Repeating plaintext letters that are in the same pair are separated with a filler letter such as x.

   balloon ⇒ ba lx lo on

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

3. Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last.

   mu ⇒ CM

4. Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column

occupied by the other plaintext letter
hs ⇒ BP and ea ⇒ IM or JM

Security Features

- Playfair cipher is a great advance over simple monoalphabetic ciphers.
- There are only 26 letters whereas there are 26×26 = 676 diagrams so that the identification of individual diagrams is more difficult.
- Frequency analysis is more difficult compared to single letter substitution.
- Despite this level of security, Playfair cipher leaves much of structure of the plaintext language intact making it relatively easy to break.

Playfair cipher

Q: Obtain the cipher text for the following:

1. keyword: computer
plaintext: parrot

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

— parrot → pa rx ro tx
ciphertext: AH GM EM RV

2. keyword: mars
plaintext: canteen

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

— canteen → ca nt ex en
ciphertext: DM LU KR GK

4. Hill Cipher:

Background:

Concepts from linear algebra

wkt $AA^{-1} = I$ where I is an identity matrix

Ex: $A = \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix}$ $|A| = \begin{vmatrix} 5 & 8 \\ 17 & 3 \end{vmatrix} = 15 - 136 = -121 \bmod 26 = 9$

$A^{-1} \bmod 26 = 9^{-1} \bmod 26 = 3$

because $9 \times 3 = 27 \bmod 26 = 1$

$A^{-1} \bmod 26 = 3\begin{pmatrix} 3 & -8 \\ -17 & 5 \end{pmatrix} = 3\begin{pmatrix} 3 & 18 \\ 9 & 5 \end{pmatrix} = \begin{pmatrix} 9 & 54 \\ 27 & 15 \end{pmatrix} = \begin{pmatrix} 9 & 2 \\ 1 & 15 \end{pmatrix}$

$AA^{-1} = \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix}\begin{pmatrix} 9 & 2 \\ 1 & 15 \end{pmatrix} = \begin{pmatrix} 53 & 130 \\ 156 & 79 \end{pmatrix} \bmod 26 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$

Hill cipher is a multiletter cipher. This encryption algorithm takes 'm' successive plaintext letters and substitutes for them with 'm' ciphertext letters. The substitution is determined by 'm' linear equations in which each character is assigned a numerical value ($a=0, b=1, \ldots, z=25$).

For $m = 3$, the system can be defined as:

$c_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26$

$c_2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26$

$c_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26$

This can be expressed in terms of row vectors and matrices

$$(c_1\ c_2\ c_3) = (p_1\ p_2\ p_3)\begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \bmod 26$$

$\rightarrow$ $\boxed{C = Kp \bmod 26}$

c: ciphertext

p: plaintext

k: encryption key

Therefore

$C = E(k, p) = Kp \bmod 26$

$P = D(k, C) = k^{-1}C \bmod 26 = kpk^{-1} = p$

Security:

- Its strength is it completely hides single letter frequencies.
- Use of a larger matrix hides more frequency information.
- A 3*3 Hill cipher hides not only single-letter but also two-letter frequency information.
- Although the Hill cipher is strong against a ciphertext only attack, it is easily broken with a known plaintext attack.

Hill cipher:

Q: Using Hill Cipher, encrypt the following:

1. Plaintext : 'security'

$key = \begin{bmatrix} 7 & 8 \\ 19 & 3 \end{bmatrix}$

— Ciphertext : $C = kp \bmod 26$

security ⇒ se cu ri ty

$\begin{bmatrix} s \\ e \end{bmatrix} = \begin{bmatrix} 18 \\ 4 \end{bmatrix} \Rightarrow \begin{bmatrix} 7 & 8 \\ 19 & 3 \end{bmatrix}\begin{bmatrix} 18 \\ 4 \end{bmatrix} = \begin{bmatrix} 158 \\ 354 \end{bmatrix} \bmod 26 = \begin{bmatrix} 2 \\ 16 \end{bmatrix} = \begin{bmatrix} C \\ Q \end{bmatrix}$

$\begin{bmatrix} c \\ u \end{bmatrix} = \begin{bmatrix} 2 \\ 20 \end{bmatrix} \Rightarrow \begin{bmatrix} 7 & 8 \\ 19 & 3 \end{bmatrix}\begin{bmatrix} 2 \\ 20 \end{bmatrix} = \begin{bmatrix} 174 \\ 98 \end{bmatrix} \bmod 26 = \begin{bmatrix} 18 \\ 20 \end{bmatrix} = \begin{bmatrix} S \\ U \end{bmatrix}$

$\begin{bmatrix} r \\ i \end{bmatrix} = \begin{bmatrix} 17 \\ 8 \end{bmatrix} \Rightarrow \begin{bmatrix} 7 & 8 \\ 19 & 3 \end{bmatrix}\begin{bmatrix} 17 \\ 8 \end{bmatrix} = \begin{bmatrix} 183 \\ 347 \end{bmatrix} \bmod 26 = \begin{bmatrix} 1 \\ 9 \end{bmatrix} = \begin{bmatrix} B \\ J \end{bmatrix}$

$\begin{bmatrix} t \\ y \end{bmatrix} = \begin{bmatrix} 19 \\ 24 \end{bmatrix} \Rightarrow \begin{bmatrix} 7 & 8 \\ 19 & 3 \end{bmatrix}\begin{bmatrix} 19 \\ 24 \end{bmatrix} = \begin{bmatrix} 325 \\ 433 \end{bmatrix} \bmod 26 = \begin{bmatrix} 13 \\ 17 \end{bmatrix} = \begin{bmatrix} N \\ R \end{bmatrix}$

Therefore ciphertext is C = CQSUBJNR

2. Plaintext : 'monday'

$Key = \begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix}$

— ciphertext : $C = pk \bmod 26$

monday → mo nd ay

$$\begin{bmatrix} m \\ o \end{bmatrix} = \begin{bmatrix} 12 \\ 14 \end{bmatrix} \Rightarrow \begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix} \begin{bmatrix} 12 \\ 14 \end{bmatrix} = \begin{bmatrix} 164 \\ 158 \end{bmatrix} \bmod 26 = \begin{bmatrix} 8 \\ 2 \end{bmatrix} = \begin{bmatrix} I \\ C \end{bmatrix}$$

$$\begin{bmatrix} n \\ d \end{bmatrix} = \begin{bmatrix} 13 \\ 3 \end{bmatrix} \Rightarrow \begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix} \begin{bmatrix} 13 \\ 3 \end{bmatrix} = \begin{bmatrix} 129 \\ 86 \end{bmatrix} \bmod 26 = \begin{bmatrix} 25 \\ 8 \end{bmatrix} = \begin{bmatrix} Z \\ I \end{bmatrix}$$

$$\begin{bmatrix} a \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 24 \end{bmatrix} \Rightarrow \begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix} \begin{bmatrix} 0 \\ 24 \end{bmatrix} = \begin{bmatrix} 96 \\ 168 \end{bmatrix} \bmod 26 = \begin{bmatrix} 18 \\ 12 \end{bmatrix} = \begin{bmatrix} S \\ M \end{bmatrix}$$

Therefore the ciphertext is C = ICZISM

Q: Using Hill cipher, decrypt the following:

1. key = $\begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix}$

ciphertext: ICZISM

→ plaintext: $p = k^{-1} c \bmod 26$

To find k^{-1}:

$$\begin{vmatrix} 9 & 4 \\ 5 & 7 \end{vmatrix} = 63 - 20 = 43 \bmod 26 = 17$$

gcd(17, 26)

17 | 26 (1
-17
9 | 17 (1
-9
8 | 9 (1
-8
1 | 8 (8
-8
0

gcd

$17^{-1} \bmod 26 = 23$

$\because 17 \times 23 \bmod 26 = 1$

$$\therefore k^{-1} = \frac{1}{17} \begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} \bmod 26$$

$$k^{-1} = 23 \begin{pmatrix} 7 & 22 \\ 21 & 9 \end{pmatrix} \bmod 26$$

$$k^{-1} = \begin{pmatrix} 161 & 506 \\ 483 & 207 \end{pmatrix} \bmod 26 = \begin{pmatrix} 5 & 12 \\ 15 & 25 \end{pmatrix}$$

$$\begin{bmatrix} I \\ C \end{bmatrix} = \begin{bmatrix} 8 \\ 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 5 & 12 \\ 15 & 25 \end{bmatrix} \begin{bmatrix} 8 \\ 2 \end{bmatrix} = \begin{bmatrix} 64 \\ 170 \end{bmatrix} \bmod 26 = \begin{bmatrix} 12 \\ 14 \end{bmatrix} = \begin{bmatrix} m \\ o \end{bmatrix}$$

$$\begin{bmatrix} Z \\ I \end{bmatrix} = \begin{bmatrix} 25 \\ 8 \end{bmatrix} \Rightarrow \begin{bmatrix} 5 & 12 \\ 15 & 25 \end{bmatrix} \begin{bmatrix} 25 \\ 8 \end{bmatrix} = \begin{bmatrix} 221 \\ 575 \end{bmatrix} \bmod 26 = \begin{bmatrix} 13 \\ 3 \end{bmatrix} = \begin{bmatrix} n \\ d \end{bmatrix}$$

$$\begin{bmatrix} S \\ M \end{bmatrix} = \begin{bmatrix} 18 \\ 12 \end{bmatrix} \Rightarrow \begin{bmatrix} 5 & 12 \\ 15 & 25 \end{bmatrix} \begin{bmatrix} 18 \\ 12 \end{bmatrix} = \begin{bmatrix} 234 \\ 570 \end{bmatrix} \bmod 26 = \begin{bmatrix} 0 \\ 24 \end{bmatrix} = \begin{bmatrix} a \\ y \end{bmatrix}$$

Therefore the plaintext is p = monday

q	x_1	x_2	x	t_1	t_2	$t = t_1 - q \times t_2$
1	26	17	9	0	1	$0-1(1)=-1$
1	17	9	8	1	-1	$1-1(-1)=2$
1	9	8	1	-1	2	$-1-1(2)=-3$
8	8	1	0	2	-3	$2-1(-3)=5$
-	1	0	-	-3	5	

↳ inverse of 17 mod 26 ⇒ $-3+26 = 23$

Extended Euclidean Algorithm

Q If plaintext = monday
ciphertext = ICZISM
find 2×2 key for the Hill cipher

— key : $K = C P^{-1} \bmod 26$

plaintext : $P = (12, 14, 13, 3, 0, 24)$: monday

ciphertext : $C = (8, 2, 25, 8, 18, 12)$: ICZISM

$$P = \begin{vmatrix} 12 & 13 \\ 14 & 3 \end{vmatrix} = 36 - 182 = -146 \bmod 26 = 10$$

$$\text{corresponding } C = \begin{pmatrix} 8 & 25 \\ 2 & 8 \end{pmatrix}$$

$$\text{key} = K = \frac{1}{10}\begin{pmatrix} 8 & 25 \\ 2 & 8 \end{pmatrix}\begin{pmatrix} 3 & -13 \\ -14 & 12 \end{pmatrix} \bmod 26$$

$$K = \frac{1}{10}\begin{pmatrix} 8 & 25 \\ 2 & 8 \end{pmatrix}\begin{pmatrix} 3 & 13 \\ 12 & 12 \end{pmatrix} \bmod 26$$

$$K = \frac{1}{10}\begin{pmatrix} 324 & 404 \\ 102 & 122 \end{pmatrix} \bmod 26$$

$$K = \frac{1}{10}\begin{pmatrix} 12 & 14 \\ 24 & 18 \end{pmatrix} \bmod 26$$

Since gcd ≠ 1 extended Euclidean algorithm cannot be used to find $10^{-1} \bmod 26$

gcd (10, 26)

```
10 ) 26 ( 2
    -20
     6 ) 10 ( 1
         6
         4 ) 6 ( 1
           -4
           2 ) 4 ( 2
             -4
              0
```

gcd

Let $\frac{12}{10} = y \bmod 26 \Rightarrow \frac{10y - 12}{26}$ such that rem = 0
$\therefore y = 9$

Let $\frac{14}{10} = y \bmod 26 \Rightarrow \frac{10y - 14}{26}$ such that rem = 0
$\therefore y = 4$

Let $\frac{24}{10} = y \bmod 26 \Rightarrow \frac{10y - 24}{26}$ such that rem = 0
$\therefore y = 5$

Let $\frac{18}{10} = y \bmod 26 \Rightarrow \frac{10y - 18}{26}$ such that rem = 0
$\therefore y = 7$

Therefore, key is $k = \begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix}$ //

Q: If plaintext = friday
ciphertext = PQCFKU
find the key k.

— key: $k = Cp^{-1} \bmod 26$

plaintext: p = friday = (5, 17, 8, 3, 0, 24)

ciphertext: C = PQCFKU = (15, 16, 2, 5, 10, 20)

$$P = \begin{vmatrix} 5 & 8 \\ 17 & 3 \end{vmatrix} = 15 - 136 = -121 \bmod 26 = 9$$

Extended Euclidean Algorithm can be used because gcd = 1

q	r_2	r_1	r	t_1	t_2	$t = t_1 - qt_2$
2	26	9	8	0	1	-2(1) = -2
1	9	8	1	1	-2	1-1(-2) = 3
8	8	1	0	-2	3	-2-8(3) = -26
-	1	0	-	3	-26	

gcd(9, 26)

9) 26 (2
-18
8) 9 (1
-8
1) 8 (8
-8
0

gcd = 1

↳ inverse of 9 mod 26

$9^{-1} \bmod 26 = 3$ ∵ $9 \times 3 = 27 \bmod 26 = 27 - 26 = 1$

key: $k = 3\begin{bmatrix} 15 & 2 \\ 16 & 5 \end{bmatrix}\begin{bmatrix} 5 & 8 \\ 17 & 3 \end{bmatrix}^{-1} \bmod 26$

$k = 3\begin{bmatrix} 15 & 2 \\ 16 & 5 \end{bmatrix}\begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} \bmod 26$

$k = 3\begin{bmatrix} 15 & 2 \\ 16 & 5 \end{bmatrix}\begin{bmatrix} 3 & 18 \\ 9 & 5 \end{bmatrix} \bmod 26$

$k = 3\begin{bmatrix} 63 & 280 \\ 93 & 313 \end{bmatrix} \bmod 26$

$k = 3\begin{bmatrix} 11 & 20 \\ 15 & 1 \end{bmatrix} \bmod 26$

$k = \begin{bmatrix} 33 & 60 \\ 45 & 3 \end{bmatrix} \bmod 26 = \begin{bmatrix} 7 & 8 \\ 19 & 3 \end{bmatrix}$ is the key.

Q: If plaintext: 'hillcipher'
ciphertext: HCRZSSXNSP
Find the 2x2 key

— key : $k = C_p^{-1} \bmod 26$

plaintext p = hillcipher = (7, 8, 11, 11, 2, 8, 15, 7, 4, 17)

ciphertext C = HCRZSSXNSP = (7, 2, 17, 25, 18, 18, 23, 13, 18, 15)

$$p = \begin{vmatrix} 7 & 11 \\ 8 & 11 \end{vmatrix} = 77 - 88 = -11 \bmod 26 = 15$$

As gcd = 1 for 15 and 26, the extended Euclidean algorithm can be used

q	r_2	r_1	r	t_1	t_2	$t = t_1 - qt_2$
1	26	15	11	0	1	$-1(1) = -1$
1	15	11	4	1	-1	$1-1(-1) = 2$
2	11	4	3	-1	2	$-1-2(2) = -5$
1	4	3	1	2	-5	$2-1(-5) = 7$
3	3	1	0	-5	7	$-5-3(7) = -26$
-	1	0	-	7	-26	

↳ inverse of 15 mod 26

$15^{-1} \bmod 26 = 7$ ∵ $15 \times 7 = 105 \bmod 26 = 1$

gcd (15, 26)

15) 26 (1
- 15
11) 15 (1
- 11
4) 11 (2
- 8
3) 4 (1
- 3
1) 3 (3
- 3
0

gcd = 1

$$\text{key: } k = 7 \begin{bmatrix} 7 & 17 \\ 2 & 25 \end{bmatrix} \begin{bmatrix} 7 & 11 \\ 8 & 11 \end{bmatrix}^{-1} \bmod 26$$

$$k = 7 \begin{bmatrix} 7 & 17 \\ 2 & 25 \end{bmatrix} \begin{bmatrix} 11 & -11 \\ -8 & 7 \end{bmatrix} \bmod 26$$

$$k = 7 \begin{bmatrix} 7 & 17 \\ 2 & 25 \end{bmatrix} \begin{bmatrix} 11 & 15 \\ 18 & 7 \end{bmatrix} \bmod 26$$

$$k = 7 \begin{bmatrix} 383 & 224 \\ 472 & 205 \end{bmatrix} \bmod 26$$

$$k = 7 \begin{bmatrix} 19 & 16 \\ 4 & 23 \end{bmatrix} \bmod 26$$

$$k = \begin{bmatrix} 133 & 112 \\ 28 & 161 \end{bmatrix} \bmod 26 = \begin{bmatrix} 3 & 8 \\ 2 & 5 \end{bmatrix}$$ is the key.

Q: If plaintext : 'pay'
and key is $k = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$ Find the ciphertext

— plaintext : $p = pay = (15, 0, 24)$
ciphertext : $c = kp \bmod 26$

$$\therefore C = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix} \begin{bmatrix} 15 \\ 0 \\ 24 \end{bmatrix} \bmod 26$$

$$C = \begin{bmatrix} 375 \\ 819 \\ 486 \end{bmatrix} \bmod 26 = \begin{bmatrix} 11 \\ 13 \\ 18 \end{bmatrix} = \begin{bmatrix} L \\ N \\ S \end{bmatrix}$$

Therefore ciphertext $C = LNS$

Q: If ciphertext is LNS
and key is $k = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$

Find the plaintext.

— ciphertext : $C = LNS = (11, 13, 18)$
plaintext : $p = Ck^{-1} \bmod 26$

To find k^{-1}

$$|k| = \begin{vmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{vmatrix} = 5100 - 6069 + 30 = -989 \bmod 26$$

$$= 23$$

$-9 \bmod 26 = 17$

$\therefore 23^{-1} \bmod 26 = 17$

To find cofactor of k

$cf_{11} = (18 \times 19 - 21 \times 2) \bmod 26 = 300 \bmod 26 = 14$

$cf_{12} = -(21 \times 19 - 21 \times 2) \bmod 26 = -357 \bmod 26 = 7$

$cf_{13} = (21 \times 2 - 18 \times 2) \bmod 26 = 6$

gcd (26, 23)

23) 26 (1
-23
3) 23 (7
-21
2) 3 (1
-2
1) 2 (2
-2
0

gcd

Since gcd = 1, the extended euclidean algorithm can be used

q	r_2	r_1	r	t_1	t_2	$t = t_1 - q t_2$
1	26	23	3	0	1	-1(1) = -1
7	23	3	2	1	-1	1-7(-1) = 8
1	3	2	1	-1	8	-1-1(8) = -9
2	2	1	0	8	-9	8-2(-9) = 26
-	1	0	-	-9	26	

↳ inverse of 23 mod 26
$(-9 + 26) = 17$

$cf_{21} = -(17 \times 19 - 5 \times 2) \bmod 26 = -313 \bmod 26 = 25$
$cf_{22} = (17 \times 19 - 5 \times 2) \bmod 26 = 313 \bmod 26 = 1$
$cf_{23} = (17 \times 2 - 17 \times 2) \bmod 26 = 0$
$cf_{31} = (17 \times 21 - 5 \times 18) \bmod 26 = 267 \bmod 26 = 7$
$cf_{32} = -(17 \times 21 - 5 \times 21) \bmod 26 = -252 \bmod 26 = 8$
$cf_{33} = (17 \times 18 - 21 \times 17) \bmod 26 = -51 \bmod 26 = 1$

Therefore cofactor of k is

$$\text{cf of } k = \begin{bmatrix} 14 & 7 & 6 \\ 25 & 1 & 0 \\ 7 & 8 & 1 \end{bmatrix}$$

$$\text{Adjoint of } k = \begin{bmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{bmatrix}$$

$$\therefore k^{-1} = \frac{\text{Adj } k}{|k|} = 17 \begin{bmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{bmatrix} \bmod 26$$

$$k^{-1} = \begin{bmatrix} 238 & 425 & 119 \\ 119 & 17 & 136 \\ 102 & 0 & 17 \end{bmatrix} \bmod 26$$

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

Therefor the plaintext is:

$$P = K\vec{C} \bmod 26$$

$$P = \begin{bmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{bmatrix} \begin{bmatrix} 11 \\ 13 \\ 18 \end{bmatrix} \bmod 26$$

$$P = \begin{bmatrix} 431 \\ 494 \\ 570 \end{bmatrix} \bmod 26 = \begin{bmatrix} 15 \\ 0 \\ 24 \end{bmatrix} = \begin{bmatrix} P \\ a \\ y \end{bmatrix} \quad \therefore P = \text{'pay'}$$

NOTE:

- An encryption scheme is unconditionally secure means: the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available.
- Kerckhoff's Principle:

One should always assume that the adversary, everyone knows the encryption/decryption algorithm. The resistance of the cipher to attack must be based only on the secrecy of the key.

5. Polyalphabetic Ciphers:

Polyalphabetic ciphers use different monoalphabetic substitutions as one proceeds through the plaintext message. All the techniques have following features in common:

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

- Vigenère Cipher:

Encryption: The first letter of the key is added to first letter of the plaintext. The key is repeated until all of the plaintext sequence is encrypted.

General equation of the encryption process:

$C_i = (p_i + k_{i \bmod m}) \bmod 26$

General equation of the decryption process:

$p_i = (C_i - k_{i \bmod m}) \bmod 26$

Example:

keyword: deceptive

message: "we are discovered save yourself"

Encryption:

key: deceptivedeceptivedeceptive

plaintext: wearediscoveredsaveyourself

ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

3 mod 26 = 3 | 25 mod 26 = 0 | 28 mod 26 = 2 | 32 mod 26 = 6 | 37 mod 26 = 11 | 32 mod 26 = 6

The strength of this cipher is that there are multiple ciphertext letters for each plaintext letter, one for each unique letter of the keyword. Thus the letter frequency is obscured.

However not all knowledge of the plaintext structure is lost. An improvement is achieved over Playfair cipher but considerable frequency distribution information remains.

As it is encrypted with the same monoalphabetic ciphers one can use the known frequency characteristics of the plaintext language to attack each of the monoalphabetic ciphers separately. Thus periodic nature of the keyword can be eliminated by using a nonrepeating keyword that is as long as the message itself. This is referred to as an Autokey system.

Autokey System:

Here a keyword is concatenated with the plaintext itself to provide a running key.

Example:

key: deceptivewearediscoveredsav

plaintext: wearediscoveredsaveyourself

ciphertext: ZICVTWQNGKZEIIGASXSTSLVVWLA

36 mod 26 = 10 | 28 mod 26 = 6 | 26 mod 26 = 0 | 45 mod 26 = 19 | 37 mod 26 = 11 | 26 mod 26 = 0

Even this scheme is vulnerable to cryptanalysis because the key and the plaintext share the same frequency distribution of letters thus a statistical technique can be applied

- Vernam Cipher:

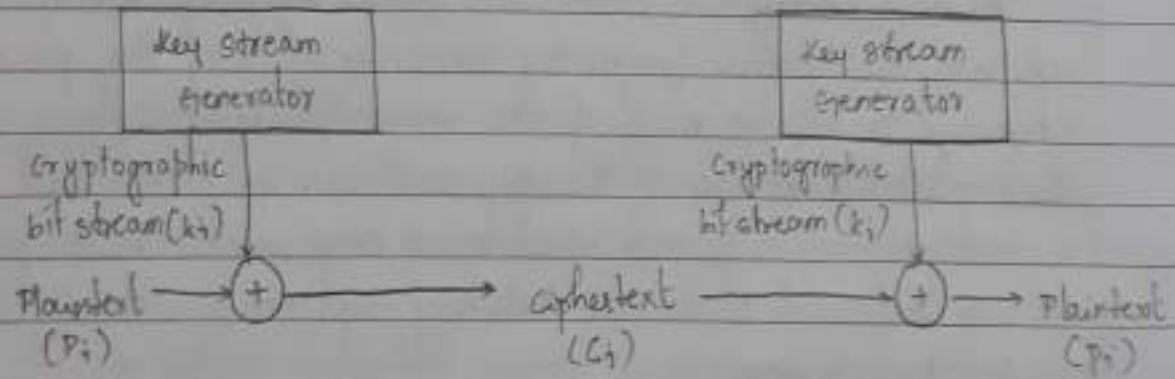
Encryption: To choose a keyword that is as long as the plaintext and has no statistical relationship to it. This system was introduced by an AT&T engineer named Gilbert Vernam in 1918. This system works on binary data (bits) rather than letters the system can be expressed as:

$$c_i = p_i \oplus k_i$$

where p_i : i^{th} binary digit of the plaintext
k_i : i^{th} binary digit of the key
c_i : i^{th} binary digit of the ciphertext
$\oplus$: Exclusive OR (XOR) operation

ciphertext is generated by performing the bit wise XOR of the plaintext and the key. Because of the properties of the XOR, decryption simply involves the same bitwise operation:

$$p_i = c_i \oplus k_i$$



Although with a long key it poses difficulty for cryptanalytic attacks, it can be broken with sufficient ciphertext, known or probable plaintext sequences or both.

6 One-time Pad:

It uses a random key that is as long as the message, so that key need not be repeated. Also, key is to be used to encrypt and decrypt a single message and then is discarded. Each new message requires a new key of the same length as the new message. Such a scheme is known as a one-time pad and is unbreakable.

It produces random output that bears no statistical relationship to the plaintext. Because the ciphertext contains no information whatsoever about the plaintext, there is no way to break the code.

Example: 27 characters with space character.

→ alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ

1. key: px lmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih
plaintext: mr mustard with the candlestick in the hall

2. key: mfugpmiyqaqazgoufnkqllmhsqdqogtewbqfgyovuhwt
plaintext: miss scarlet with the knife in the library

Security of the one time pad is entirely due to randomness of the key. The one-time pad offers complete security but in practice it has fundamental difficulties:

1. There is the practical problem of making large quantities of random keys. Any heavily used system might require millions of random characters on a regular basis. Supplying truly random characters in this volume is a significant task.
2. Another problem is key distribution and protection. For every message to be sent, a key of equal length is needed by both sender and receiver. Thus, a mammoth key distribution problem exists.

Thus it is used for low bandwidth channels requiring very high security. This is the only cryptosystem that exhibits perfect secrecy.

- **Transposition Techniques:**

A transposition cipher is where mapping is achieved by performing some sort of permutation on the plaintext letters.

1. **Rail Fence:**

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

Ex: message = "meet me after the toga party"

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

ciphertext: MEMATRHTGPRYETEFETEOAAT

This technique is simple to cryptanalyze.

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

Ex: key: 4 3 1 2 5 6 7 (Matrix form)

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

(Keyed transposition cipher)

ciphertext: TTNAAPTMTSUOAODWCOIXKNTYPETZ

A pure transposition cipher is easily recognizable because it has the same letter frequency as the original plaintext.

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

Thus, if the foregoing message is reencrypted using the same algorithm, then we get:

key: 4 3 1 2 5 6 7
input: t t n a a p t
m t s u o a o
d w c o i x k
n t y p e t z
output: TNSCYAUOPTTWTTMDNAOIEPAXTTOKW

To visualize the result of this double transposition, designate the letters in the original plaintext message by the numbers designating their position. Thus with 28 letters in the message, the original sequence of letters is:

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

After first transposition, we have:

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

which has a somewhat regular structure. But after the second transposition, we have:

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

This is a much less structured permutation and is much more difficult to cryptanalyze.

- **Product Ciphers:**

ciphers using substitutions or transpositions are not secure because of language characteristics. Hence several ciphers in succession makes it harder:

- two substitutions make a more complex substitution
- two transpositions make a more complex transposition
- a substitution followed by a transposition makes a new harder cipher

This is a bridge from classical to modern ciphers.

NOTE:

- Properties:

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

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

$(a \times b) \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$

- SLE: Stagnography and program on multiplicative inverse of modulus.

→ Stagnography:

A plaintext message can be hidden in one of two ways:

- stagnography - conceal the existance of the message
- cryptography - render the message unintelligible to outsiders by various transformation of the text.

A simple form of steganography is one in which an arrangement of words or letters within an apparently harmless text spells out the real message. But it is time consuming to construct.

Some of the techniques that have been historically are:

1. Character marking:

selected letters of printed or typewritten text are overwritten in pencil. The marks are ordinarily not visible unless the paper is held at an angle to bright light.

2. Invisible Ink:

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

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

4. Typewriter correction Ribbon: Used between lines typed with a black ribbon, the results of typing with the correction

tape are visible only under a strong light.

Although these techniques may seem archaic, they have contemporary equivalents.

Steganography has a number of drawbacks when compared to encryption.

- It requires a lot of overhead to hide a relatively few bits of information.
- Once the system is discovered, it becomes virtually worthless. (can be overcomed by insertion of some sort of key)

Alternatively a message can be first encrypted and then hidden using stagnography.

UNIT - 2

Block Cipher and Encryption Standards

- **Introduction:**

A block cipher is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length. Typically a block size of 64 or 128 bits is used.

A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. Examples of classical stream ciphers are the autokeyed Vigenere cipher and the Vernam cipher.

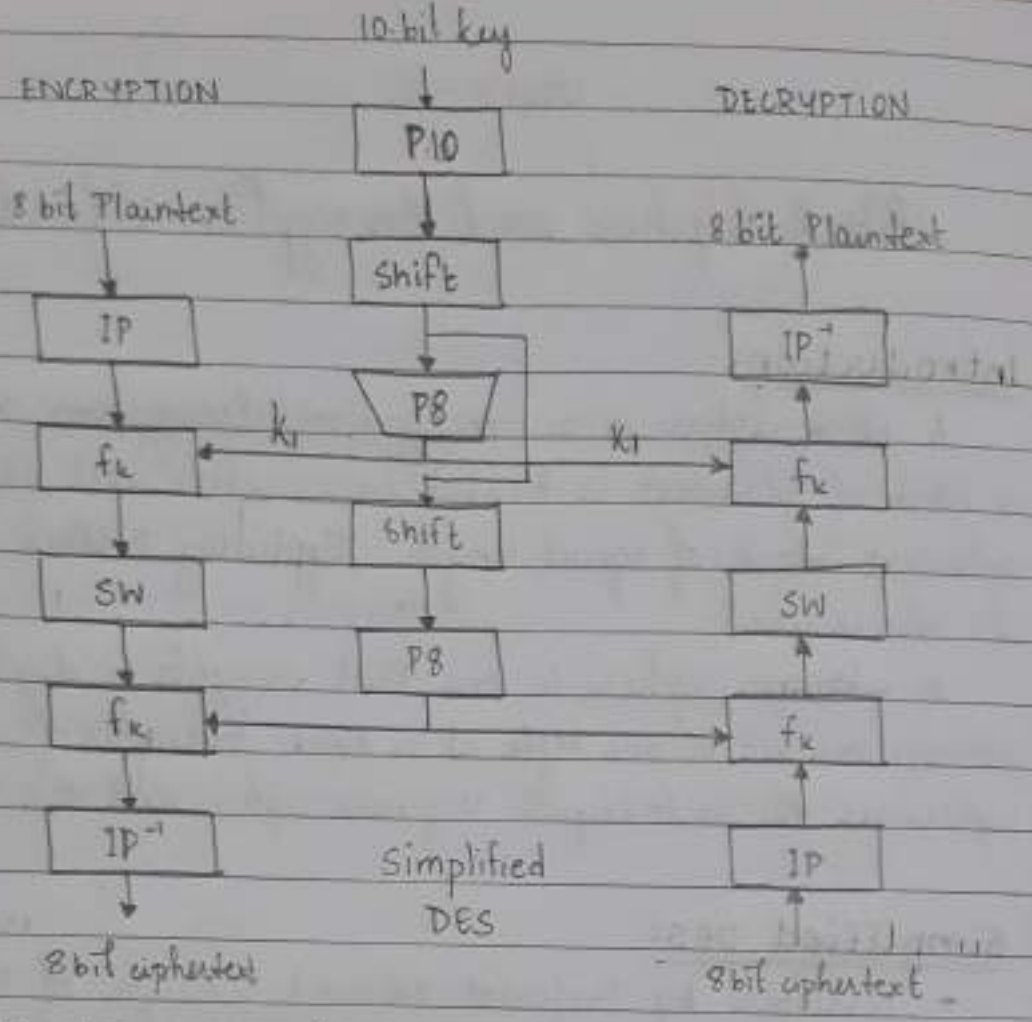
- **Simplified DES:**

- Developed by Professor Edward Schaefer of Santa Clara University in 1996.
- It is an educational algorithm rather than a secure encryption algorithm.
- It has similar properties and structure to Data Encryption Standard (DES) with much smaller parameters.
- Simplified DES algorithm for encryption takes
  a. 8-bit block of plaintext (Ex: 10101100)
  b. 10-bit key as input

and produces an output of 8-bit block of cipher text.

- Simplified DES decryption algorithm takes
  a. 8-bit block of ciphertext
  b. 10-bit key that was used for encryption

and produces the original 8-bit block of plaintext



Simplified DES Overview

The encryption algorithm involves five functions:

1. an initial permutation (IP).
2. a complex function labeled f_k, which involves both permutation and substitution operations and depends on a key input.
3. a simple permutation function that switches (SW) the two halves of the data
4. the function f_k again
5. a permutation function that is the inverse of the initial permutation (IP^{-1})

Therefore

$$\text{ciphertext} = IP^{-1}\left(f_{k_2}\left(SW\left(f_{k_1}\left(IP\left(\text{plaintext}\right)\right)\right)\right)\right)$$

where

$$k_1 = P8\left(\text{shift}\left(P10\left(\text{key}\right)\right)\right)$$

$$k_2 = P8\left(\text{shift}\left(\text{shift}\left(P10\left(\text{key}\right)\right)\right)\right)$$

similarly for decryption

plaintext = $IP^{-1}(f_{k_1}(SW(f_{k_2}(IP(ciphertext)))))$

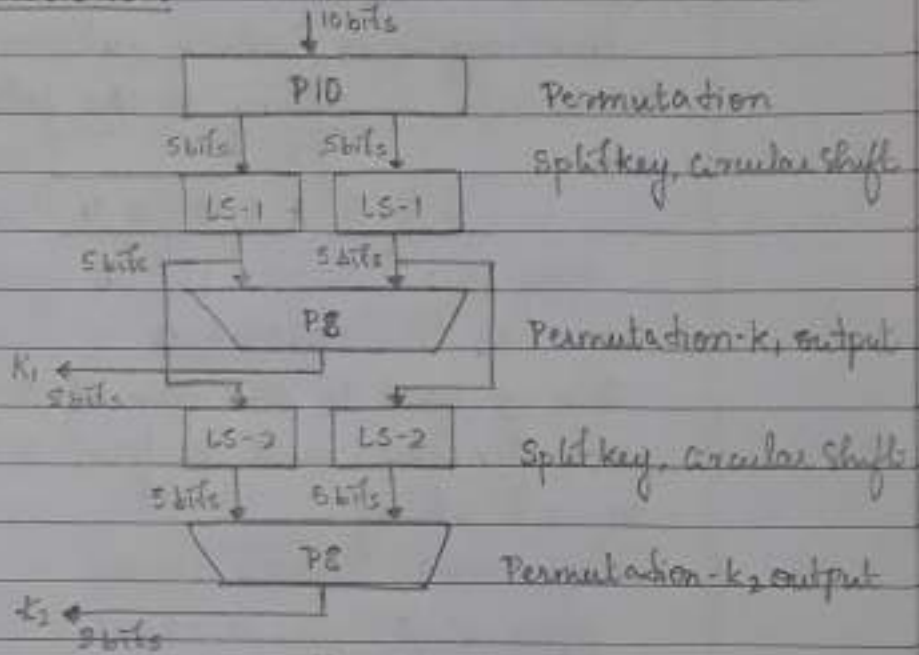
which is reverse of encryption.

- Simplified DES key generation

• Simplified DES uses a 10 bit key (shared between sender and receiver) from which two 8-bit subkeys are generated.

• The 10-bit key is first subjected to a permutation (P10). Then a shift operation is performed.

• Output of the shift operation is then passed through a permutation function that produces an 8-bit output (i.e., from P8) for the first subkey K_1.



• Output of the shift operation also feeds into another shift and another instance of P8 to produce the second subkey K_2.

* key generation procedure:

Let the 10-bit key be

$k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}$

Then the permutation P10 is defined as

$P10(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10})$

$= k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6$

ie.,

P10									
3	5	2	7	4	10	1	9	8	6

Example: key : 1010000010

P10(k) = 1000001100

The first output bit is bit 3 of the input; the second output bit is bit 5 of the input and so on.

Next to perform circular left shift (LS-1) or rotation, we should split the permuted key into two 5-bit halves.

P(10): 10000 01100

After LS-1: 00001 11000

Next apply P8, which picks out and permutes 8 bits out of the 10 bits according to the following rule:

P8
6 3 7 4 8 5 10 9

This results in subkey 1 (K_1)

After LS-1: 00001 11000

K_1 : 10100100

Then go back to the pair of 5-bit strings produced by the LS-1 functions and perform a circular shift of 2 bit positions on each string.

After LS-1: 00001 11000

After LS-2: 00100 00011

Finally permutation P8 is applied to produce subkey 2 (K_2)

After LS-2: 00100 00011

P(8) → K_2 : 01000011

- Simplified DES Encryption:

Encryption algorithm involves the sequential application of five functions.

* Initial and Final Permutations:

Input to the algorithm is an 8-bit block of plaintext which we can first permute using the IP function.

IP
2 6 3 1 4 8 5 7

This retains all 8 bits of the plaintext but mixes them up. At the end of the algorithm, the inverse permutation is used.

IP-1
4 1 3 5 7 2 8 6

The second permutation is the reverse of the first.

$IP^{-1}(IP(X)) = X$

* The function f_K:

f_K consists of a combination of permutation and substitution functions.

Let L and R be the leftmost and rightmost 4 bits of the 8-bit input to f_K respectively.

Let F be a mapping from 4 bit strings to 4 bit strings.

$f_K(L,R) = (L \oplus F(R,SK), R)$

where SK : subkey

$\oplus$: bit by bit XOR

Ex: output of IP stage is

10111101

and $F(R,SK) = 1110$

for some key SK

Then $f_K(L,R) = f_K(10111101)$

$= (1011 \oplus 1110, 1101)$

$= 01011101$

Input is a 4-bit number:

$n_1 n_2 n_3 n_4$

- Expansion / Permutation

E/P
4 1 2 3 2 3 4 1

ie., $n_4 \mid n_1 \quad n_2 \mid n_3$

$n_2 \mid n_3 \quad n_4 \mid n_1$

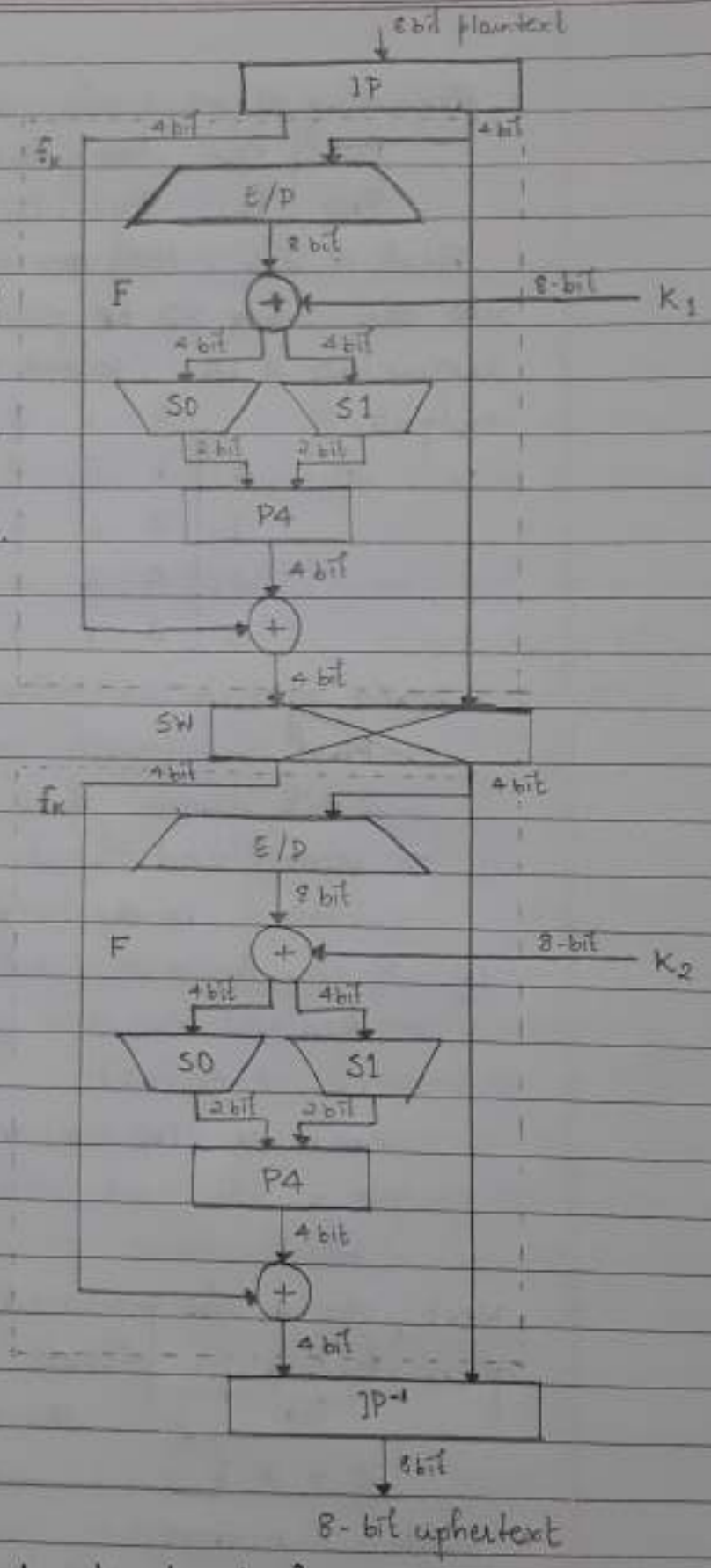
The 8-bit subkey

$K_1 = (k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}, k_{17}, k_{18})$

is added to the above value using XOR.

i.e., $n_4 \oplus k_{11} \mid n_1 \oplus k_{12} \quad n_2 \oplus k_{13} \mid n_3 \oplus k_{14}$

$n_2 \oplus k_{15} \mid n_3 \oplus k_{16} \quad n_4 \oplus k_{17} \mid n_1 \oplus k_{18}$



Renaming these 8 bits, we get

$P_{0,0}$	$P_{0,1}$	$P_{0,2}$	$P_{0,3}$
$P_{1,0}$	$P_{1,1}$	$P_{1,2}$	$P_{1,3}$

First 4 bits (first row of the preceding matrix) are fed into the S-box S0 to produce a 2-bit output and the remaining 4 bits (second row) are fed into another 2-bit output.

$S0 =$

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2

$S1 =$

	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

S-boxes operate as follows:

- First and fourth input bits are treated as a 2-bit number that specify a row of the S-box.
- second and third input bits specify a column, in base 2, is the 2-bit output.

Ex: If $(P_{0,0}\ P_{0,3}) = (00)$ and $(P_{0,1}\ P_{0,2}) = (10)$
then the output is from row 0 and column 2 of S0 i.e., 3 or (11)
Similarly $(P_{1,0}\ P_{1,3})$ and $(P_{1,1}\ P_{1,2})$ are used to index into a row and column of S1 to produce an additional 2 bits.

Next, the 4 bits produced by S0 and S1 undergo a further permutation as follows:

P4
2 4 3 1

The output of P4 is the output of the function F.

* Switch Function

Function f_k only alters the leftmost 4 bits of the input. The switch function (SW) interchanges the left and right 4 bits so that second instance of f_k operates on a different 4 bits. In second instance, the E/P, S0, S1 and P4 functions are same

with the key input is K_2.

Example:

plaintext : 0111 0010

From key generation

K1 : 1010 0100

K2 : 0100 0011

Encryption:

→ P : 0111 0010

→ IP : 1010 1001

→ EP : 1100 0011

→ K1 : 1010 0100 (⊕)

01100111

S0 S1

→ S0 : 0110 row 00 → 0, column 11 → 3 (10)

→ S1 : 0111 row 01 → 1, column 11 → 3 (11)

→ S0S1 : 1011

→ P4 : 0111 (⊕) 1010

1101

→ switch input : 1101 1001

switch output : 1001 1101

→ EP : 1110 1011

→ K2 : 0100 0011 (⊕)

1010 1000

S0 S1

→ S0 : 1010 row 10 → 2, column 01 → 1 (10)

→ S1 : 1000 row 10 → 2, column 00 → 0 (11)

→ S0S1 : 1011

→ P4 : 0101 (⊕) 1001

1100

→ Input to IP^{-1} : 1100 1101

Output of IP^{-1} : 0101 0111

Therefore ciphertext is 01010111 for the plaintext 01110010

- Analysis of Simplified DES:

A brute-force attack on simplified DES is feasible. With a 10-bit key, there are only $2^{10} = 1024$ possibilities. Given a ciphertext an attacker can try each possibility and analyze the result.

Each of the permutations and additions in the algorithm is a linear mapping. Nonlinearity comes from the S-boxes. Alternating linear maps with the nonlinear maps results in very complex polynomial expressions for the ciphertext bits, making cryptanalysis difficult.

- **Data Encryption Standard**: (DES)

- Applications of DES:

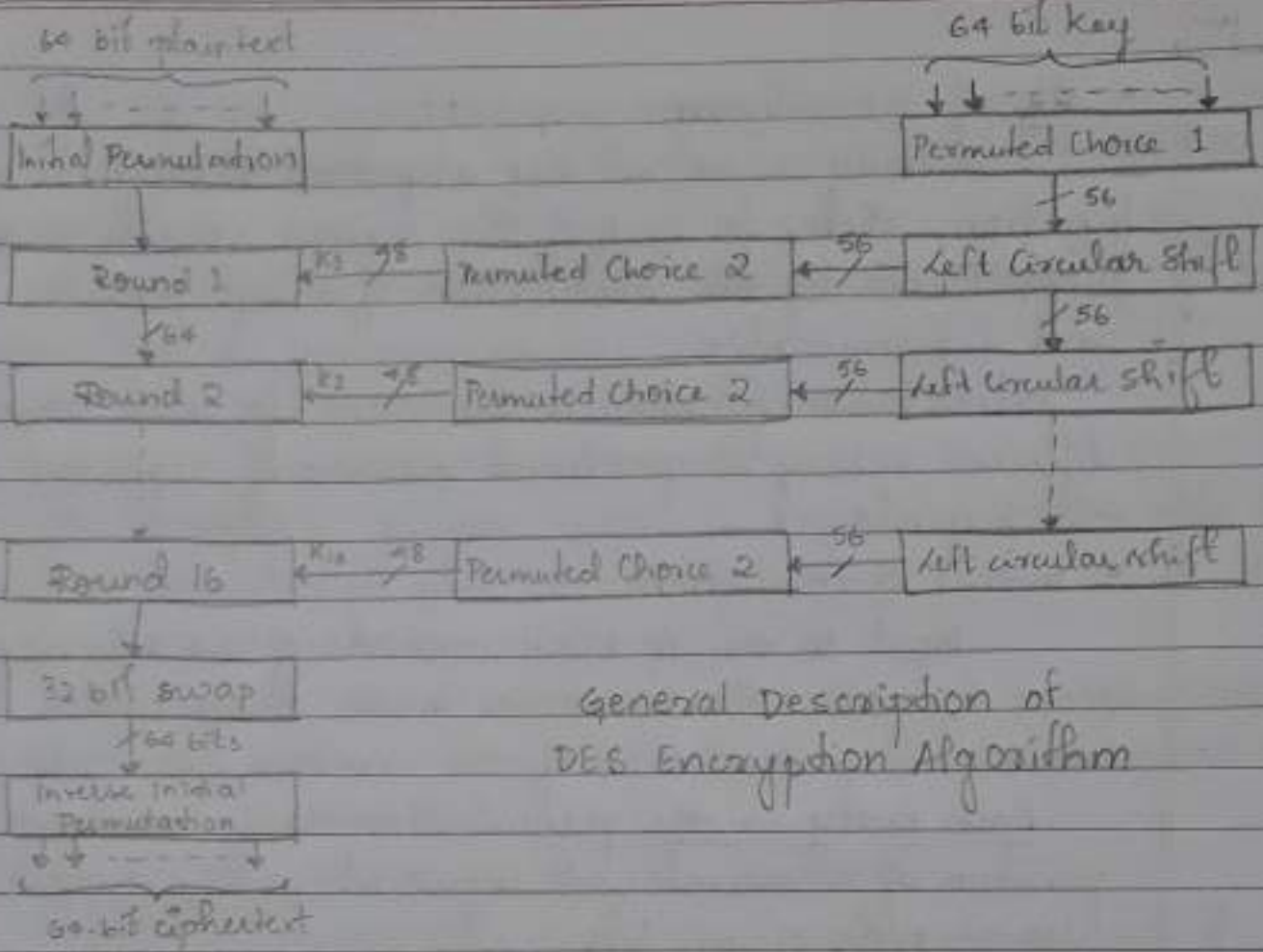
  - DES is widely used in financial applications
  - 1994: NIST reaffirmed DES for federal use and recommended it for applications other than the protection of classified information.
  - 1999: NIST issued a new version of its standard (FIPS PUB 46-3) that indicated that DES should be used only for legacy systems and that triple DES be used. (Triple DES - repeating DES algorithm thrice on the plaintext using two or three different keys to produce the ciphertext)

- DES Encryption:

There are two inputs to the encryption function:

1. plaintext: 64 bits
2. key: 56 bits

The input to the function is 64 bits but only 56 of these bits are used. The other 8 bits are used as parity bits or simply be set arbitrarily.



General Description of DES Encryption Algorithm

The processing of plaintext proceeds in three phases:

1. The 64 bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permutated input.
2. In second phase, there is sixteen rounds of the same function which involves both permutation and substitution functions. Output of the sixteenth (last) round consists of 64 bits that are a function of the input plaintext and the key.
   Left and right halves of the output are then swapped to produce the preoutput.
3. Preoutput is passed through a permutation that is inverse of initial permutation function to produce the 64-bit ciphertext.

With the exception of the initial and final permutations DES has the exact structure of a Feistel cipher.

* Use of 64-bit key:

Initially a 64-bit key is passed through a permutation function. Then, for each of the sixteen rounds a subkey (k_i) is produced by the combination of a left circular shift and a permutation.

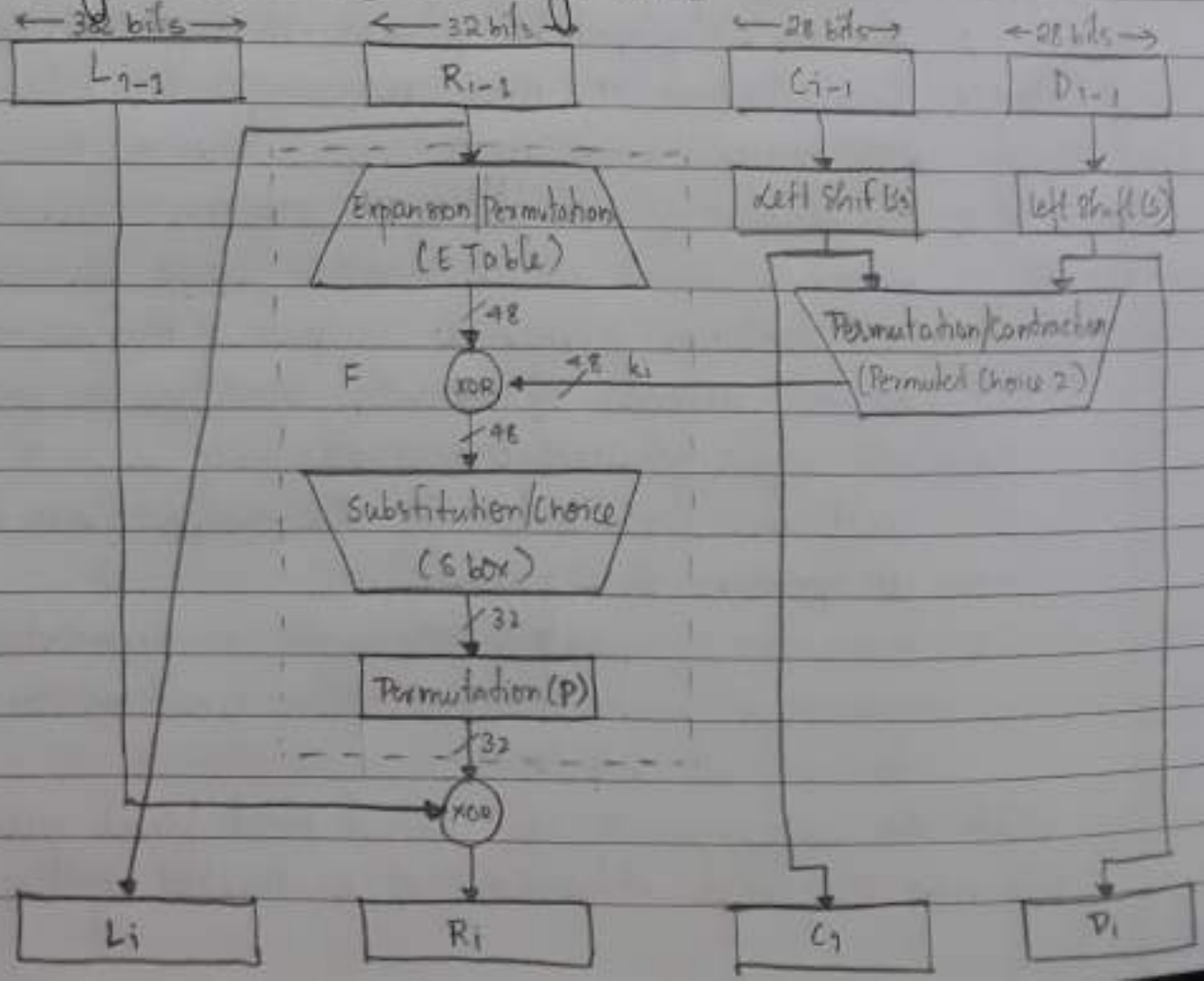
Permutation function is the same for each round but a different subkey is produced because of repeated shifts of the key bits.

* Initial Permutation:

Input to the IP block consists of 64 bits numbered from 1 to 64. The 64 entries in the permutation table contain a permutation of the numbers from 1 to 64.

Each entry in the permutation table indicates the position of a numbered input bit in the output which also consists of 64 bits.

* Single Round of DES Algorithm



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

As any classic Feistel cipher, the overall processing at each round can be summarized as:

$$L_i = R_{i-1} \quad \text{and} \quad R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

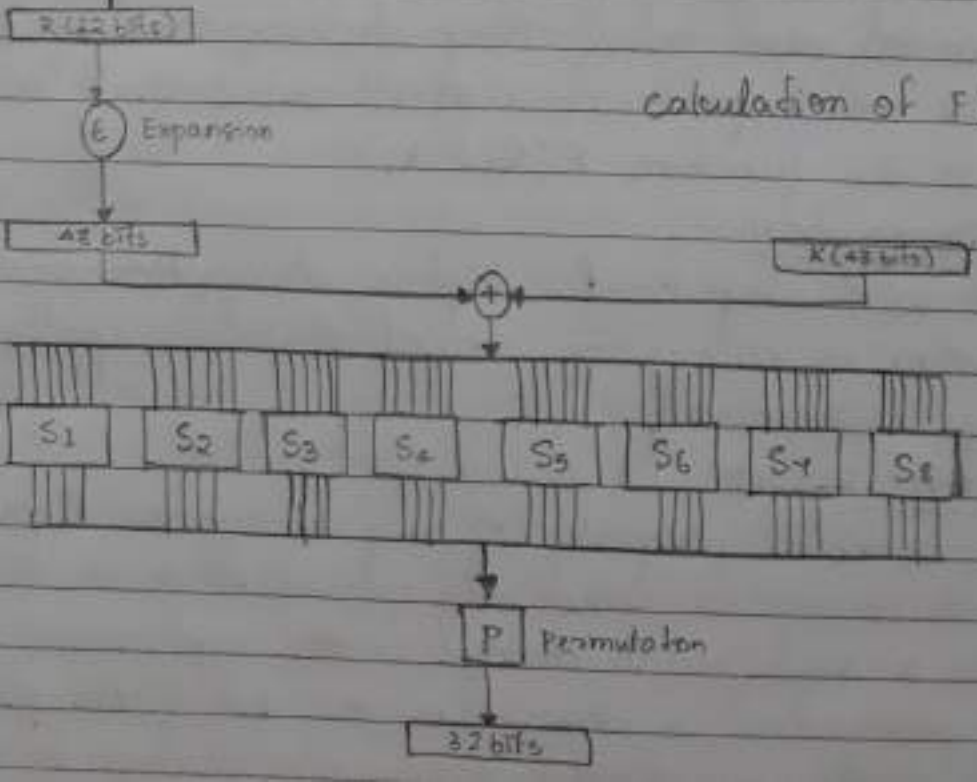
round key K_i is 48 bits

The R input is 32 bits which is expanded to 48 bits by using a permutation plus expansion resulting in 48 bits. This is then XORed with K_i.

This 48 bit result is then passed through a substitution function that produces a 32-bit output which is permuted by permutation function (P).

Rules of the S-boxes in the function F:

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



calculation of F(R, K)

Getting output of DES S-box:

- First and last bits of the input to box S_i forms a 2-bit binary number to select one of four substitutions defined by the four rows in the table for S_i.
- Middle four bits select one of the sixteen columns.
- Decimal value in the cell selected by the row and column is converted into its 4-bit representation to produce the output.

- DES Key Generation

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

Key is first subjected to a permutation governed by permuted choice 1. The resulting 56-bit key is then treated as two 28 bit quantities labeled as C_0 and D_0.

At each round C_{i-1} and D_{i-1} are separately subjected to a circular left shift of 1 or 2 bits. These shifted values serve as input to the next round. They also serve as input to permuted choice 2 which produces a 48 bit ouput that serves as input to the function $F(R_{i-1}, K_i)$.

- DES Decryption

As with any Feistel cipher, decryption uses the same algorithm as encryption, except that the application of the subkeys is reversed.

- Avalanche Effect

A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the ciphertext.

In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of ciphertext. This is referred to as the avalanche effect.

If the change were small, this might provide a way to reduce the size of the plaintext or key space to be searched. DES exhibits a strong avalanche effect.

- **Strength of DES:**

- Use of 56-Bit Key:

With key length of 56 keys, there are 2^{56} possible keys, which is approximately 7.2×10^{16} keys. Thus, brute-force attack appears impractical.

But in July 1998, DES finally and definitively proved insecure, when the Electronic Frontier Foundation (EFF) announced that it had broken a DES encryption using a special purpose DES cracker machine.

- Plaintext:

If the message is just plaintext in English, then the result pops out easily. If the text message has been compressed before encryption, then recognition is more difficult.

If the message is general type of data, like a numerical file, and this has been compressed, the problem becomes even more difficult to automate.

Thus, to supplement the brute-force approach, some knowledge about the expected plaintext is needed.

- Nature of DES Algorithm:

Cryptanalysis is possible by exploiting the characteristics of the DES algorithm. The focus of concern has been on the eight substitution tables or S-boxes that are used in each iteration. But so far no fatal weaknesses are discovered in the S-boxes.

- Timing Attacks:

A timing attack is one where information about the key or the plaintext is obtained by observing the time taken for a

given implementation to perform decryption on various ciphertext. It exploits the fact that an encryption or decryption algorithm takes slightly different amounts of time on different inputs.

Attacker may try to obtain hamming weight of the secret key. This is a long way from knowing the actual key but may serve as a first step.

DES appears to be fairly resistant to a successful timing attack. Although this is an interesting line of attack, so far this technique is not successful against DES or more powerful symmetric ciphers such as triple DES and AES.

- Cryptanalytic Attacks on DES:
  a. Differential Cryptanalysis:
     - Although differential cryptanalysis is a powerful tool, it does not do very well against DES and is of only theoretical interest. This is because of the need to strengthen DES against attacks using differential cryptanalysis. It was taken care by designing the S-boxes and the permutation P.
     - DES can be broken using differential cryptanalysis with 2^{47} chosen plaintexts or 2^{55} known plaintexts. But finding 2^{47} chosen plaintexts or 2^{55} known plaintext is impractical.
     - The idea of differential cryptanalysis was introduced by Eli Biham and Adi Shamir. This is a chosen-plaintext attack.
     - Analysis uses the propagation of input differences through the cipher. The difference refers to the exclusive OR of the two different inputs (plaintexts).

- If the analyst is able somehow to get the source system to insert a message chosen by the analyst, then a chosen-plaintext attack is possible which is used for differential cryptanalysis.
- Procedure:
  - begin with two plaintext messages m and m' with a given difference.
  - after each round trace a probable pattern of differences to yield a probable difference for the plaintext.
  - submit m and m' for encryption to determine the actual difference under the unknown key and compare the result to the probable difference.
  - If there is a match, then we suspect that all the intermediate rounds are correct. With is assumption some deductions about the key bits is made.
  - This procedure must be repeated many times to determine all the key bits.

b. Linear Cryptanalysis:

- Linear cryptanalysis is newer than differential cryptanalysis thus DES is more vulnerable to linear cryptanalysis as this attack was not known to the designers of DES.
- S-boxes are not very resistant to linear cryptanalysis. DES can be broken using 2^{43} pairs of known plaintexts which is unlikely to find practically.
- It was presented by Mitsuru Matsui in 1993.
- Analysis uses known-plaintext attacks. It uses the propagation of particular set of bits through the cipher.

NOTE:

→ Diffusion hides the relationship between the ciphertext and plaintext.

→ confusion hides the relationship between the ciphertext and the key.

- **Block Cipher Design Principles and Modes of Operation**

— DES Design Criteria:

The Criteria used in the design of DES focuses on the design of S-boxes and on the P function that takes the output of the S-boxes.

* The criteria for the S-boxes:

1. No output bit of any S-box should be too close a linear function of the input bits.
2. Each row of an S-box should include all 16 possible output bit combinations.
3. If two inputs to an S-box differ in exactly one bit, the outputs must differ in atleast two bits.
4. If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in atleast two bits.
5. If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same.
6. For any non zero 6-bit difference between inputs, no more than eight of the 32 pairs of inputs exhibiting that difference may result in the same output difference.

The first criterion is needed because the S-boxes are the only nonlinear part of DES. Remaining criteria were aimed to prevent differential cryptanalysis and to provide good confusion properties.

★ The criteria for the permutation P:

These criteria are intended to increase the diffusion of the algorithm.

• SLE – **Block Cipher Design Principles:**

There are three critical aspects of block cipher design:

1. Number of Rounds:

The greater the number of rounds, the more difficult it is to perform cryptanalysis, even for a relatively weak F.

2. Design of Function F.

In DES this function relies on the use of S-boxes.

Design criteria for F:

- It provides the element of confusion in a Feistel Cipher
- Function F must be nonlinear. The more nonlinear F, the more difficult any type of cryptanalysis will be.
- Must have good avalanche properties.
- SAC: Strict Avalanche Criterion: Any output bit j of an s-box should change with probability 1/2 when any single input bit i is inverted for all i, j.
- BIC: Bit Independence Criterion: Output bits j and k should change independently when any single input bit i is inverted for all i, j and k.
- SAC and BIC strengthen the effectiveness of the confusion function.

S-box design:

- Any change to the input vector to an S-box must result in random looking changes to the output.
- The relationship should be nonlinear and difficult to approximate with linear functions.
- One characteristic of S-box is its size. An m×n S-box has n-input bits and m output bits.

DES has 6×4 S-boxes

- Larger S-boxes are more resistant to differential and linear cryptanalysis.
- The larger the dimension n, the larger the look up table. For practical reasons, a limit of n equal to about 8 to 10 is usually imposed. Also larger the S-box, the more difficult it is to design.
- S-box must satisfy both SAC and BIC.
- All linear combinations of S-box columns should be bent (special class of Boolean functions that are highly nonlinear)
- Another criterion of S-boxes is GA: Guaranteed Avalanche: S-box satisfies GA of order r if, for a change of 1-bit in input, at least r output bits change.

3. Key Schedule Algorithm:

With any Feistel block cipher, the key is used to generate one subkey for each round.

Subkeys must be selected to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key.

Key schedule should guarantee key/ciphertext SAC and BIC.

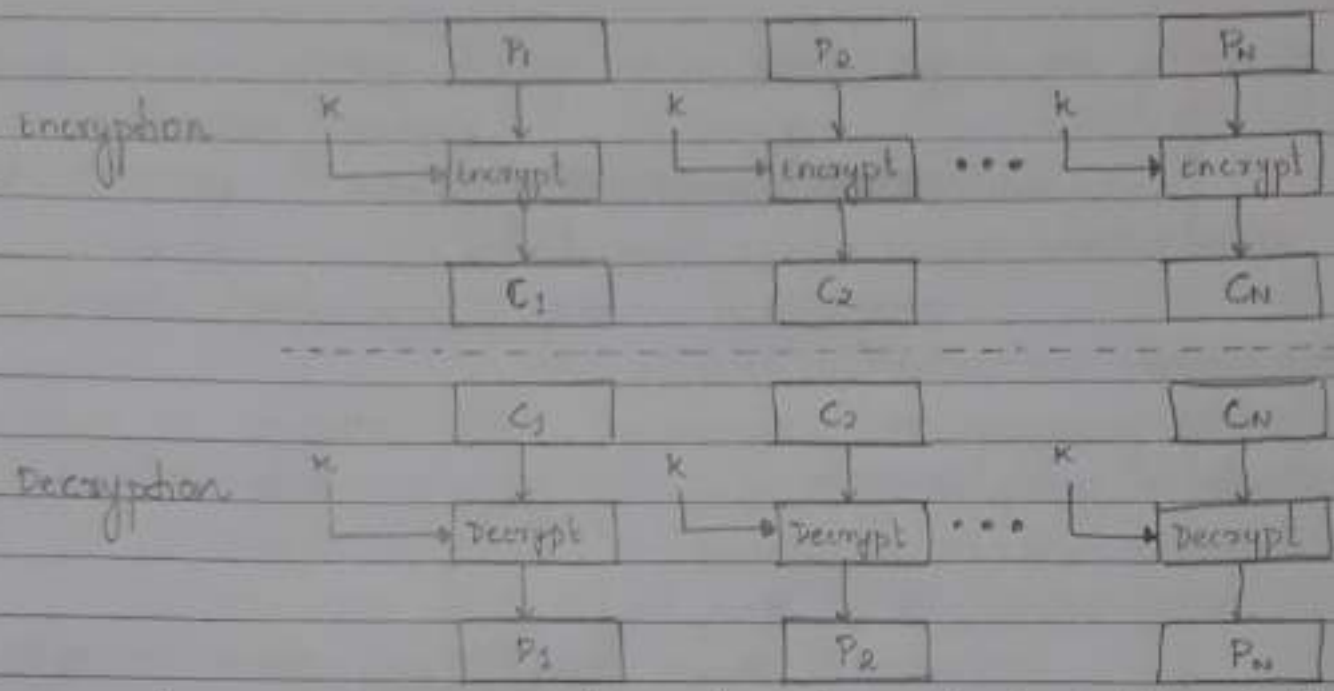
- Block Cipher Modes of Operation:

To apply DES in applications, 5 modes of operations are defined.

1. Electronic Codebook (ECB):

Each block of 64 bits plaintext is encoded independently using the same key.

Application: Secure transmission of single values (Ex an encryption key)



Procedure: Message is broken into independent blocks which are encrypted. Each block is a value which is substituted like a codebook. Each block is encoded independently of the other blocks : $C_i = E_K(P_i)$

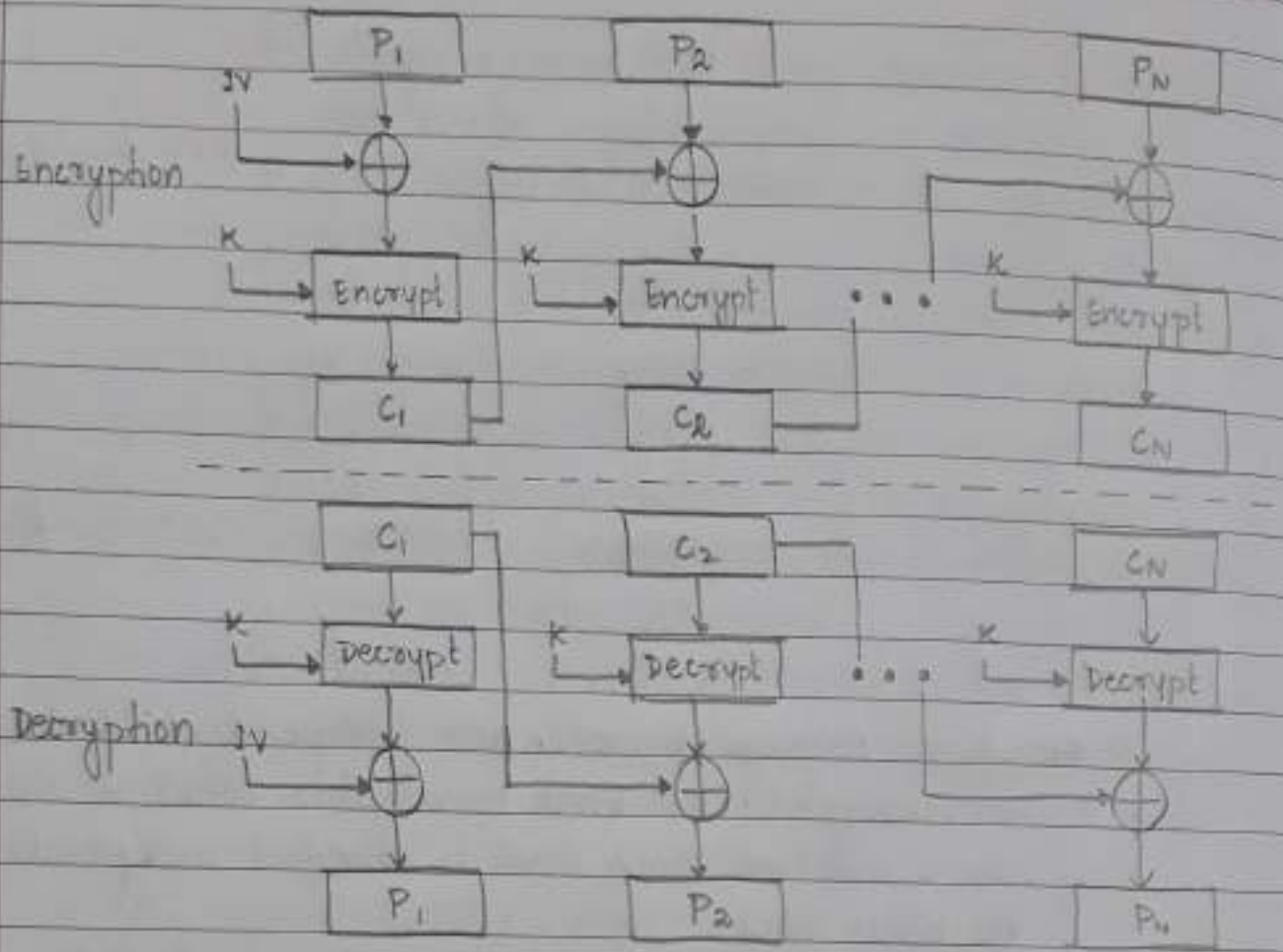
- ECB method is ideal for a short amount of data, such as an encryption key.
- The most significant characteristic of ECB is that if the same bit block of plain text appears more than once in the message, it will always produce the same ciphertext.
- For lengthy messages, the ECB mode may not be secure.
- If the message is highly structured, it may be possible for a cryptanalyst to exploit these regularities.

Disadvantage: Message repetitions may show in ciphertext.

2. Cipher Block Chaining (CBC):

The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.

Application: General purpose block oriented transmission
Authentication
(Bulk data encryption.)



Procedure: Message is broken into blocks that are linked together in encryption operation. Each previous block of ciphertext is chained with current plaintext block. It uses initial vector (IV) to start the process.

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

IV prevents same P from making same C.

Message Padding: At the end of the message, it must handle a possible last short block which is not as large as the blocksize of the cipher.

- pad either with known non-data value
  Ex: nulls
- pad last block along with count of pad size
  Ex: [b1 b2 b3 0 0 0 0 5]
  → 3 data bytes, 5 bytes pad + count

Disadvantages: 1. A ciphertext block depends on all blocks before it.

2. Any change to a block affects all following ciphertext blocks.
3. Needs Initialization Vector (IV)
   - which must be known to sender and receiver.
   - if sent in clear, attacker can change bits of first block, by changing corresponding bits of IV. Hence IV must either be a fixed value, or derived in a way hard to manipulate or sent encrypted in ECB mode before rest of the message or message integrity must be checked.

3. Cipher FeedBack (CFB):

Input is processed as s bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.

Applications: General Purpose stream oriented transmission Authentication.

Procedure: Message is treated as a stream of bits which is added to the output of the block cipher. Then the result is fed back to the next stage.

Standard allows any number of bits like 1, 8, 64 or 128 etc, to be fed back which are denoted as CFB-1, CFB-8, CFB-64, CFB-128 etc.

The most efficient to use all bits in the block are 64 or 128 bits.

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

Initialization Vector = IV

Advantages and Disadvantages:

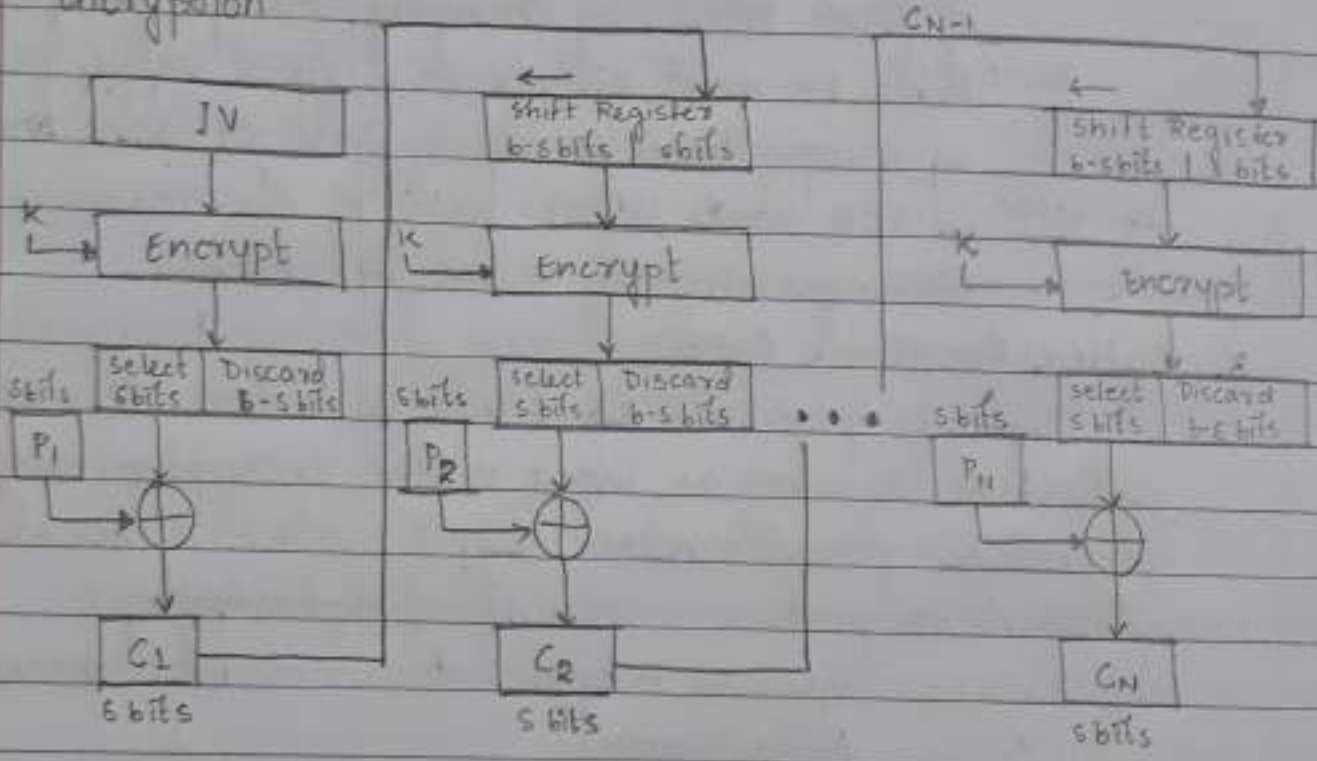
→ 1. Most common stream mode
2. Appropriate when data arrives in bits/bytes.

→ 1. The need to stall while doing block encryption after every s-bit.

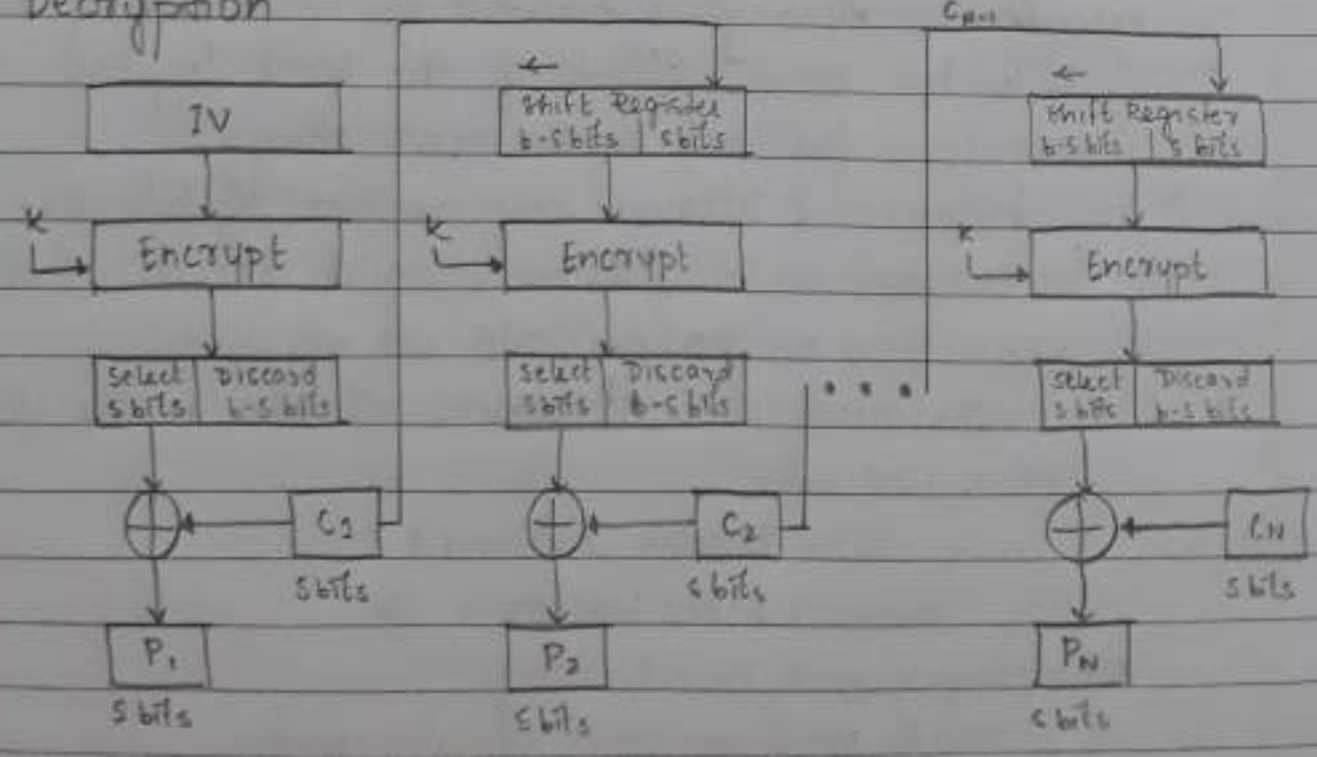
2. The block cipher is used in encryption mode at both ends

3. Errors propagate for several blocks after the error.

Encryption



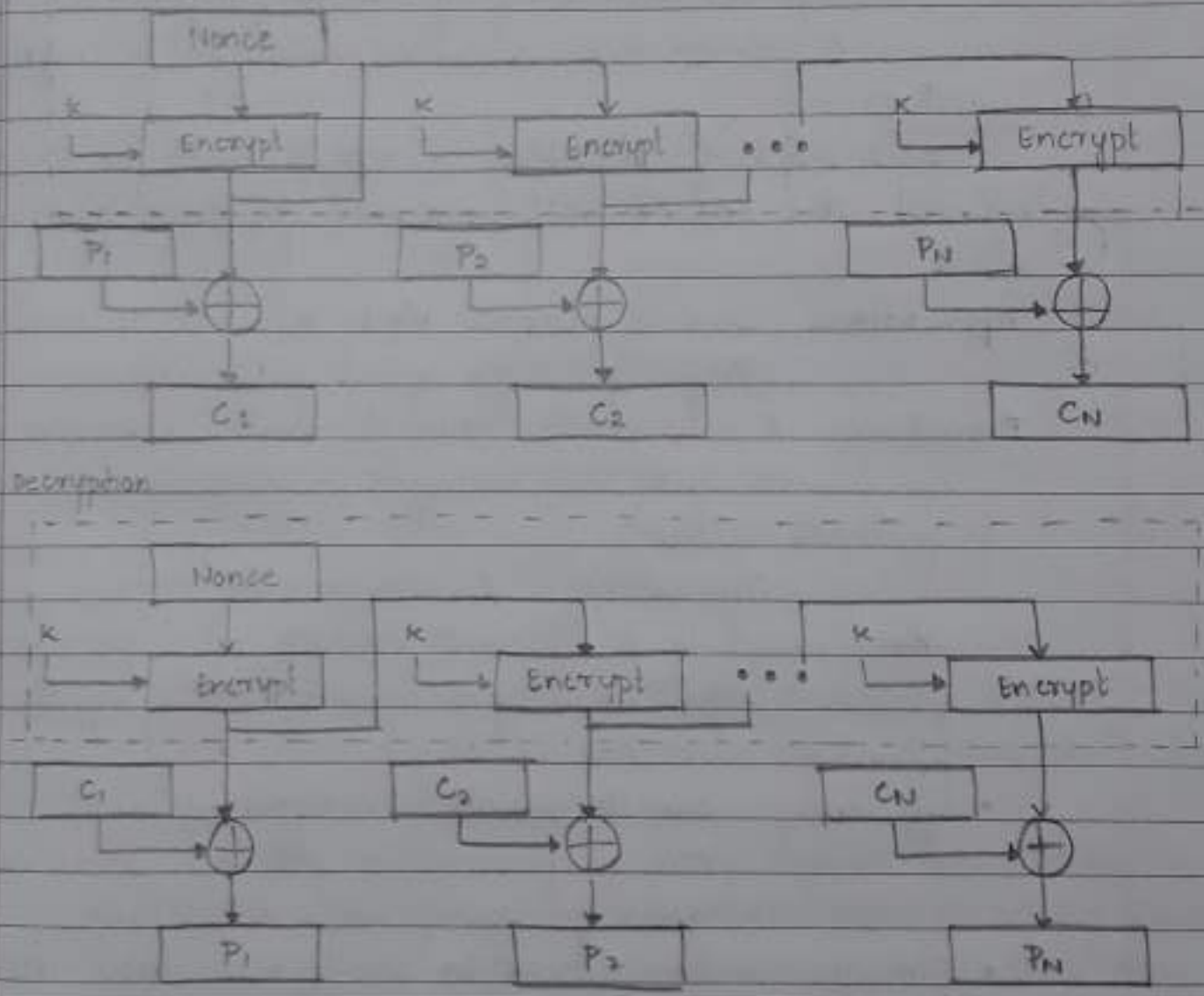
Decryption



4. Output Feedback (OFB):

Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output and full blocks are used

Applications: Stream-oriented transmission over noisy channel (Ex: Satellite communication)



Procedure: Message is treated as a stream of bits. Here the output of cipher is added to the message, then that output is fed back

$O_i = E_K(O_{i-1})$ IV : Initialization Vector

$C_i = P_i$ XOR O_i

Feedback is independent of message and can be computed in advance.

Advantages and Disadvantages:

→ 1. Can be precomputed
2. Bit errors do not propagate

→ 1. Needs an unique IV for each use, if reused attacker can recover outputs.
2. More vulnerable to message stream modification (change bits arbitrarily by changing ciphertext)
3. Sender and receiver must be in sync.

5. Counter:

Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.

Applications: General purpose block oriented transmission
Useful for high speed requirements

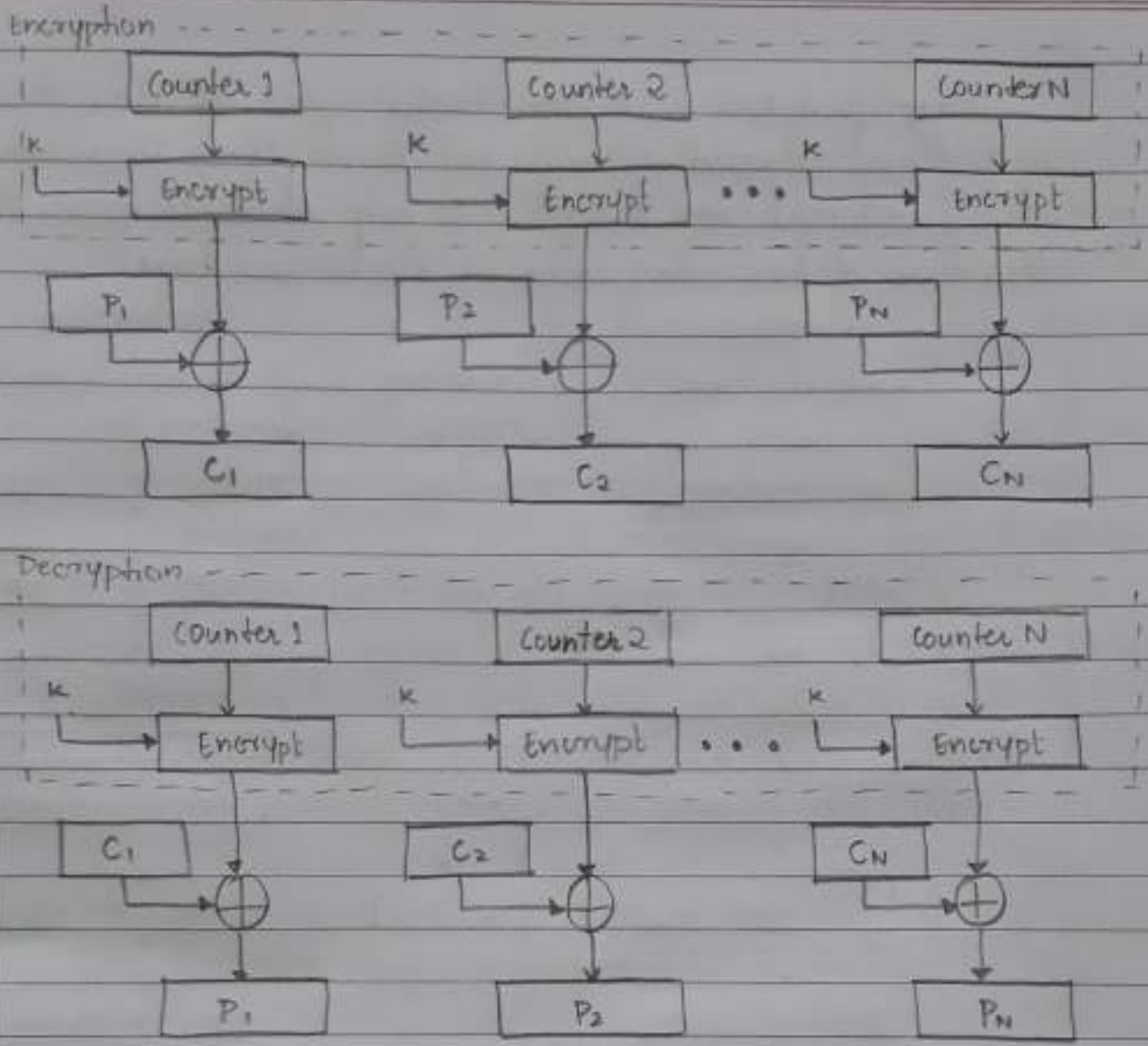
Procedure: A new mode, though proposed later it is similar to OFB but encrypts counter value instead of feedback value.

$$O_i = E_K(i) \qquad C_i = P_i \text{ XOR } O_i$$

There must be a different counter value for every plaintext block. Usually a counter value equal to the plaintext block size is used.

- Efficiency: can do parallel encryptions in h/w and s/w
  can preprocess in advance when needed
  good for bursty high speed links
- Provides random access to encrypted data blocks.
- Provides provable security.

Advantages: Hardware efficiency
Software efficiency
Preprocessing
Random Access
Provable Security
Simplicity

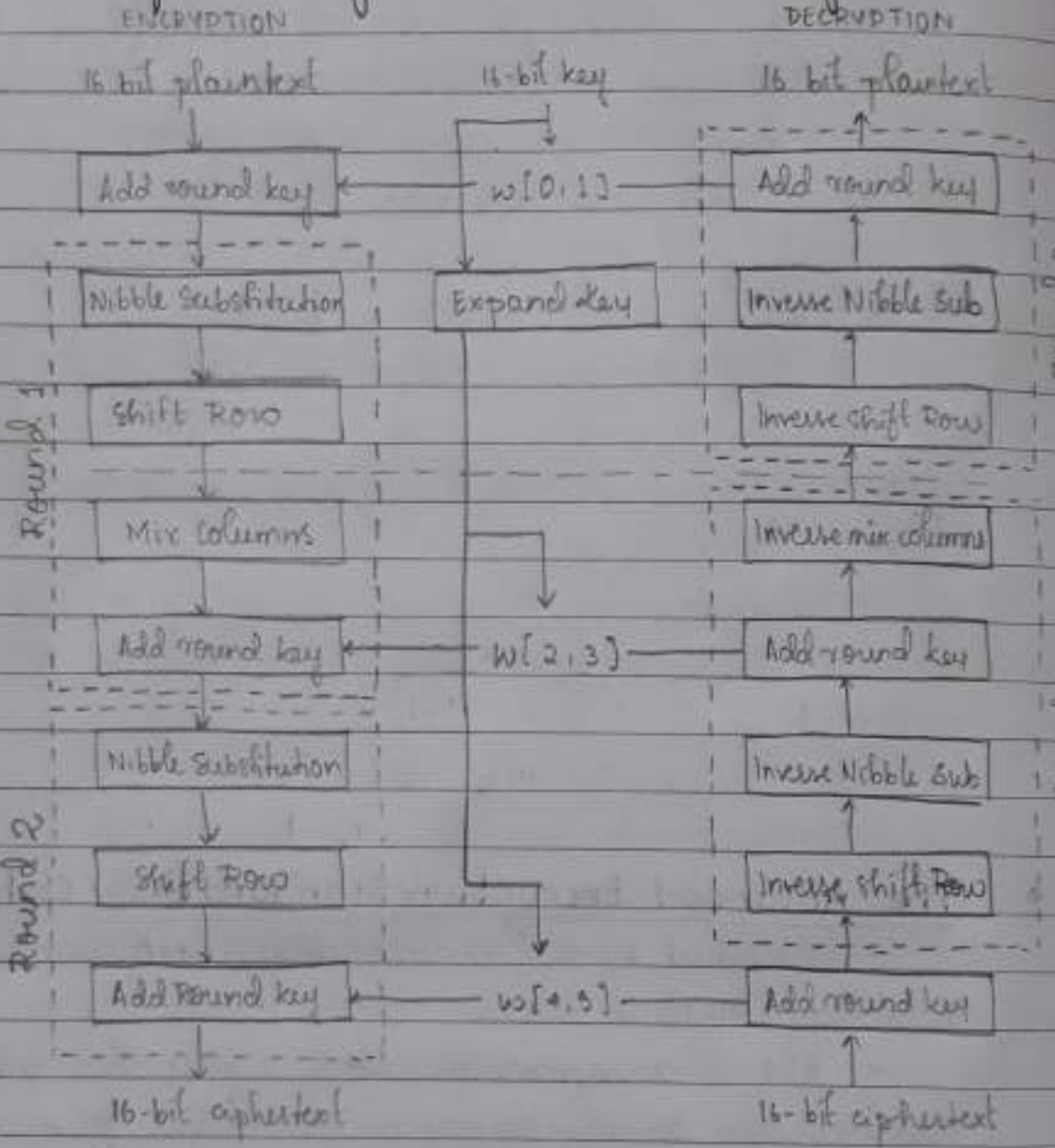


- **The Advanced Encryption Standard (AES) Cipher:**
  - Published by the National Institute of Standards and Technology (NIST) in 2001.
  - AES is a symmetric block cipher that intended to replace DES as the approved standard for many applications.
  - It uses a 128 bit block size and a key size of 128, 192 or 256 bits.
  - AES does not use a Feistel Structure.

- **Simplified AES Algorithm:**

The structure of S-AES is exactly same as AES. The differences are in the key size (16 bits), the block size (16 bits) and the number of rounds (2 rounds)

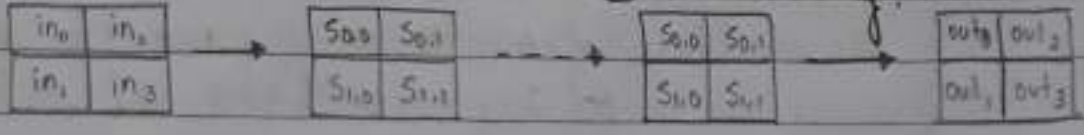
Simplified AES is a non-Feistel cipher. It was developed by Professor Edward Schaefer of Santa Clara University. It is an educational tool designed to help students learn the structure of AES using smaller blocks and keys.



- Simplified AES Encryption and Decryption:

→ State Array:

Simplified AES divides the block into a two by two array of "nibbles" (4 bits). This is called the state array.



→ Nibble Substitution / Inverse Nibble Substitution

An S-box is used to translate each nibble into a new nibble. Only one table is used for transformations of every nibble. Thus if two nibbles are same, then the transformation is also the same. The transformation is defined by a table lookup process.

To substitute a nibble, the left two bits define the row and the right two bits define the column of the substitution table. Hexadecimal number at the junction of the row and the column is the new nibble.

Transformation is done one nibble at a time. Hence SubNibbles involves 4 independent nibble to nibble transformations. This provides confusion effect.

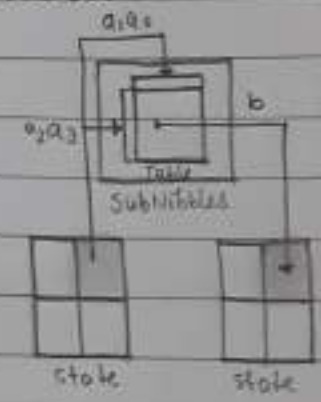
InvSubNibbles is the inverse of SubNibbles

a_0a_1 \ a_2a_3	00	01	10	11
00	9	4	A	B
01	D	1	8	5
10	6	2	0	3
11	C	E	F	7

SubNibbles Table

a_0a_1 \ a_2a_3	00	01	10	11
00	A	5	9	B
01	1	7	8	F
10	6	0	2	3
11	C	4	D	E

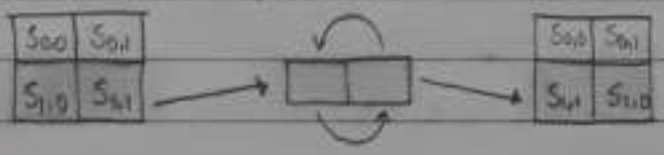
InvSubNibble Table



→ Shift Rows / Inverse Shift Rows:

Shift rows transformation permutes the nibbles. Shifting is to the left. Number of shifts depends on the row number (0 or 1) of the state matrix.

First row is not shifted and the second row is shifted left by 1 nibble. Shift Rows transformation operates one row at a time.

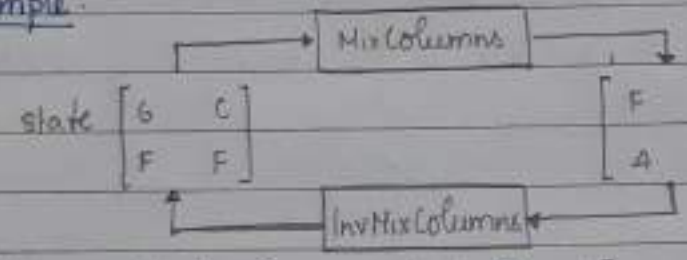


In the decryption, the shifting transformation is called InvShiftRows where the shifting is to the right. Number of shifts is the same as the number of the row (0 or 1) in the state matrix.

→ Mix Columns / Inverse Mix Columns:

Mix columns provides diffusion at bit level. It transforms each column of the state into new column. For this each column is multiplied by the matrix $\begin{bmatrix} 1 & 4 \\ 4 & 1 \end{bmatrix}$. These operations are done on GF(16).

Example:



$$\begin{bmatrix} 1 & 4 \\ 4 & 1 \end{bmatrix} \times \begin{bmatrix} 6 \\ F \end{bmatrix} = \begin{bmatrix} 1\times6 \oplus 4\times F \\ 4\times6 \oplus 1\times F \end{bmatrix} = \begin{bmatrix} 6 \oplus 9 \\ B \oplus F \end{bmatrix} = \begin{bmatrix} F \\ 4 \end{bmatrix}$$

1st column in State Matrix

$4 \times F = 0100 \times 1111$

$= x^2(x^3 + x^2 + x + 1) = x^5 + x^4 + x^3 + x^2$

Finding mod

$$\begin{array}{r|l} x^4 + x + 1 & x^5 + x^4 + x^3 + x^2 \quad | \quad x + 1 \\ \hline & x^5 + x^2 + x \\ \hline & x^4 + x^3 + x \\ & x^4 + x + 1 \\ \hline & x^3 + 1 \end{array}$$

$\Rightarrow 1001 = 9$ in Hexadecimal

$4 \times 6 = 0100 \times 0110$

$= x^2(x^2 + x) = x^4 + x^3$

Finding mod

$x^4+x+1 \;\big|\; x^4+x^3 \;\big|\; 1$

x^4+x+1

$x^3+x+1 \Rightarrow 1011 = B$ in hexadecimal

$6 \oplus 9$: 0110 ⊕ 1001 = 1111 ⇒ F

$B \oplus F$: 1011 ⊕ 1111 = 0100 ⇒ 4

Similarly

$$\begin{bmatrix} 1 & 4 \\ 4 & 1 \end{bmatrix} \times \begin{bmatrix} C \\ F \end{bmatrix} = \begin{bmatrix} 1\times C \oplus 4\times F \\ 4\times C \oplus 1\times F \end{bmatrix} = \begin{bmatrix} C \oplus 9 \\ 5 \oplus F \end{bmatrix} = \begin{bmatrix} 5 \\ A \end{bmatrix}$$

2nd column in state matrix

$4 \times C = 0100 \times 1100$

$= x^2(x^3+x^2) = x^5+x^4$

Finding mod

$x^4+x+1 \;\big|\; x^5+x^4 \;\big|\; x+1$

x^5+x^2+x

x^4+x^2+x

x^4+x+1

$x^2+1 \Rightarrow 0101 = 5$ in hexadecimal

$C \oplus 9$: 1100 ⊕ 1001 = 0101 ⇒ 5

$5 \oplus F$: 0101 ⊕ 1111 = 1010 ⇒ A

InvMixColumns is the inverse of MixColumn transformation.

• SLE

NOTE: **Finite Fields**

A field is an algebraic object with two operations: addition and multiplication (+ and *), although they will not necessarily be ordinary addition and multiplication. Cryptography focuses on finite fields. For any prime integer p and any integer n greater than or equal to 1, there is a unique field with p^n elements in it, denoted by $GF(p^n)$

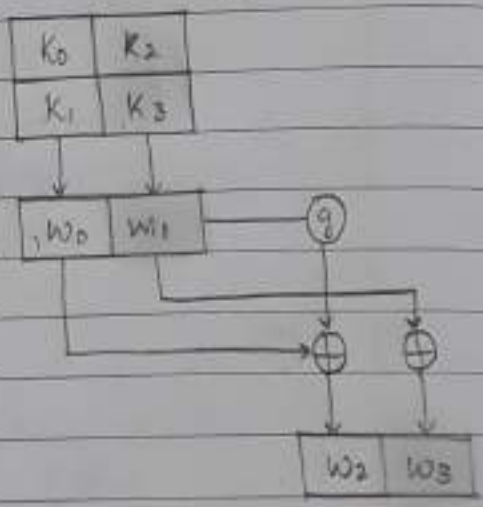
→ Add Round Key:

Add Round Key also proceeds one column at a time similar to MixColumns. In MixColumns, it multiplies a constant square matrix by each state column whereas in AddRoundKey it adds a round key word with each state column matrix.

Operations in MixColumns are matrix multiplication where as the operations in AddRoundKey are matrix addition. Addition is done in $GF(2^4)$ field because addition and subtraction in this field are the same, the AddRoundKey transformation is inverse of itself

Key Expansion

Four nibbles in the key are grouped into two 8-bit "words", which will be expanded into 6 words (from 1,2 to 3,4 to 5,6 or from 0,1 to 2,3 to 4,5).



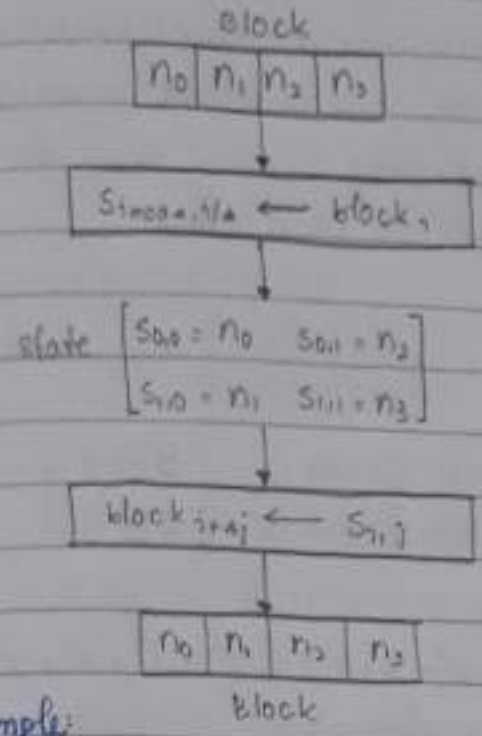
First part of the expansion which produces the third and fourth words is shown here

Rest of the expansion is done in exactly the same way, replacing w_0 and w_1 with w_2 and w_3 and replacing w_2 and w_3 with w_4 and w_5.

The function g consists of the subfunctions Rotate word, substitute word and XOR with a round constant

→ Block:

simplified AES encrypts and decrypts block data. A block in simplified AES is a group of 16 bits however, a block can be represented as a row matrix of 4 nibbles.



Block to state and state to block transformation

Example:

Let us see how a 16-bit block can be shown as a 2×2 matrix. Assume that the text block is 1011 0111 1001 0110. We first show the block as 4 nibbles. The state matrix is then filled up column by column.

Changing ciphertext to a state

Block (bits) 1011 0111 1001 0110

Block (nibble) B 7 9 6

State $\begin{bmatrix} B & 9 \\ 7 & 6 \end{bmatrix}$

Example: Perform Nibble Substitution for $\begin{bmatrix} 0 & 6 \\ 4 & 5 \end{bmatrix}$

⇒ 0,4 : 0000 0100 (index into a row, index into a column)

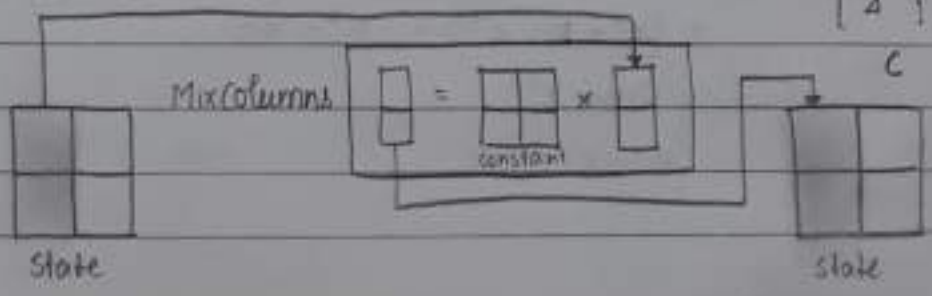
⇒ 6,5 : 0110 0101

From the SubNibble table we get the state matrix

State Matrix : $\begin{bmatrix} 9 & 8 \\ D & 1 \end{bmatrix}$

Inverse Mix Column

$\begin{bmatrix} 1 & 4 \\ 4 & 1 \end{bmatrix}$ C $\quad$ $\begin{bmatrix} 9 & 2 \\ 2 & 9 \end{bmatrix}$ C^{-1}



The MixColumns and InvMixColumns transformations are inverses to each other. If the two constant matrices are inverses to each other, it is easy to prove that the two transformations are inverses to each other.

Example:

$$\begin{bmatrix} F & 5 \\ 4 & A \end{bmatrix} \text{ state} \longrightarrow \boxed{\text{InvMixColumns}} \longrightarrow \text{state} \begin{bmatrix} 6 & C \\ F & F \end{bmatrix}$$

$$C^{-1} \begin{bmatrix} 9 & 2 \\ 2 & 9 \end{bmatrix} \times \begin{bmatrix} F \\ 4 \end{bmatrix} = \begin{bmatrix} 9\times F \oplus 2\times 4 \\ 2\times F \oplus 9\times 4 \end{bmatrix} = \begin{bmatrix} E \oplus 8 \\ D \oplus 2 \end{bmatrix} = \begin{bmatrix} 6 \\ F \end{bmatrix}$$

$\rightarrow 9\times F = 1001 \times 1111$

$= (x^3+1)(x^3+x^2+x+1) = x^6+x^5+x^4+x^3+x^3+x^2+x+1$

$= x^6+x^5+x^4+x^2+x+1$

Finding mod

$$x^4+x+1 \;\big|\; x^6+x^5+x^4+x^2+x+1 \;\big|\; x^2+x+1$$

$$x^6+x^3+x^2$$

$$x^5+x^4+x^3+x+1$$

$$x^5+x^2+x$$

$$x^4+x^3+x^2+1$$

$$x^4+x+1$$

$$x^3+x^2+x \Rightarrow 1110 = E$$

$\rightarrow 2\times F = 0010 \times 1111$

$= x(x^3+x^2+x+1) = x^4+x^3+x^2+x$

Finding mod

$$x^4+x+1 \;\big|\; x^4+x^3+x^2+x \;\big|\; 1$$

$$x^4+x+1$$

$$x^3+x^2+1 \Rightarrow 1101 = D$$

$\rightarrow 9\times 4 = 1001 \times 0100$

$= (x^3+1)(x^2)$

$= x^5+x^2$

Finding mod

$x^4+x+1 \;\big|\; x^5+x^2 \;\big|\; x$

x^5+x^2+x

$x \Rightarrow 0010 = 2$

E ⊕ 8 : 1110 ⊕ 1000 = 0110 ⇒ 6

D ⊕ 2 : 1101 ⊕ 0010 = 1111 ⇒ F

$$\begin{bmatrix} 9 & 2 \\ 2 & 9 \end{bmatrix} \times \begin{bmatrix} 5 \\ A \end{bmatrix} = \begin{bmatrix} 9\times5 \oplus 2\times A \\ 2\times5 \oplus 9\times A \end{bmatrix} = \begin{bmatrix} B \oplus 7 \\ A \oplus 5 \end{bmatrix} = \begin{bmatrix} C \\ F \end{bmatrix}$$

→ $9\times5 = 1001 \times 0101$

$= (x^3+1)(x^2+1) = x^5+x^3+x^2+1$

Finding mod

$x^4+x+1 \;\big|\; x^5+x^3+x^2+1 \;\big|\; x$

x^5+x^2+x

$x^3+x+1 \Rightarrow 1011 = B$

→ $2\times A = 0010 \times 1010$

$= x(x^3+x) = x^4+x^2$

Finding mod

$x^4+x+1 \;\big|\; x^4+x^2 \;\big|\; 1$

x^4+x+1

$x^2+x+1 \Rightarrow 0111 = 7$

→ $9\times A = 1001 \times 1010$

$= (x^3+1)(x^3+x) = x^6+x^4+x^3+x$

Finding mod

$x^4+x+1 \;\big|\; x^6+x^4+x^3+x \;\big|\; x^2+1$

$x^6+x^3+x^2$

x^4+x^2+x

x^4+x+1

$x^2+1 \Rightarrow 0101 = 5$

B ⊕ 7 : 1011 ⊕ 0111 = 1100 ⇒ C

A ⊕ 5 : 1010 ⊕ 0101 = 1111 ⇒ F

UNIT - 3

Public-key Encryption and Hash Function

- **Principles of Public-Key Cryptosystems:**

Asymmetric encryption is a form of cryptosystem which where encryption and decryption are performed using different keys: one public key and one private key. It is also known as public-key encryption.

Asymmetric encryption can be used for confidentiality, authentication or both.

- Public-key Cryptography:

Public-key algorithms are based on mathematical functions rather than on substitution and permutation. It is asymmetric involving the use of two separate keys. Public key cryptography is useful for key management and signature applications.

Asymmetric keys: Two related keys, a public key and a private key that are used to perform complementary operations like encryption and decryption or signature generation and signature verification.

Public key certificate: A digital document issued and digitally signed by the private key of a certification Authority that binds the name of a subscriber to a public key.

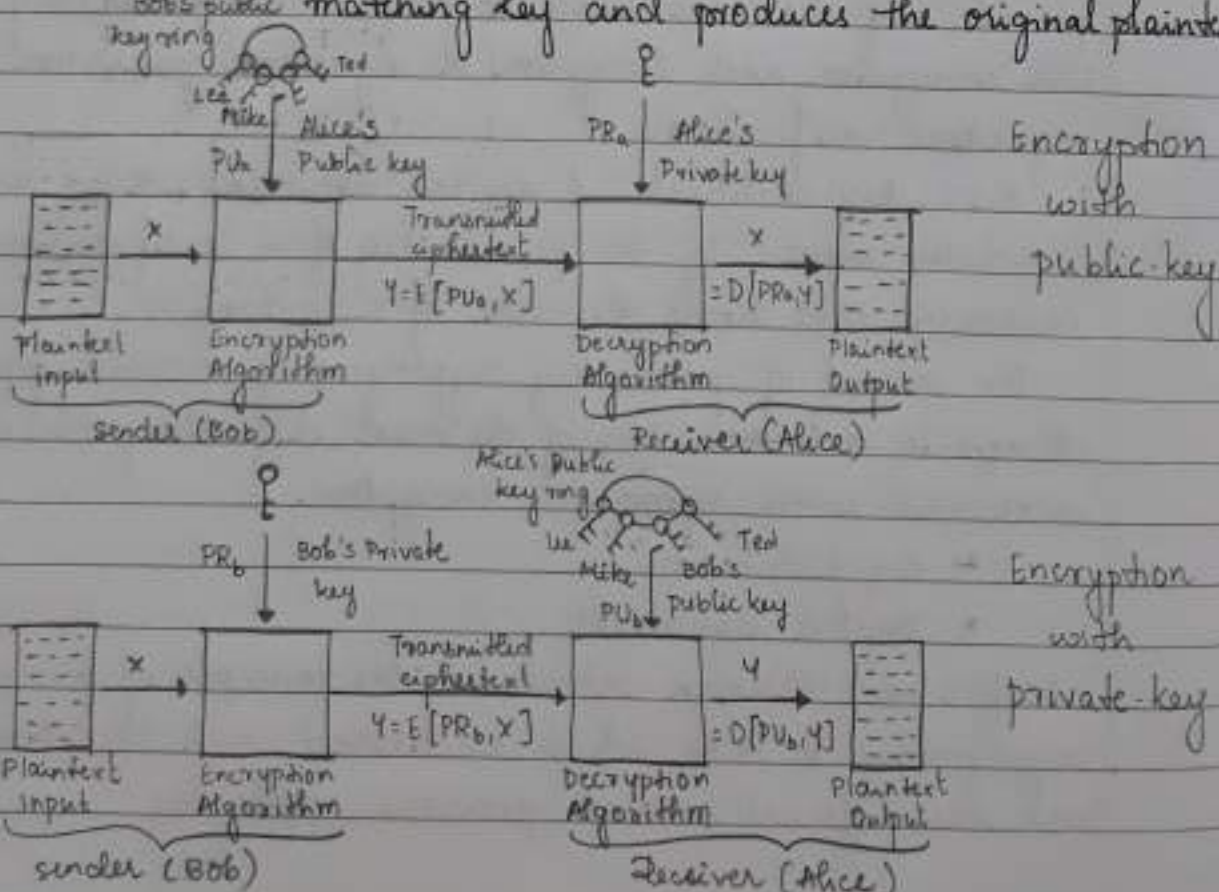
The concept of public key cryptography evolved from an attempt to overcome two of the most difficult problems associated with symmetric encryption.

- Key distribution
- Digital signatures.

Diffie and Hellman introduced the concepts of public key cryptography in 1976 which addressed both the above problems and was different from all previous approaches.

* Characteristics:
  - It is computationally infeasible to determine the decryption key given only the knowledge of the cryptographic algorithm and the encryption key.
  - One key is used for encryption and a different but related key is used for decryption.

* Ingredients of a public-key encryption scheme:
  - Plaintext: Readable message or data that is fed into the algorithm as input.
  - Encryption Algorithm: Performs various transformations on the plaintext.
  - Public and Private key: Pair of keys selected such that one is used for encryption and other for decryption
  - Ciphertext: Scrambled message produced as output which depends on the plaintext and the key. For a given message two different keys will produce two different ciphertexts.
  - Decryption Algorithm: Accepts the ciphertext and the matching key and produces the original plaintext.



* Essential Steps (Encryption using public-key)

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places only public key in a public register or other accessible file. The companion key is kept private. Each user maintains a collection of public keys obtained from others.
3. If sender (Bob) wishes to send a confidential message to the receiver (Alice), then sender (Bob) encrypts the message using receiver's (Alice's) public key.
4. When the receiver (Alice) receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only receiver (Alice) knows her private key.

* Conventional Encryption versus Public key Encryption

Conventional Encryption	Public-Key Enctyption
Needed to work:	Needed to work:
• The same algorithm with the same key is used for encryption and decryption.	• One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption.
• The sender and receiver must share the algorithm and key.	• The sender and receiver must each have one of the matched pair of keys (not the same one)
Needed for security:	Needed for security:
• The key must be kept secret.	• One of the two keys must be kept secret.
• It must be impractical to decipher a message if no other information is available.	• It must be impractical to decipher a message if no other information is available

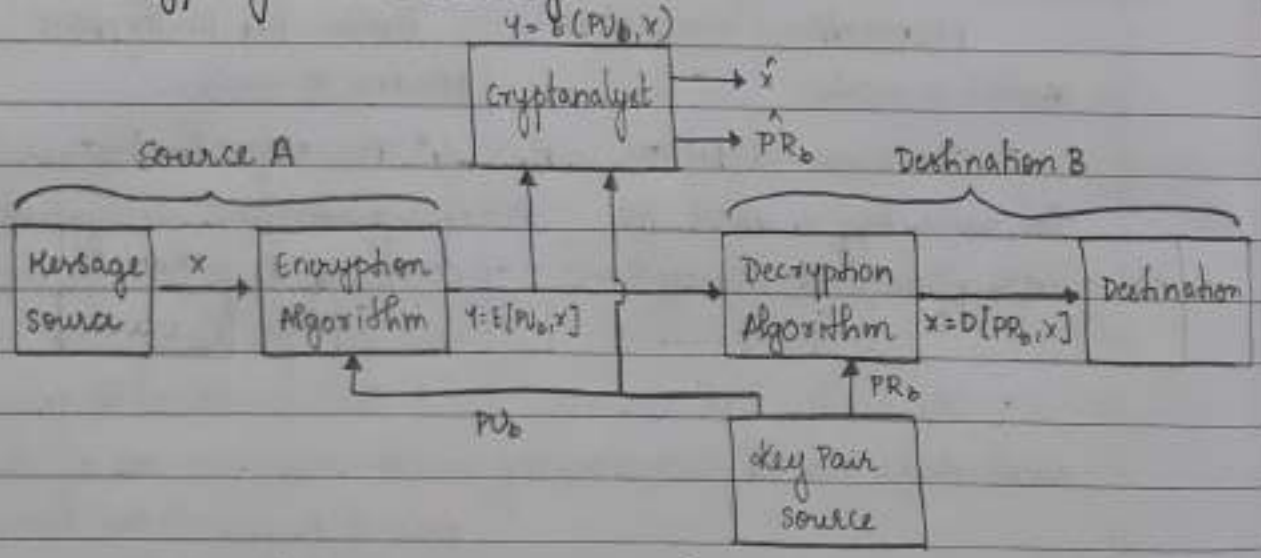
• Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	• knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

+ Notations:

PUa : Public key for user A
PRa : Private key for user A
Ka : Secret key owned by user A

Encryption of plaintext x can be performed with a
secret key : E (Ka, X)
Public key : E (PUa, X)
Private key : E (PRa, X)

+ Public Cryptosystem : Secrecy



Source A produces a message in plaintext $X = [X_1, X_2, \ldots, X_M]$. There are M elements of X which are letters in finite alphabet. The message is intended for the destination B. B generates a related pair of keys: a public key PU_b and a private key PR_b. PR_b is only known to B, whereas PU_b is publicly available and therefore accessible by A.

Operation: Secrecy provides confidentiality.

With the message X and the encryption key PU_b as input, A forms the ciphertext:

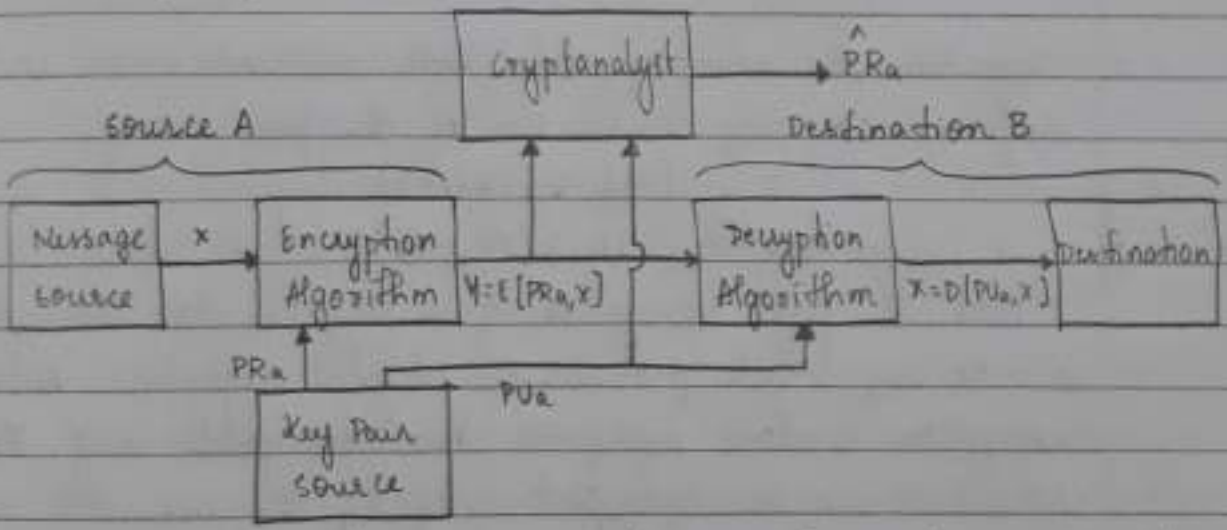
$$Y = [Y_1, Y_2, \ldots, Y_N] \text{ i.e., } Y = E(PU_b, X)$$

Intended receiver inverts the transformation as:

$$X = D(PR_b, Y)$$

An adversary, observing Y and having access to PU_b, but not having access to PR_b or X, attempts to recover X and PR_b. It is assumed that the adversary has knowledge of the encryption (E) and decryption (D) algorithms. It focuses on recovering X by generating a plaintext estimate $\hat{X}$ and/or recovering PR_b by generating an estimate $\hat{PR}_b$.

Public Cryptosystem: Authentication:



Operation: It provides authentication in this scheme. But does not provide confidentiality because any observer can decrypt the message by using sender's public key.

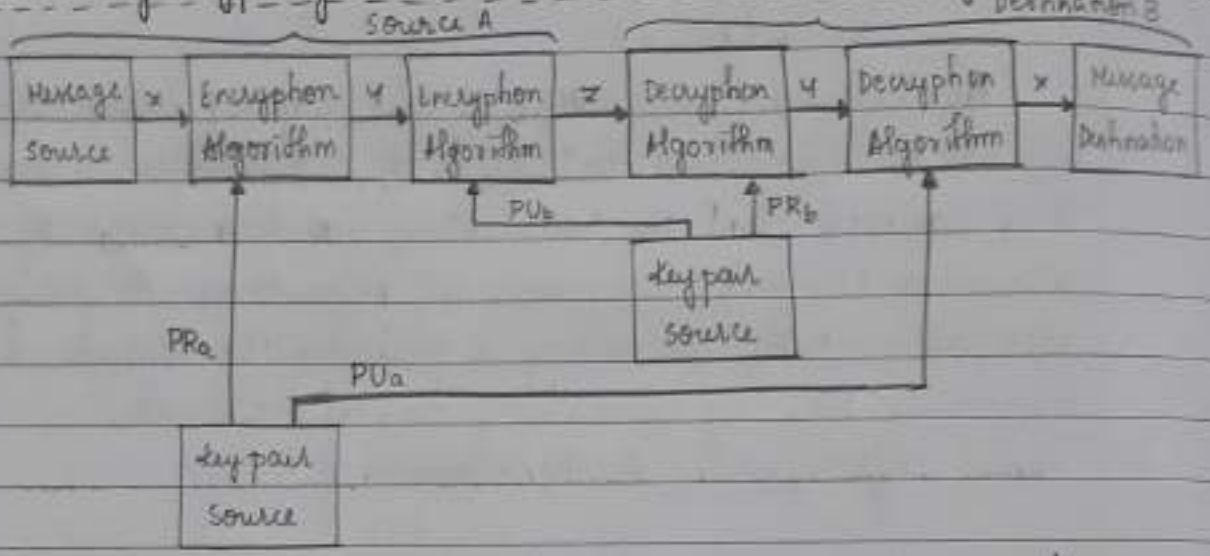
Encryption and Decryption are carried out as:

$$Y = E(PR_a, X) \text{ and } X = D(PU_a, Y)$$

Here, A prepares a message to B and encrypts it using A's private key before transmitting it. B can decrypt the message using A's public key because the message was encrypted using A's private key, only A could have prepared the message.

Therefore the entire encrypted message serves as a digital signature. Also it is impossible to alter the message without access to A's private key, so the message is authenticated both in terms of source and data integrity.

* Public-key Cryptosystem: Authentication and Secrecy



Operation: This scheme provides both authentication and confidentiality by double use of the public-key.

$$Z = E(PU_b, E(PR_a, X))$$

$$X = D(PU_a, D(PR_b, Z))$$

Here, the first encryption of message is done using the sender's private key which provides the digital signature. Another encryption is done using the receiver's public key. Final ciphertext can be decrypted only by the intended receiver who alone has the matching key. Thus confidentiality is provided.

Disadvantage: The public key algorithm which is complex must be executed four times rather than two in each communication.

* Applications of Public key Cryptosystem:
  - Encryption / Decryption
    Sender encrypts a message with the recipient's public key

- Digital Signature:
  Sender signs a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.
- Key exchange:
  Used to exchange a session key between the sender and the receiver.

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

* Requirements:
1. It is computationally easy for a party B to generate a pair of public key (PU_b) and a private key (PR_b)
2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext:
$$C = E(PU_b, M)$$
3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:
$$M = D[PR_b, C) = D[PR_b, E(PU_b, M)]$$
4. It is computationally infeasible for an adversary, knowing the public key, PU_b, to determine the private key, PR_b.
5. It is computationally infeasible for an adversary, knowing the public key, PU_b, and a ciphertext, C, to recover the original message M.
6. The sixth requirement is useful but not necessary for

all public-key applications. The two keys can be applied in either order:

$$M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$$

* One-way function

The requirements highlights the need for a trap-door one-way function. A one-way function maps a domain into a range such that every function value has a unique inverse, with the condition that the calculation of the function is easy, whereas the calculation of the inverse is infeasible i.e.,

$Y = f(x)$ is easy and
$X = f^{-1}(Y)$ is infeasible

Easy means that a problem can be solved in polynomial time as a function of input length.

ex: If length of input is n bits, then the time to compute the function is proportional to n^a, where a is a fixed constant.

In general, a problem is infeasible if the effort to solve it grows faster than polynomial time as a function of input size.

ex: If length of input is n bits and the time to compute the function is proportional to 2^n, then the problem is considered to be infeasible.

This computational complexity focuses on the worst case or average case complexity of an algorithm. These measures are inadequate for cryptography, which requires that it is infeasible to invert a function for virtually all inputs, not for the worse case or even average case.

Trap-door one-way function:

Trap-door one-way function is easy to calculate in on direction and infeasible to calculate in the other direction unless certain additional information is known.

With the additional information the inverse can be calculated in polynomial time.

A trapdoor one-way function is a family of invertible functions f_k such that

$Y = f_k(X)$ easy, if k and X are known

$X = f_k^{-1}(Y)$ easy, if k and Y are known

$X = f_k^{-1}(Y)$ infeasible, if Y is known but k is unknown.

Development of a practical public-key system depends on design of a suitable trap-door one-way function

- Public-key Cryptanalysis:

A public-key encryption is vulnerable to a brute-force attack. This can be overcomed by usage of large keys. The key size must be large enough to make brute-force attack impractical but small enough for practical encryption and decryption to avoid too slow execution. Public-key encryption is currently confined to key management and signature applications.

Probable-message attack is peculiar to public key systems. If a message sent is short, ex: 56-bit DES key, then the attacker can encrypt all possible 56-bit DES keys using the public key and discover the encrypted key by matching the transmitted ciphertext.

Thus even if the key size of the public key scheme is large, the attack is reduced to a brute force attack on a 56 bit key. This attack can be prevented by appending some random bits to such simple messages.

- **The RSA Algorithm:**

Rivest-Shamir-Adleman (RSA) is a general purpose-public key encryption technique. It was published in 1978 by Ron Rivest, Adi Shamir and Len Adleman at MIT.

RSA scheme is a block cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n. A typical size for n is 1024 bits ie., integer less than 2^{1024}

Encryption and decryption are given by:

$C = M^e \bmod n$

$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$

for some plaintext block M and ciphertext block C.

Here both sender and receiver must know the value of n. Sender knows the value of e and the receiver knows the value of d. Thus, this is a public-key encryption algorithm with a public key of $PU = \{e, n\}$ and private key of $PR = \{d, n\}$.

Relationship between e and d can be expressed as

$ed \bmod \phi(n) = 1$

or $ed \equiv 1 \bmod \phi(n)$

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

i.e., e and d are multiplicate inverses of mod $\phi(n)$. This is true only if $\gcd(\phi(n), d) = 1$.

- Requirements of RSA algorithm for public key encryption

1. It is possible to find values of e, d, n such that $M^{ed} \bmod n = M$ for all $M < N$.
2. It is relatively easy to calculate $M^e \bmod n$ and $C^d \bmod n$ for all values of $M < n$.
3. It is infeasible to determine d given e and n.

- RSA Algorithm:

- Key generation Alice:
  - Select p, q: p and q both prime, $p \neq q$
  - Calculate $n = p \times q$
  - Calculate $\phi(n) = (p-1)(q-1)$
  - Select integer e: $\gcd(\phi(n), e) = 1$; $1 < e < \phi(n)$
  - Calculate d: $d = e^{-1} \pmod{\phi(n)}$

- Public key: $PU = \{e, n\}$
- Private key: $PR = \{d, n\}$

- Encryption by Bob with Alice's Public key
  - Plaintext: $M < n$
  - Ciphertext: $C = M^e \bmod n$
- Decryption by Alice with Alice's Private key
  - Ciphertext: C
  - Plaintext: $M = C^d \bmod n$

Q1. Given $p = 17$ and $q = 11$, calculate public and private keys for RSA algorithm. Also find ciphertext for $M = 88$

1. $p = 17$, $q = 11$
2. calculate n:
   $n = p \times q = 17 \times 11 = \underline{187}$
3. calculate $\phi(n)$:
   $\phi(n) = (p-1)(q-1) = (17-1)(11-1) = 16 \times 10 = \underline{160}$
4. select integer e:
   $\gcd(\phi(n), e) = 1 \; ; \; 1 < e < \phi(n)$
   Select e such that e is relatively prime to $\phi(n) = 160$ and less than $\phi(n)$. Therefore $\underline{e = 7}$.
5. calculate d
   $d = e^{-1} \bmod \phi(n)$
   i.e., $de = \bmod \phi(n)$
   $de \equiv 1 \bmod 160$ and $d < 160$
   The correct value of $\underline{d = 23}$
   $23 \times 7 = 161 = 1 \times 160 + 1$
   d can be calculated using the Extended Euclid's algorithm
6. Public key: $PU = \{7, 187\}$
7. Private key: $PR = \{23, 187\}$
8. Ciphertext: $C = M^e \bmod n$
   $C = 88^7 \bmod 187$
   $= [(88^4 \bmod 187)(88^2 \bmod 187)(88^1 \bmod 187)] \bmod 187$

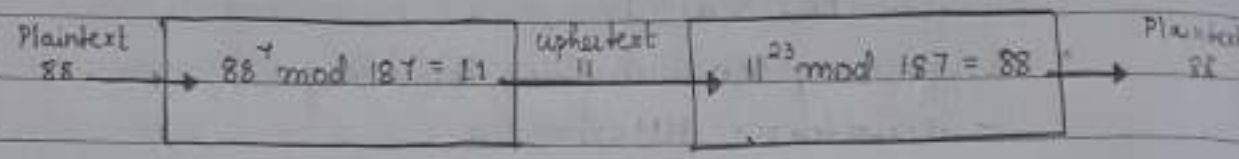
$88^1 \bmod 187 = 88$

$88^2 \bmod 187 = 7744 \bmod 187 = 77$

$88^4 \bmod 187 = 59969536 \bmod 187 = 132$

$\therefore 88^7 \bmod 187 = (88 \times 77 \times 132) = 894432 \bmod 187 = 11$

Therefore $C = 11$



For decryption

$M = C^d \bmod n$: Plaintext

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

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

$11^1 \bmod 187 = 11$

$11^2 \bmod 187 = 121$

$11^4 \bmod 187 = 14641 \bmod 187 = 55$

$11^8 \bmod 187 = 214358881 \bmod 187 = 33$

$\therefore 11^{23} \bmod 187 = (11 \times 121 \times 55 \times 33 \times 33) \bmod 187$

Therefore $P = 79720245 \bmod 187 = 88$

- Computational Aspects:

Algorithm for computing $a^b \bmod n$

```
c ← 0; f ← 1
for i ← k downto 0
    do c ← 2×c
       f ← (f×f) mod n
    if b_i = 1
    then c ← c+1
         f ← (f×a) mod n
return f
```

Value c is included for explanatory purpose. The final value of f is the value of the exponent.

- The security of RSA:

Four possible approaches to attack the RSA algorithm are

1. Brute force:

This involves trying all possible private keys

Defense against the brute force approach is to use a larger key space (just like other cryptosystems). Thus, the larger the number of bits in d, the better. However, because the calculations are involved such as in key generation and in encryption / decryption that are complex, the larger the size of the key, the slower the system will be.

2. Mathematical Attacks:

There are several approaches, all equivalent in effort to factoring the product of two primes.

a. Factor n into its two prime factors.

This enables calculation of $\phi(n) = (p-1)(q-1)$, which in turn enables determination of $d \equiv e^{-1} \bmod \phi(n)$.

b. Determine $\phi(n)$ directly, without first determining p and q. This enables determination of $d \equiv e^{-1} \bmod \phi(n)$.

c. Determine d directly, without first determining $\phi(n)$.

3. Timing Attacks:

These depend on the running time of the decryption algorithm. Attacker can determine a private key by keeping track of how long a computer takes to decipher messages. It is a ciphertext only attack. Applicable to all public-key cryptography systems.

Countermeasures to timing attack:

a. Constant Exponentiation time: Ensure that all exponentiations take the same amount of time before returning a result.

b. Random delay: Adding a random delay to the exponentiation algorithm to confuse the attacker.

c. Binding: Multiply the ciphertext by random number before performing exponentiation. This prevents the attacker from knowing what ciphertext bits are being processed inside the computer and therefore prevents the bit-by-bit analysis essential to the timing attack.

- Numericals:

Q2: Perform encryption and decryption using the RSA algorithm for the following:

a. p = 3 ; q = 11 ; e = 7 ; M = 5

1. p = 3 ; q = 11
2. Calculate $n = p \times q$

   $n = 3 \times 11 = 33 \Rightarrow n = 33$
3. Calculate $\phi(n) = (p-1)(q-1)$

   $\phi(n) = (3-1)(11-1) = 2 \times 10 = 20 \Rightarrow \phi(n) = 20$
4. e = 7
5. Calculate d : $de = 1 \bmod \phi(n)$

   $de = 1 \bmod 20$

   $d = 3 ; \quad 3 \times 7 = 21 \bmod 20 = 1 \Rightarrow d = 3$
6. Public key : $PU = \{e, n\}$

   $PU = \{7, 33\}$
7. Private key : $PR = \{d, n\}$

   $PR = \{3, 33\}$
8. ciphertext : (Encryption)

   $C = M^e \bmod n$

   $C = 5^7 \bmod 33$

   $C = [(5^4 \bmod 33)(5^2 \bmod 33)(5^1 \bmod 33)] \bmod 33$

   $5^1 \bmod 33 = 5$

   $5^2 \bmod 33 = 25 \bmod 33 = 25$

   $5^4 \bmod 33 = 625 \bmod 33 = 31$

   $\therefore C = (5 \times 25 \times 31) \bmod 33 = 3875 \bmod 33$

   $\Rightarrow C = 14$

9. Plaintext: (Decryption)

$M = C^d \bmod n$

$M = 14^3 \bmod 33$

$M = [(14^2 \bmod 33)(14^1 \bmod 33)] \bmod 33$

$14^1 \bmod 33 = 14$

$14^2 \bmod 33 = 196 \bmod 33 = 31$

$\therefore M = (14 \times 31) \bmod 33 = 434 \bmod 33$

$\Rightarrow M = 5$

b. $p = 5 ; q = 11 ; e = 3 ; M = 9$

1. $p = 5 ; q = 11$
2. Calculate $n = p \times q$

   $n = 5 \times 11 = 55 \quad \Rightarrow n = 55$
3. Calculate $\phi(n) = (p-1)(q-1)$

   $\phi(n) = (5-1)(11-1) = 4 \times 10 = 40 \quad \Rightarrow \phi(n) = 40$
4. $e = 3$
5. Calculate d: $de = 1 \bmod \phi(n)$

   $de = 1 \bmod 40$

   $27 \times 3 = 81 \bmod 40 = 1 \quad \Rightarrow d = 27$
6. Public key: $PU = \{e, n\} \quad \Rightarrow PU = \{3, 55\}$
7. Private key: $PR = \{d, n\} \quad \Rightarrow PR = \{27, 55\}$
8. Encryption: Ciphertext: $C = M^e \bmod n$

   $C = 9^3 \bmod 55$

   $C = [(9^2 \bmod 55)(9^1 \bmod 55)] \bmod 55$

   $9^1 \bmod 55 = 9$

   $9^2 \bmod 55 = 81 \bmod 55 = 26$

   $\therefore C = (26 \times 9) \bmod 55 = 234 \bmod 55$

   $\Rightarrow C = 14$
9. Decryption: Plaintext: $M = C^d \bmod n$

   $M = 14^{27} \bmod 55$

   $M = [(14^8 \bmod 55)(14^8 \bmod 55)(14^8 \bmod 55)(14^2 \bmod 55)$

   $(14^1 \bmod 55)] \bmod 55$

$14^1 \bmod 55 = 14$

$14^2 \bmod 55 = 196 \bmod 55 = 31$

$14^8 \bmod 55 = 14757890 56 \bmod 55 = 16$

$\therefore M = (16 \times 16 \times 16 \times 31 \times 14) \bmod 55 = 1777664 \bmod 55$

$\Rightarrow M = 9$

c. $p = 7$; $q = 11$; $e = 17$; $M = 8$

1. $p = 7$; $q = 11$
2. Calculate $n = p \times q$

   $n = 7 \times 11 = 77$ $\Rightarrow n = 77$
3. Calculate $\phi(n) = (p-1)(q-1)$

   $\phi(n) = (7-1)(11-1) = 6 \times 10 = 60$ $\Rightarrow \phi(n) = 60$
4. $e = 17$
5. Calculate d: $de = 1 \bmod \phi(n)$

   $de = 1 \bmod 60$

   Since gcd = 1, the extended Euclidean algorithm can be used

q	r_2	r_1	r	t_1	t_2	$t = t_1 - q t_2$
3	60	17	9	0	1	$t = 0 - 3(1) = -3$
1	17	9	8	1	-3	$t = 1 - 1(-3) = 4$
1	9	8	1	-3	4	$t = -3 - 1(4) = -7$
8	8	1	0	4	-7	$t = 4 - 8(-7) = 60$
-	1	0	-	-7	60	

→ inverse of 17 mod 60

$\Rightarrow d = 53$ $(-7 \bmod 60 = -7 + 60 = 53)$

$\gcd(\phi(n), e) = \gcd(60, 17)$

17) 60 (3
-51
9) 17 (1
-9
8) 9 (1
-8
1) 8 (8
gcd -8
0

6. Public key : $PU = \{e, n\}$ $\Rightarrow PU = \{17, 77\}$
7. Private key : $PR = \{d, n\}$ $\Rightarrow PR = \{53, 77\}$
8. Encryption : ciphertext : $C = M^e \bmod n$

   $C = 8^{17} \bmod 77$

   $C = [(8^8 \bmod 77)(8^8 \bmod 77)(8^1 \bmod 77)] \bmod 77$

$8^1 \bmod 77 = 8$

$8^5 \bmod 77 = 167772 16 \bmod 77 = 71$

$\therefore C = (71 \times 71 \times 8) \bmod 77 = 40328 \bmod 77$ •

$\Rightarrow C = 57$

9. Decryption: Plaintext: $M = C^d \bmod n$

$M = 57^{53} \bmod 77$

$M = [(57^4 \bmod 77)(57^4 \bmod 77)(57^4 \bmod 77)(57^4 \bmod 77)$
$(57^4 \bmod 77)(57^4 \bmod 77)(57^4 \bmod 77)(57^4 \bmod 77)$
$(57^4 \bmod 77)(57^4 \bmod 77)(57^4 \bmod 77)(57^4 \bmod 77)$
$(57^4 \bmod 77)(57^1 \bmod 77)] \bmod 77$

$57^1 \bmod 77 = 57$

$57^4 \bmod 77 = 10556001 \bmod 77 = 71$

$\therefore M = [71 \times 71 \times 71 \times 71 \times 71 \times 71 \times 71 \times 71 \times 71 \times 71 \times 71 \times 71 \times 71$
$\times 57] \bmod 77$

$M = [71^{13} \times 57] \bmod 77$

$M = [(71^8 \bmod 77)(71^4 \bmod 77)(71^4 \bmod 77)(71^1 \bmod 77)$
$(57^1 \bmod 77)] \bmod 77$

$71^1 \bmod 77 = 71$

$71^4 \bmod 77 = 25411681 \bmod 77 = 64$

$57^1 \bmod 77 = 57$

$\therefore M = (64 \times 64 \times 64 \times 71 \times 57) \bmod 77$

$M = 1060896768 \bmod 77$

$\Rightarrow M = 8$

d. $p = 11;\ q = 13,\ e = 11,\ M = 7$

1. $p = 11;\ q = 13$
2. Calculate $n = p \times q$

$n = 11 \times 13 = 143 \quad \Rightarrow n = 143$

3. Calculate $\phi(n) = (p-1)(q-1)$

$\phi(n) = (11-1)(13-1) = 10 \times 12 = 120 \quad \Rightarrow \phi(n) = 120$

4. $e = 11$

5. Calculate d : $de = 1 \bmod \phi(n)$

$de = 1 \bmod 120 \quad (d \neq e) \Rightarrow d \neq 11$

$d = 11 + 120 = 131 \quad \Rightarrow d = 131$

6. Public key $PU = \{e, n\} \quad \Rightarrow PU = \{11, 143\}$

7. Private key $PR = \{d, n\} \quad \Rightarrow PR = \{131, 143\}$

8. Encryption: Ciphertext : $C = M^e \bmod n$

$C = 7^{11} \bmod 143$

$C = [(7^8 \bmod 143)(7^2 \bmod 143)(7^1 \bmod 143)] \bmod 143$

$7^1 \bmod 143 = 7$

$7^2 \bmod 143 = 49 \bmod 143 = 49$

$7^8 \bmod 143 = 5764801 \bmod 143 = 42$

$\therefore C = (42 \times 49 \times 7) \bmod 143 = 14406 \bmod 143$

$\Rightarrow \underline{C = 106}$

9. Decryption: Plaintext : $M = C^d \bmod n$

$M = 106^{131} \bmod 143$

$M = [(106^4 \bmod 143)^{32} (106^2 \bmod 143)(106^1 \bmod 143)] \bmod 143$

$106^1 \bmod 143 = 106$

$106^2 \bmod 143 = 11236 \bmod 143 = 82$

$106^4 \bmod 143 = 126247696 \bmod 143 = 3$

$\therefore M = (3^{32} \times 82 \times 106) \bmod 143$

$M = [(3^{16} \bmod 143)(3^{16} \bmod 143) \times 82 \times 106] \bmod 143$

$3^{16} \bmod 143 = 43046721 \bmod 143 = 3$

$\therefore M = (3 \times 3 \times 82 \times 106) \bmod 143 = 78228 \bmod 143$

$\Rightarrow \underline{M = 7}$

c $\underline{p = 17;\ q = 31;\ e = 7;\ M = 2}$

1. $p = 17,\ q = 31$

2. Calculate $n = p \times q$

$n = 17 \times 31 = 527 \quad \Rightarrow n = 527$

3. Calculate $\phi(n) = (p-1)(q-1)$

$\phi(n) = (17-1)(31-1) = 16 \times 30 = 480 \quad \Rightarrow \phi(n) = 480$

4. $e = 7$

5. calculate d: $de = 1 \bmod \phi(n)$

$de = 1 \bmod 480$

Since gcd = 1, the extended Euclidean algorithm can be used

q	r_2	r_1	r	t_1	t_2	$t = t_1 - q t_2$
68	480	7	4	0	1	$t = 0 - 68(1) = -68$
1	7	4	3	1	-68	$t = 1 - 1(-68) = 69$
1	4	3	1	-68	69	$t = -68 - 1(69) = -137$
3	3	1	0	69	-137	$t = 69 - 3(-137) = 480$
-	1	0	-	-137	480	

$\gcd(\phi(n), e) = \gcd(480, 7)$

$7 \overline{)480\ (68}$ -476

$4 \overline{)7\ (1}$ -4

$3 \overline{)4\ (1}$ -3

$\gcd \quad 1 \overline{)3\ (3}$ -3 0

$\Rightarrow d = 343$ ↳ inverse of $7 \bmod 480 \Rightarrow -137 \bmod 480 = 343$

6. Public key: $PU = \{e, n\}$ $\Rightarrow PU = \{7, 527\}$

7. Private key: $PR = \{d, n\}$ $\Rightarrow PR = \{343, 527\}$

8. Encryption: Ciphertext: $C = M^e \bmod n$

$C = 2^7 \bmod 527$

$C = 128 \bmod 527 = 128$ $\Rightarrow \underline{C = 128}$

9. Decryption: Plaintext: $M = C^d \bmod n$

$M = 128^{343} \bmod 527$

$M = [(128^8 \bmod 527)^{85} (128^2 \bmod 527)(128^1 \bmod 527)] \bmod 527$

$128^1 \bmod 527 = 128$

$128^2 \bmod 527 = 16384 \bmod 527 = 47$

$128^8 \bmod 527 = 268435456 \bmod 527 = 101$

$M = [101^{85} \times 47 \times 128] \bmod 527$

$M = [(101^4 \bmod 527)^{21} (101^1 \bmod 527) \times 47 \times 128] \bmod 527$

$101^1 \bmod 527 = 101$

$101^4 \bmod 527 = 104060401 \bmod 527 = 35$

$M = [35^{21} \times 101 \times 47 \times 128] \bmod 527$

$M = [(35^4 \bmod 527)^5 (35^1 \bmod 527) \times 101 \times 47 \times 128] \bmod 527$

$35^1 \bmod 527 = 35$

$35^4 \bmod 527 = 1500625 \bmod 527 = 256$

$M = [256^5 \times 35 \times 101 \times 47 \times 128] \bmod 527$

$M = [(256^4 \bmod 527)(256^1 \bmod 527) 35 \times 101 \times 47 \times 128] \bmod 527$

$256^4 \bmod 527 = 4294967296 \bmod 527 = 35$

$256^1 \bmod 527 = 256$

$\therefore M = [35 \times 256 \times 35 \times 101 \times 47 \times 128] \bmod 527$

$35^2 \bmod 527 = 1225 \bmod 527 = 171$

$M = [171 \times 256 \times 101 \times 47 \times 128] \bmod 527$

$(171 \times 256) \bmod 527 = 43776 \bmod 527 = 35$

$M = [35 \times 101 \times 47 \times 128] \bmod 527$

$M = 21266560 \bmod 527$

$\Rightarrow M = 2$

Q3: In a public-key system using RSA, you intercept the ciphertext $c = 10$ sent to a user whose public key is $e = 5$, $n = 35$, what is the plaintext M ?

— Given: RSA system

$c = 10$; $e = 5$; $n = 35$

wkt: $n = p \times q \Rightarrow p \times q = 35$

$\therefore p = 7$ $q = 5$

1. $p = 7$; $q = 5$
2. $n = 35$
3. calculate $\phi(n) = (p-1)(q-1)$

   $\phi(n) = (7-1)(5-1) = 6 \times 4 = 24 \Rightarrow \phi(n) = 24$
4. $e = 5$
5. Calculate d: $de = 1 \bmod \phi(n)$

   $de = 1 \bmod 24$ $\quad d \neq e \Rightarrow d \neq 5$

   $\therefore d = 5 + 24 = 29$
6. Public key: $PU = \{e, n\}$ $\Rightarrow$ $PU = \{5, 35\}$
7. Private key: $PR = \{d, n\}$ $\Rightarrow$ $PR = \{29, 35\}$
8. Encryption: ciphertext: $C = 10$ $C = M^e \bmod n$
9. Decryption: Plaintext: $M = C^d \bmod n$

$M = 10^{29} \bmod 35$

$M = [(10^8 \bmod 35)(10^8 \bmod 35)(10^8 \bmod 35)$
$(10^4 \bmod 35)(10^1 \bmod 35)] \bmod 35$

$10^1 \bmod 35 = 10$

$10^4 \bmod 35 = 10000 \bmod 35 = 25$

$10^8 \bmod 35 = 100000000 \bmod 35 = 30$

$\therefore M = [30 \times 30 \times 30 \times 25 \times 10) \bmod 35 = 6750000 \bmod 35$

$\Rightarrow M = 5$

- **Key Management:**

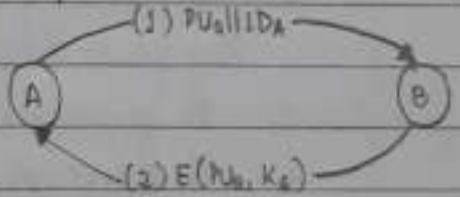
Key distribution is the function that delivers a key to two parties for exchanging secure encrypted data. Some sort of mechanism or protocol is used for the secure distribution of keys.

– Symmetric Key Distribution using Asymmetric Encryption:

- Simple Secret Key Distribution:

By Merkle

Simple use of Public-key Encryption to establish a session key.



If A wishes to communicate with B, the following procedure is employed.

1. A generates a public/private key pair {PUa, PRa} and transmits a message to B consisting of PUa and an identifier of A, IDA.
2. B generates a secret key, Ks, and transmits it to A, which is encrypted with A's public key.
3. A computes D(PRa, E(PUa, Ks)) to recover the secret key. Because only A can decrypt the message, only A and B will know the identity of Ks.
4. A discards PUa and PRa and B discards PUa.

note: || ⇒ x||y means that x is concatenated with y

A and B can now securely communicate using conventional encryption and the session key Ks. At the completion of the exchange, both A and B discard Ks.

It is a simple and attractive protocol. No key exists before the start of the communication and none exist after the completion of communication. Thus, the risk of compromise of the keys is minimal. At the same time, the communication is secure from eavesdropping.

This protocol is insecure against an adversary who can intercept messages and substitute another message. Such an attack is known as a man-in-the-middle attack.

In the case of a man in the middle attack, if an adversary E, has control of the intervening communication channel, then E can compromise the communication between A and B in the following way without being detected.

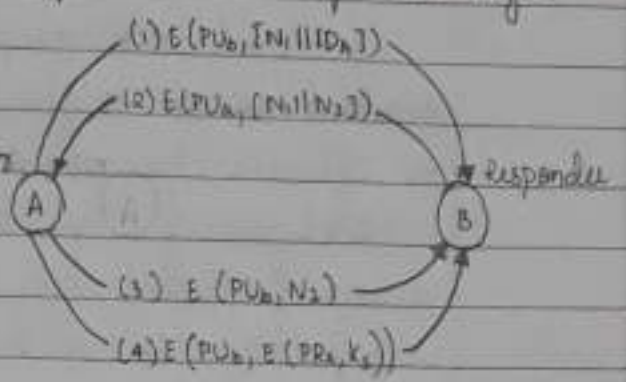
1. A generates a public/private key pair $\{PU_a, PR_a\}$ and transmits a message intended for B consisting of PU_a and an identifier of A, ID_A.
2. E intercepts the message, creates its own public/private key pair $\{PU_e, PR_e\}$ and transmits $PU_e \| ID_A$ to B.
3. B generates a secret key, K_s and transmits $E(PU_e, K_s)$.
4. E intercepts the message and learns K_s by computing $D(PR_e, E(PU_e, K_s))$.
5. E transmits $E(PU_a, K_s)$ to A

The result is that both A and B know K_s and are not aware that K_s has also been revealed to E. A and B can now exchange messages using K_s. E no longer actively interferes with the communication channel by simply eavesdrops. Knowing K_s, E can decrypt all the messages.

- Secret Key Distribution with Confidentiality and Authentication

This scheme provides protection against both active and passive attacks. It is assumed that A and B have already exchanged public keys by a suitable scheme. This scheme ensures both confidentiality and authentication in the exchange of secret key

Public key distribution of secret keys



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

- Hybrid Scheme:

This scheme retains the use of a key distribution center (KDC) that shares a secret master key with each user. A public key scheme is used to distribute the master keys. KDC distributes secret session keys encrypted with the master key.

Advantages:

- Performance: There are many applications especially transaction-oriented applications, in which the session keys change frequently. Distribution of session keys by public-key encryption could degrade overall system performance because of the relatively high computational load of public key encryption and decryption. With a hybrid approach, public key encryption is used only occasionally to update the master key between a user and the KDC.

- Backward compatibility: The hybric scheme is easily overlaid on an existing KDC scheme with minimal disruption or software changes.

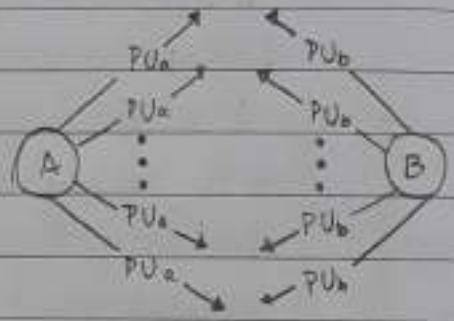
Addition of a public key layer provides a secure, efficient means of distributing master keys. This is an advantage in a configuration in which a single KDC serves a widely distributed set of users.

- Distribution of Public Keys:

Several techniques have been proposed for the distribution of public keys. They can be grouped into the following general schemes

• Public Announcement:

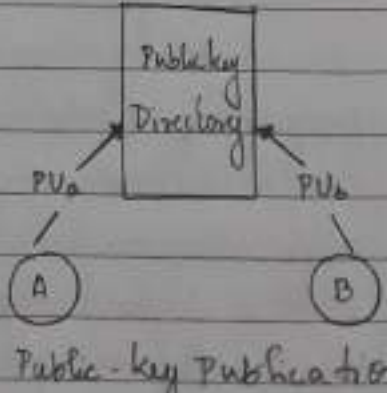
Any participant can send his public key to the community at large using public-key algorithm. Although this approach is convenient, it has a major weakness. Anyone can forge such a public announcement. Some user could pretend to be user A and send a public key to another participant or broadcast such a public key. Until such time as user A discovers the forgery and alerts other participants, forger is able to read all encrypted messages intended for A and can use the forged keys for authentication.



Uncontrolled Public-key Distribution

• Publicly Available Directory:

A greater degree of security can be achieved as compared to the public announcement approach by maintaining a publicly available dynamic directory of public keys.



Public-key Publication

Maintenance and distribution of the public directory is the responsibility of some trusted entity or organization.

Elements:

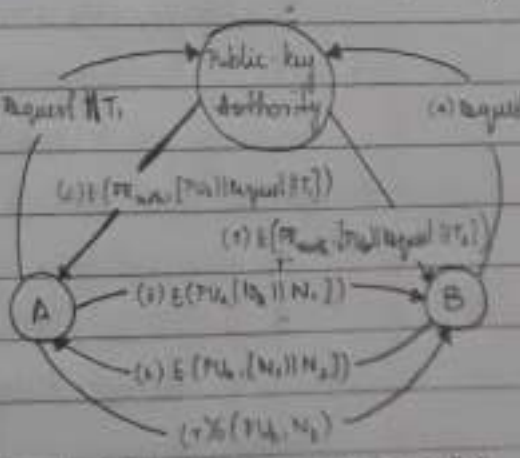
1. Directory Authority: maintains a directory with a ⟨name, public key⟩ entry for each participant.
2. Each participant registers a public key with the directory authority. Registration must be in person or by some form of secure authenticated communication.
3. A participant may replace the existing key with a new one at any time, either because of the desire to replace a public key that has already been used for a large amount of data, or because the corresponding private key has been compromised in some way.
4. Participants can also access the directory electronically. For this purpose, secure, authenticated communication from the authority to the participant is mandatory.

Vulnerabilities:

- If an adversary succeeds in obtaining or computing the private key of the directory authority, adversary can authoritatively distribute counterfeit public keys and subsequently impersonate any participant and eavesdrop on messages sent to any participant.
- Adversary may also tamper with the records kept by the authority.

- Public-key Authority:

It provides stronger security for public-key distribution by tighter control over the distribution of public keys from the directory using "Public key Authority".



Public-key Distribution scenario using public key authority

This scheme also assumes that a central authority maintains a dynamic directory of public keys of all participants. Each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key.

Steps:

1. Request $\| T_1$

A sends a timestamped message to the public-key authority containing a request for the current public key of B

2. $E(PR_{auth}, [PU_b \| Request \| T_1])$

The authority responds with a message that is encrypted using the authority's private key, PR_{auth}. Thus A is able to decrypt the message using the authority's public key. Therefore, A is assured that the message originated with the authority. The message includes the following:

- B's public key, PU_b, which A can use to encrypt messages destined for B
- The original request used to enable A to match this response with the corresponding earlier request and to verify that the original request was not altered before reception by the authority.
- The original timestamp given so that A can determine that this is not an old message from the authority containing a key other than B's current public key

3. $E(PU_b, [ID_A \| N_1])$

A stores B's public key and also uses it to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1) which is used to identify this transaction uniquely

4. Request $\| T_2$

5. $E(PR_{auth}, [PU_a \| Request \| T_2])$

B retrieve A's public key from the authority in the same manner as A retrieved B's public key

6. $E(PU_a, [N_1 \| N_2])$

B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as a new nonce generated by B (N_2). Because only B could have decrypted message (3), the presence of N_1 in message (6) assures A that the correspondent is B.

7. $E(PU_b, N_2)$

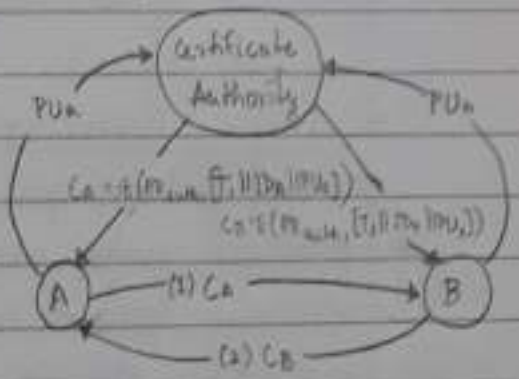
A returns N_2, which is encrypted using B's public key, to assure B that its correspondent is A.

Disadvantages:

- Public-key authority can be a bottle-neck in the system, because, a user must appeal to the authority for a public key for every other user that it wishes to contact.
- The directory of names and public keys maintained by the authority is vulnerable to tampering.

- Public Key Certificates:

Each participant applies to the certificate authority, supplying a public key and requesting a certificate. Application must be in person or by some form of secure authenticated communication.



Exchange of Public-key certificates.

For participant A, the authority provides a certificate of the form $C_A = E(PR_{auth}, [T \| ID_A \| PU_a])$ where PR_{auth} is the private key used by the authority and T is a timestamp.

A may then pass this certificate on to any other participant, who reads and verifies the certificate as:

$D(PU_{auth}, C_A) = D(PU_{auth}, E(PR_{auth}, [T \| ID_A \| PU_a])) = (T \| ID_A \| PU_a)$

The recipient uses the authority's public key, PU_{auth}, to decrypt the certificate. Because the certificate is readable only

using the authority's public key, this verifies that the certificate came from the certificate authority. The elements ID_A and PU_A provide the recipient with the name and public key of the certificate's holder. The timestamp T validates the currency of the certificate.

Use of timestamp:

Suppose that A's private key is learned by an adversary.

- A generates a new private/public key pair and applies to the certificate authority for a new certificate.
- The adversary replays the old certificate to B.
- If B then encrypts messages using the compromised old public key, adversary can read those messages.
- There is risk involved until all possible communicants are aware that the old certificate is obsolete. Thus, timestamp serves as an expiration date.
- If a certificate is sufficiently old, it is assumed to be expired.

— Diffie-Hellman Key Exchange:

It is a simple public key algorithm. Many commercial products make use of this key exchange technique. Diffie Hellman protocol has been applied to many security protocols including the Secure Socket Layer (SSL) and Transport Layer Security (TLS).

This protocol enables two users to establish a secret key using a public-key scheme based on discrete logarithms. It is secure only if the authenticity of the two participants can be established. Otherwise it is vulnerable to man-in-the-middle attack.

Purpose of this algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. It is limited to the exchange of secret values

The algorithm depends for its effectiveness on the difficulty of computing discrete logarithms.

Discrete Logarithm

A primitive root of a prime number is one whose powers modulo p generate all the integers from 1 to p-1, i.e., if 'a' is a primitive root of the prime number 'p', then the numbers

$a \bmod p, a^2 \bmod p, \ldots\ldots, a^{p-1} \bmod p$

are distinct and consist of the integers from 1 to p-1 in some permutation.

For any integer 'b' and a primitive root 'a' of prime number 'p', we can find a unique exponent i such that:

$b = a^i \pmod p$ where $0 \leq i \leq (p-1)$.

Exponent i is referred to as the discrete logarithm of b for the base a, mod p. We express this value as $dlog_{a,p}(b)$

Algorithm:

- There are two publicly known numbers
  - a prime number q
  - an integer α that is a primitive root of q
- Suppose the users A and B wish to exchange a key:
  - User A selects a random integer $X_A < q$ and computes
    $Y_A = \alpha^{X_A} \bmod q$
  - User B independently selects a random integer $X_B < q$ and computes $Y_B = \alpha^{X_B} \bmod q$

Each side keeps the X value private and makes the Y value available publicly to the other side

- User A computes the key as:
  $k = (Y_B)^{X_A} \bmod q$
- User B computes the key as
  $k = (Y_A)^{X_B} \bmod q$

These two calculations produce identical results:

$$
\begin{aligned}
K &= (Y_B)^{X_A} \bmod q \\
&= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\
&= (\alpha^{X_B})^{X_A} \bmod q \\
&= \alpha^{X_A X_B} \bmod q \\
&= (\alpha^{X_A})^{X_B} \bmod q \\
&= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\
&= (Y_A)^{X_B} \bmod q
\end{aligned}
$$

The result is that the two sides have exchanged a secret value. Furthermore, because X_A and X_B are private, an adversary only has the following ingredients to work with: q, α, Y_A, Y_B

Thus the adversary is forced to take a discrete logarithm to determine the key. Ex: to determine the private key of user B, an adversary must compute $X_B = \text{dlog}_{\alpha,q}(Y_B)$

The adversary can the compute the key K in the same manner as user B calculates it.

Diffie-Hellman Key Exchange Algorithm:

- Global Public Elements:
  - q : prime number
  - α : $\alpha < q$, α a primitive root of q
- User A Key Generation:
  - select private X_A : $X_A < q$
  - calculate public Y_A : $Y_A = \alpha^{X_A} \bmod q$
- User B Key Generation:
  - select private X_B : $X_B < q$
  - calculate public Y_B : $Y_B = \alpha^{X_B} \bmod q$
- Calculation of secret key by User A
  - $K = (Y_B)^{X_A} \bmod q$
- Calculation of secret key by User B
  - $K = (Y_A)^{X_B} \bmod q$

Security:

Security of Diffie-Hellman key exchange lies in the fact that, while it is relatively easy to calculate exponentials modulo a prime, it is very difficult to calculate discrete logarithms. For large primes, the latter task is considered infeasible.

Example: Key Exchange

- prime number $q = 353$
- primitive root of q : $\alpha = 3$
- secret keys: $X_A = 97$ $X_B = 233$

Each completes its public key as:

A: $Y_A = \alpha^{X_A} \bmod q$

$Y_A = 3^{97} \bmod 353 = [(3^{16} \bmod 353)^6 (3^1 \bmod 353)] \bmod 353$

$3^{16} \bmod 353 = 43046721 \bmod 353 = 136$

$3^1 \bmod 353 = 3$

$\therefore Y_A = (136^6 \times 3) \bmod 353$

$Y_A = [(136^4 \bmod 353)(136^2 \bmod 353) \times 3] \bmod 353$

$136^2 \bmod 353 = 18496 \bmod 353 = 140$

$136^4 \bmod 353 = 342102016 \bmod 353 = 185$

$Y_A = (185 \times 140 \times 3) \bmod 353 = 77700 \bmod 353$

$\therefore \underline{Y_A = 40}$

B: $Y_B = \alpha^{X_B} \bmod q$

$Y_B = 3^{233} \bmod 353 = [(3^{16} \bmod 353)^{14} (3^8 \bmod 353)(3^1 \bmod 353)] \bmod 353$

$3^{16} \bmod 353 = 136$

$3^8 \bmod 353 = 6561 \bmod 353 = 207$

$3^1 \bmod 353 = 3$

$Y_B = (136^{14} \times 207 \times 3) \bmod 353$

$Y_B = [(136^4 \bmod 353)^3 (136^2 \bmod 353) \times 207 \times 3] \bmod 353$

$136^4 \bmod 353 = 185$

$136^2 \bmod 353 = 140$

$Y_B = [185^3 \times 140 \times 207 \times 3] \bmod 353$

$Y_B = [(185^2 \bmod 353)(185^1 \bmod 353) 140 \times 207 \times 3] \bmod 353$

$185^1 \bmod 353 = 185$

$185^2 \bmod 353 = 34225 \bmod 353 = 337$

$\therefore Y_B = [337 \times 185 \times 140 \times 207 \times 3] \bmod 353$

$Y_B = [(1806758100 \bmod 353) \times 3] \bmod 353$

$Y_B = 318 \times 3 \bmod 353 = 954 \bmod 353$

$\therefore Y_B = 248$

After they exchange public keys each can compute the common secret key.

A: $K = (Y_B)^{X_A} \bmod 353$

$K = (248)^{97} \bmod 353$

$K = [(248^4 \bmod 358)^{24} (248^1 \bmod 353)] \bmod 353$

$248^1 \bmod 353 = 248$

$248^4 \bmod 353 = 3782742016 \bmod 353 = 17$

$K = [17^{24} \times 248] \bmod 353$

$K = [(17^8 \bmod 353)^3 \times 248] \bmod 353$

$17^8 \bmod 353 = 6975757441 \bmod 353 = 185$

$K = [185^3 \times 248] \bmod 353$

$K = 15702430000 \bmod 353$

$\therefore K = 160$

B: $K = (Y_A)^{X_B} \bmod 353$

$K = (40)^{233} \bmod 353$

$K = [(40^8 \bmod 353)^{29} (40^1 \bmod 353)] \bmod 353$

$40^8 \bmod 353 = 40$

$40^9 \bmod 353 = 655360000000 \bmod 353 = 171$

$K = (171^{29} \times 40) \bmod 353$

$K = [(171^4 \bmod 353)^7 (171^1 \bmod 353) \times 40] \bmod 353$

$171^4 \bmod 353 = 171$

$171^4 \bmod 353 = 855036081 = 187$

$K = (187^7 \times 171 \times 40) \bmod 353$

$K = [(187^4 \bmod 353)(187^2 \bmod 353)(187^1 \bmod 353) \times 171 \times 40] \bmod 353$

$187^1 \bmod 353 = 187$

$187^2 \bmod 353 = 34969 \bmod 353 = 22$

$187^4 \bmod 353 = 1222830961 \bmod 353 = 131$

$\therefore K = (131 \times 22 \times 187 \times 171 \times 40) \bmod 353$

$K = 36863085600 \bmod 353$

$\therefore \underline{K = 160}$

Man-in-the-Middle Attack:

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

1. Darth prepares for an attack by generating two random private key X_{D1} and X_{D2} and then computing the corresponding public keys Y_{D1} and Y_{D2}.
2. Alice transmits Y_A to Bob.
3. Darth intercepts Y_A and transmits Y_{D1} to Bob. Darth also calculates $K2 = (Y_A)^{X_{D2}} \bmod q$.
4. Bob receives Y_{D1} and calculates $K1 = (Y_{D1})^{X_B} \bmod q$.
5. Bob transmits Y_B to Alice.
6. Darth intercepts Y_B and transmits Y_{D2} to Alice. Darth calculates $K_1 = (Y_B)^{X_{D1}} \bmod q$.
7. Alice receives Y_{D2} and calculates $K2 = (Y_{D2})^{X_A} \bmod q$.

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key K_1 and Alice and Darth share secret key K_2.

All future communication between Bob and Alice is compromised in the following way:

1. Alice sends an encrypted message M: $E(K_2, M)$.
2. Darth intercepts the encrypted message and decrypts it to recover M.
3. Darth sends Bob $E(K_1, M)$ or $E(K_1, M')$, where M' is any message. In first case, Darth simply wants to eavesdrop on the communication without altering it. In the second case, Darth wants to modify the message going to Bob.

The key exchange protocol is vulnerable to such an attack because it does not authenticate the participants. This vulnerability can be overcome with the use of digital signatures and public-key certificates.

Q4: Users A and B use the Diffie-Hellman key exchange technique with a common prime $q = 11$ and a primitive root $\alpha = 5$

i.) If user A has private key $X_A = 3$. What is A's public key Y_A?

ii) If user B has private key $X_B = 2$. What is B's public key Y_B?

iii) What is the shared secret key K_A and K_B?

— Given:

$q = 11$; $\alpha = 5$; $X_A = 3$; $X_B = 2$

i. A's public key Y_A:

$$Y_A = \alpha^{X_A} \bmod q$$

$$Y_A = 5^3 \bmod 11 = 125 \bmod 11$$

$$\therefore Y_A = 4$$

ii. B's public key Y_B:

$$Y_B = \alpha^{X_B} \bmod q$$

$$Y_B = 5^2 \bmod 11 = 25 \bmod 11$$

$$\therefore Y_B = 3$$

iii. Shared secret key K_A and K_B:

$$K_A = (Y_B)^{X_A} \bmod q$$

$$K_A = 3^3 \bmod 11 = 27 \bmod 11 = 5$$

$$\therefore K_A = 5$$

$$K_B = (Y_A)^{X_B} \bmod q$$

$$K_B = 4^2 \bmod 11 = 16 \bmod 11$$

$$\therefore K_B = 5$$

Q5: In Diffie-Hellman key exchange, $q = 71$, its primitive root $\alpha = 7$. A's private key is 5, B's private key is 12. Find:

i) A's public key ii) B's public key iii) Shared secret key

- Given: $q = 71$; $\alpha = 7$; $X_A = 5$; $X_B = 12$

i. A's public key:

$Y_A = \alpha^{X_A} \bmod q$

$Y_A = 7^5 \bmod 71 = 16807 \bmod 71$

$\therefore Y_A = 51$

ii. B's public key:

$Y_B = \alpha^{X_B} \bmod q$

$Y_B = 7^{12} \bmod 71 = 13841287200 \bmod 71$

$\therefore Y_B = 4$

iii. Shared secret key:

$K_A = (Y_B)^{X_A} \bmod q$

$K_A = (4)^5 \bmod 71 = 1024 \bmod 71$

$\therefore K_A = 30$

$K_B = (Y_A)^{X_B} \bmod q$

$K_B = 51^{12} \bmod 71 = [(51^4 \bmod 71)^3] \bmod 71$

$51^4 \bmod 71 = 6765201 \bmod 71 = 37$

$\therefore K_B = (37)^3 \bmod 71 = 50653 \bmod 71$

$\therefore K_B = 30$

Q6 Assume 7 is a primitive root of 29, let Alice choose $a = 3$, let Bob choose $b = 10$. Find the keys that are exchanged by Alice and Bob. What is the secret key generated by Bob and Alice.

- Given: $q = 29$, $\alpha = 7$; $X_A = 3$; $X_b = 10$

A's public key:

$Y_A = \alpha^{X_A} \bmod q$

$Y_A = 7^3 \bmod 29 = 343 \bmod 29 = 24$

$\therefore Y_A = 24$

B's public key:

$Y_B = \alpha^{X_B} \bmod q$

$Y_B = 7^{10} \bmod 29 = 282475249 \bmod 29$

$\therefore Y_B = 24$

Shared Secret key

$K_A = (Y_B)^{X_A} \bmod q$

$K_A = (24)^3 \bmod 29 = 13824 \bmod 29$

$\therefore K_A = 20$

$K_B = (Y_A)^{X_B} \bmod q$

$K_B = (24)^{10} \bmod 29 = [(24^4 \bmod 29)^2 (24^2 \bmod 29)] \bmod 29$

$24^2 \bmod 29 = 576 \bmod 29 = 25$

$24^4 \bmod 29 = 331776 \bmod 29 = 16$

$K_B = (16^2 \times 25) \bmod 29 = 6400 \bmod 29$

$\therefore K_B = 20$

QT: Consider a Diffie-Hellman scheme with a common prime $q = 11$ and a primitive root $\alpha = 2$.

i) Show that 2 is a primitive root of 11.

iii) If user A has a public key $Y_A = 9$, what is A's private key X_A?

→ Given: $q = 11$; $\alpha = 2$; $Y_A = 9$

i. 2 is a primitive root of 11

Let 'p' be the prime number and 'a' be the primitive root of 'p'. Then $a \bmod p$, $a^2 \bmod p$ ..... $a^{p-1} \bmod p$ consists of all integers from 1 to p-1, in some permutation.

For any integer

$b = a^i \pmod p$; $0 \le i \le p-1$, the exponent 'i' is called the discrete logarithm of b for the base a, mod p.

From Euler's theorem, for every a and n that are relatively prime: $a^{\phi(n)} = 1 \pmod n$ where $\phi(n)$: Euler's totient function is the number of positive integers less than n and relatively prime to n.

$\phi(n) = 11 - 1 = 10$

$2^{10} \bmod n = 2^{10} \bmod 11 = 1024 \bmod 11 = 1$

or $2^{10} = 1024 = 1 \bmod 11$

Therefor 2 is a primitive root of 11. For $n < 10$, none of the values of 2^n is 1 mod 11.

ii. A's private key X_A:

$Y_A = \alpha^{X_A} \bmod q$

$9 = 2^{X_A} \bmod 11$

Therefore,

$X_A = 6$

Computing $b = a^i \pmod p$

$0 \le i \le p-1$

$2 \bmod 11 = 2$ $\quad 2^2 \bmod 11 = 4$

$2^3 \bmod 11 = 8$ $\quad 2^4 \bmod 11 = 5$

$2^5 \bmod 11 = 10$ $\quad 2^6 \bmod 11 = 9$

6) Given: $q = 17$; $\alpha = 3$; $X_A = 15$; $X_B = 13$

- A's public key

$Y_A = \alpha^{X_A} \bmod q$

$Y_A = 3^{15} \bmod 17 = 14348907 \bmod 17$

$\therefore Y_A = 6$

B's public key

$Y_B = \alpha^{X_B} \bmod q$

$Y_B = 3^{13} \bmod 17 = 1594323 \bmod 17$

$\therefore Y_B = 12$

Shared secret key

$K_A = Y_B^{X_A} \bmod 17$

$K_A = 12^{15} \bmod 17 = [(12^8 \bmod 17)(12^4 \bmod 17)(12^2 \bmod 17)(12^1 \bmod 17)] \bmod 17$

$12^1 \bmod 17 = 12$

$12^2 \bmod 17 = 144 \bmod 17 = 8$

$12^4 \bmod 17 = 20736 \bmod 17 = 13$

$12^8 \bmod 17 = 429981696 \bmod 17 = 16$

$\therefore K_A = (16 \times 13 \times 8 \times 12) \bmod 17 = 10$

$K_B = Y_A^{X_B} \bmod 17$

$K_B = 6^{13} \bmod 17 = [(6^8 \bmod 17)(6^4 \bmod 17)(6^1 \bmod 17)] \bmod 17$

$6^8 \bmod 17 = 1679616 \bmod 17 = 16$

$6^4 \bmod 17 = 1296 \bmod 17 = 4$

$6^1 \bmod 17 = 6$

$\therefore K_B = (16 \times 4 \times 6) \bmod 17 = 384 \bmod 17 = 10$

- Suppose $q = 17$; $\alpha = 3$; $y_A = 6$, find x_A

$y_A = \alpha^{x_A} \bmod q$

$6 = 3^{x_A} \bmod 17$

$\therefore x_A = 15$

They should be numbers from 1 to 16 (17-1) in some permutation.

$3^1 \bmod 17 = 3$ $3^9 \bmod 17 = 14$
$3^2 \bmod 17 = 9$ $3^{10} \bmod 17 = 8$
$3^3 \bmod 17 = 10$ $3^{11} \bmod 17 = 7$
$3^4 \bmod 17 = 13$ $3^{12} \bmod 17 = 4$
$3^5 \bmod 17 = 5$ $3^{13} \bmod 17 = 12$
$3^6 \bmod 17 = 15$ $3^{14} \bmod 17 = 2$
$3^7 \bmod 17 = 11$ $3^{15} \bmod 17 = 6$
$3^8 \bmod 17 = 16$

- Elliptic Curve Cryptography (ECC)

It is a public-key cryptography introduced in 1985. It provides equal security with smaller keysize as compared to non ECC algorithm e.g., RSA. i.e., ECC provides high security with smaller key size. Examples of applicability are to encrypt internet traffic and for encryption on mobile devices.

- Elliptic Curves:

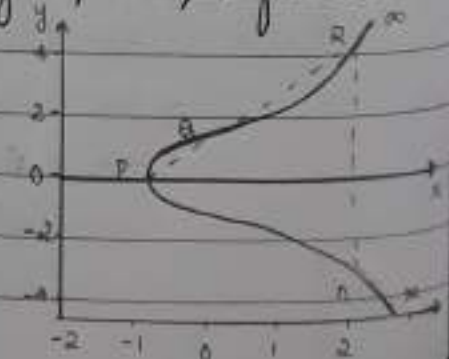
Elliptic curves are not ellipses. They are described by cubic equations, similar to those used for calculating circumference of ellipse. Here we limit to equations of the form:

$$y^2 = x^3 + ax + b$$

To plot such a curve, we need to compute

$$y = \sqrt{x^3 + ax + b}$$

It means, for given values of a and b, the plot consists of positive and negative values of y for each value of x. Thus each curve is symmetric about $y = 0$, i.e., symmetric about x-axis.



Ex: Let $E_p(a,b)$ be the elliptic curve. Consider the equation: $Q = kP$ where Q and P are points on the curve and $k < n$. It is relatively easy to calculate Q

given k and P, but it is relatively hard to determine k given Q and P. This is called the discrete logarithm problem for elliptic curves.

$E_p(a,b)$ consists of all pairs of integers (x,y), such that a and b are coefficients of x and y variables.

Ex: If p=23, for the elliptic curve

$$y^2 = x^3 + x + 1 \; ; \; a=1, \; b=1$$

$$E_p(a,b) = E_{23}(1,1)$$

- ECC Diffie-Hellman Key Exchange:
  - Global Public Elements:
    - $E_q(a,b)$: elliptic curve with parameters a, b and q, where q is a prime or an integer of the form 2^m
    - G : Base point: point on elliptic curve whose order is larger than n.
  - User A key Generation:
    - Select private n_A : $n_A < n$
    - Calculate public P_A : $P_A = n_A \times G$
  - User B key Generation:
    - Select private n_B : $n_B < n$
    - Calculate public P_B : $P_B = n_B \times G$
  - Calculation of secret key by User A:
    - $K = n_A \times P_B$
  - Calculation of secret key by User B:
    - $K = n_B \times P_A$

A key exchange between users A and B can be accomplished as follows:

1. A selects an integer n_A less than n. This is A's private key. A then generates a public key $P_A = n_A \times G$; the public key is a point in $E_q(a,b)$.
2. B similarly selects a private key n_B and computes a public key P_B.

3. A generates the secret key $k = n_A P_B$. B generates the secret key $k = n_B \times P_A$.

The two calculations in step 3 produce the same result because

$$n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A$$

To break this scheme, an attacker would need to be able to compute k given G and kG, which is assumed to be hard.

Secret key is a pair of numbers. If this key is to be used as a session key for conventional encryption, then a single number must be generated. We could use the x coordinates or some simple function of the x-coordinate.

- Elliptic Curve Encryption / Decryption:
  - Let plaintext message be M. First encode this message M into a point on elliptic curve.
  - Let the point be Pm. The point Pm will be encrypted as a ciphertext and subsequently decrypted.

We cannot simply encode the message as the x or y coordinate of a point, because not all such coordinates are in $E_q(a,b)$.

As with the key exchange system an encryption / decryption system requires a point G and an Elliptic group $E_q(a,b)$ as parameters. Each user selects a private key and generates a public key: $P_A = n_A \times G$ (n_A: private key).

For Encryption:

- User A chooses a random positive integer k.
- Ciphertext Cm consists of the pair of points:

  $$Cm = \{kG, P_m + kP_B\}$$

- A user B's public key P_B.
- This point Cm will be sent to the receiver B.

For Decryption:

- $Cm = \{kG, P_m + kP_B\}$

User B multiplies the first point in the pair by its

secret key and subtracts the result from the second point.

$P_m + kP_B - n_B(kG)$

$= P_m + k(n_B \times G) - n_B(kG) = P_m$ (because $P_B = n_B \times G$)

- A had marked the message P_m by adding kP_B to it. Only A knows the value of k.
- Because only B knows the corresponding private key n_B, only B can recover the message.
- Security of ECC:

It depends on how difficult it is to determine k given kP and P. This is refered to as the elliptic curve logarithm problem. Considerably smaller key size can be used for ECC compared to RSA. There is a computational advantage of using ECC with a shorter key length than a comparable secure RSA.

• Applications of Cryptographic Hash Functions:

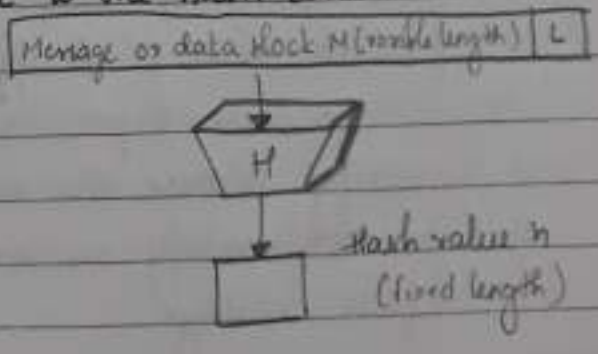
- Hash Function:

It is a variation of message authentication code. As with the message authentication code, a hash function accepts a variable size message M as input and produces a fixed-size output referred to as a hash code, H(M).

Unlike a MAC, a hash code does not use a key but is a function only of the input message.

Hash code is also referred to as a message digest or hash value. It is a function of all the bits of the message and provides an error-detection capability. A change to any bit or bits in the message results in a change to the hash code.

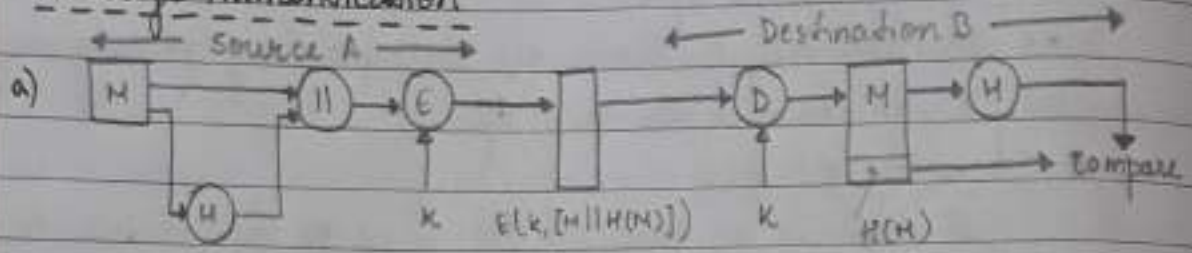
Typically, input is padded out to an integer multiple of some fixed length and padding also includes the value of the length of the original message in bits.



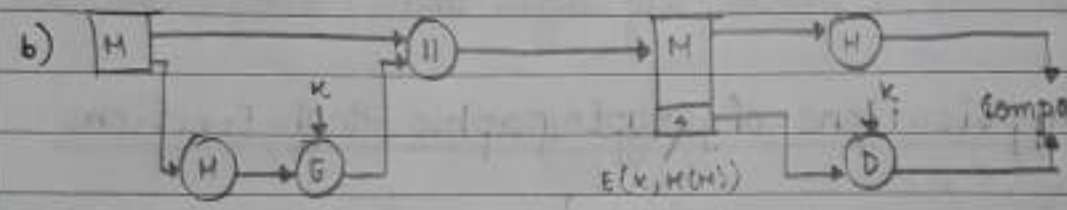
Length field is a security measure to increase the difficulty for an attacker to produce an alternative message with the same hash value.

- Applications:

1. Message Authentication



The message plus concatenated hash code is encrypted using symmetric encryption. The hash code provides the structure and redundancy required to achieve authentication.

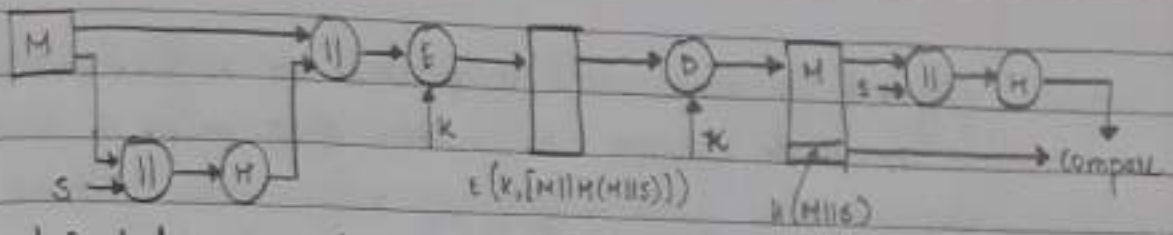


Only the hash code is encrypted, using symmetric encryption. This reduces the processing burden for those applications that do not require confidentiality.

Hash function is used but no encryption for message authentication. It assumes that both A and B share a common secret value S. Hash value is computed over the concatenation of message M and secret value S. A appends the resulting hash value to M.

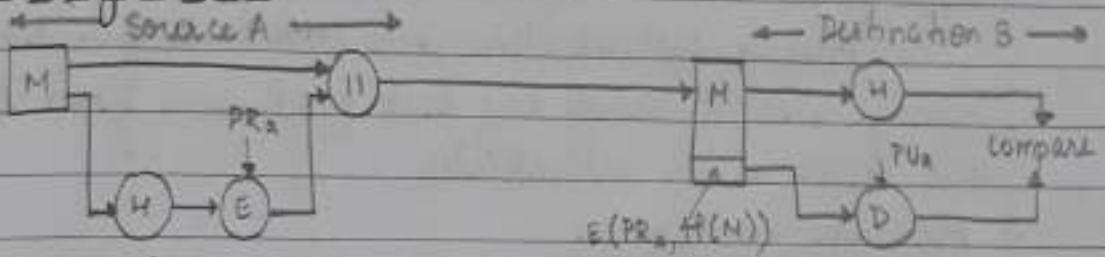
Because B possesses S, it can recompute the hash value to verify. Because the secret value itself is not sent, an opponent cannot modify an intercepted message and cannot generate a false message.

d)

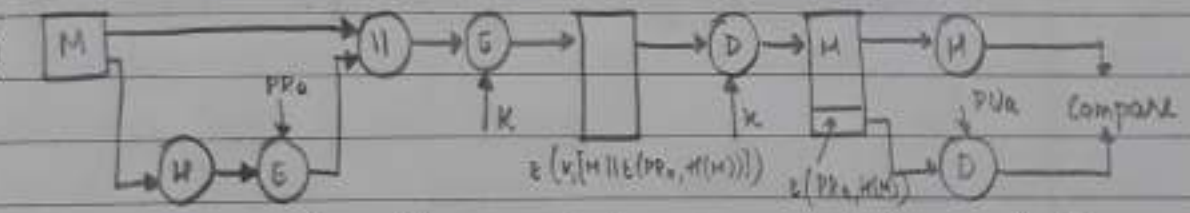


confidentiality can be added to previous approach by encrypting the entire message plus the hash code

2. Digital Signatures



The hash code is encrypted, using public-key encryption with the sender's private key. It provides authentication and also a digital signature, because only the sender could have produced the encrypted hash code.



The message plus the private-key encrypted has code is encrypted using a symmetric secret key

- **Message Authentication Functions:**

Message authentication is a procedure to verify that received messages come from the alleged source and have not been altered. It may also verify sequencing and timeliness

A digital signature is an authentication technique that also includes measures to counter repudiation by the source.

Any message authentication or digital signature mechanism has two levels of functionality. At the lower level, there must be a function that produces an authenticator: a value to be used to authenticity of a message.

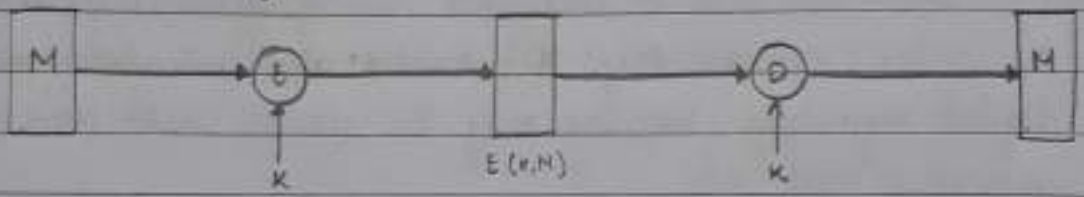
Types of functions used to produce an authenticator:

1. Hash function: A function that maps a message of any length into a fixed-length has value which serves as the authenticator.

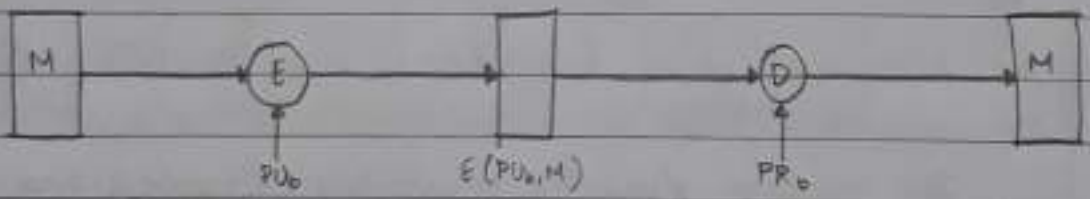
2. Message Encryption: The ciphertext of the entire message serves as an authenticator.

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

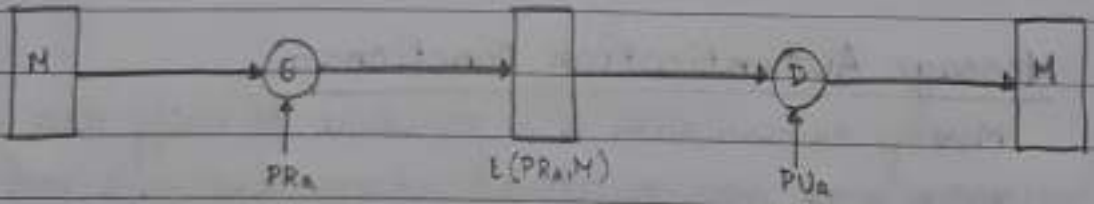
- Message Encryption:



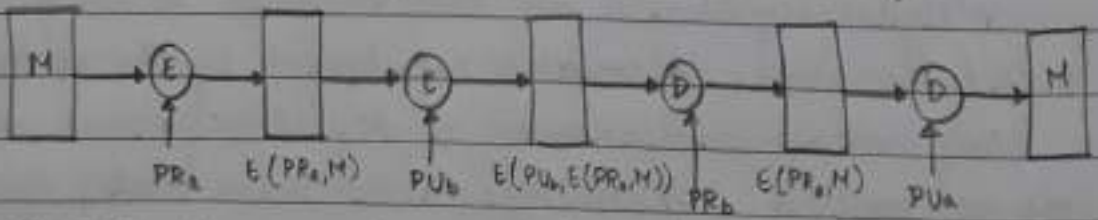
a) Symmetric Encryption: confidentiality and Authentication



b) Public-key Encryption: Confidentiality



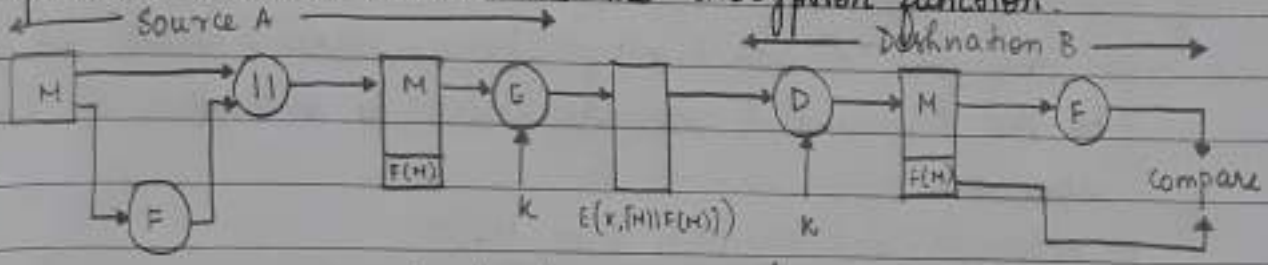
c) Public-key Encryption: authentication and Signature



d) Public-key encryption: confidentiality, authentication and Signature

It may be difficult to determine automatically if incoming ciphertext decrypts to intelligible plaintext. Thus, an opponent could achieve a certain level of disruption simply by issuing messages with random content purporting to come from a legitimate user.

One solution to this problem is to force the plaintext to have some structure that is easily recognized but that cannot be replicated without recourse to the encryption function.



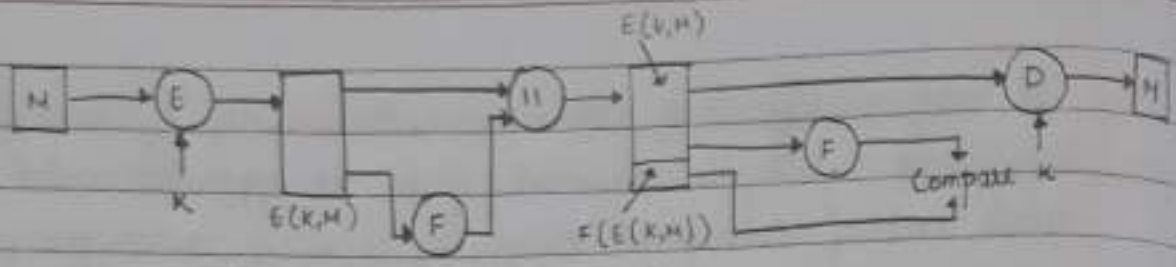
a) Internal error control

Append an error detecting code, also known as a frame check sequence (FCS) or checksum, to each message before encryption. A prepares a plaintext message M and then provides this as input to a function F that produces an FCS. The FCS is appended to M and the entire block is then encrypted.

At the destination, B decrypts the incoming block and treats the results as a message with an appended FCS. B applies the same function F to try to reproduce the FCS. If the calculated FCS is equal to the incoming FCS, then the message is considered authentic.

Based on the order in which the FCS and encryption functions are performed, the sequence (FCS) is referred to as internal error control, i.e., FCS followed by encryption. With internal error control, authentication is provided because an opponent would have difficulty generating ciphertext that when decrypted would have valid error control bits.

In external error control, encryption is followed by FCS. If the FCS is the outer code, an opponent can construct messages with valid error control codes.



b) External Error Control

Although the opponent cannot know what the decrypted plaintext will be, he can still hope to create confusion and disrupt operations.

Any sort of structuring like an error-control code added to the transmitted message serves to strengthen the authentication capability.

- Message Authentication Code (MAC):

Authentication technique which involves the use of a secret key to generate a small fixed-size block of data, known as a cryptographic checksum or MAC, that is appended to the message. This technique assumes that two communicating parties, say A and B, share a common secret key K.

When A has a message to send to B, it calculates the MAC as a function of the message and the key:

MAC = MAC (K,M)

where M = input message
C = MAC function
K = shared secret key
MAC = Message authentication code.

The message plus MAC are transmitted to the intended recipient. The recipient performs the same calculation on the received message, using the same secret key, to generate a new MAC. Received MAC is compared to the calculated MAC. If the received MAC matches the calculated MAC, then the receiver is assured that the message has not been altered.

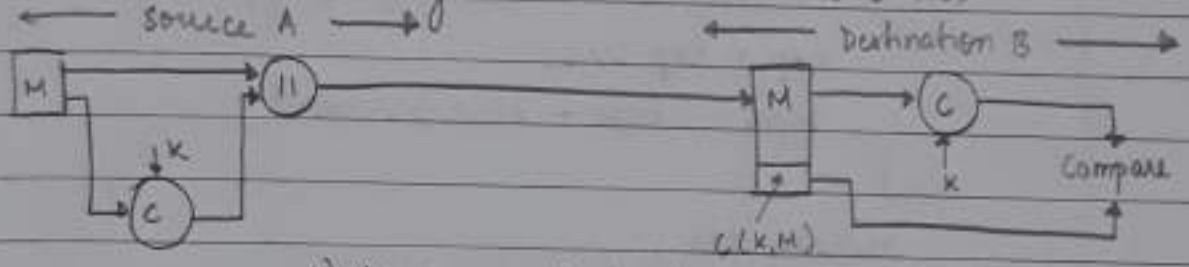
If we assume that only the receiver and the sender know the secret key and if the received MAC matches the

calculated MAC, then

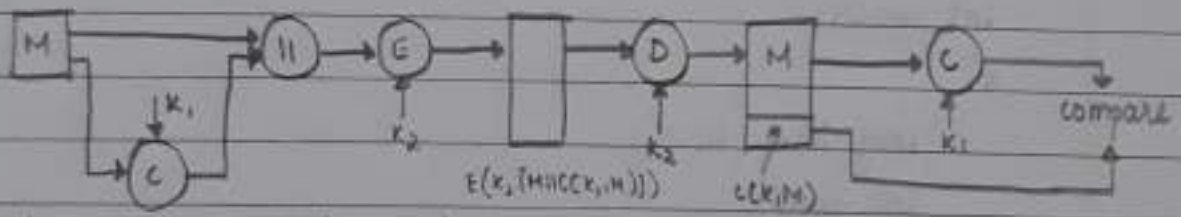
1. Receiver is assured that the message has not been altered
2. Receiver is assured that the message is from the alleged sender
3. If the message includes a sequence number, then the receiver can be assured of the proper sequence.

A MAC function is similar to encryption. One difference is that the MAC algorithm need not be reversible as it must be for decryption. MAC does not provide a digital signature because both sender and receiver share the same key.

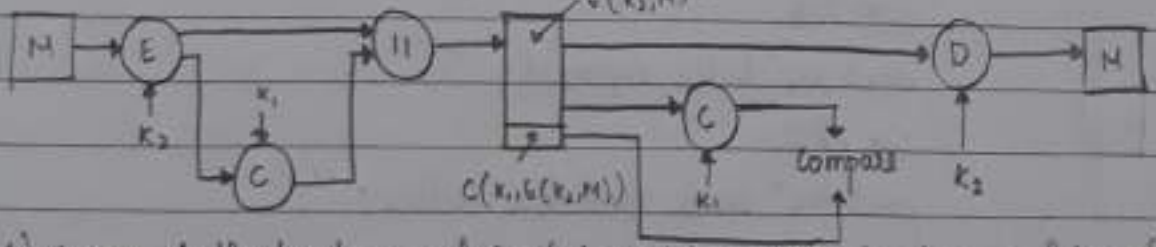
Basic uses of Message Authentication Code (MAC)



a) Message authentication



b) Message Authentication and Confidentiality: Authentication tied to plaintext



c) Message Authentication and Confidentiality: Authentication tied to ciphertext.

- **SLE: Program on a^b modulus n**

```
#include <iostream>
using namespace std;
long long modular (long long base, long long exp, int mod)
{
    long long res = 1
    while (exp>0)
    {
        if (exp % 2 == 1)
            res = (res * base) % mod;
        exp = exp >>1;
        base = (base * base) % mod;
    }
    return res;
}
int main()
{
    long long b, e;
    int mod;
    cout << " Enter base : ";
    cin >>b;
    cout << " Enter exponent : ";
    cin >>e;
    cout << " Enter modular value : ";
    cin >> mod;
    cout << modular (b,e,mod);
    return 0;
}
```