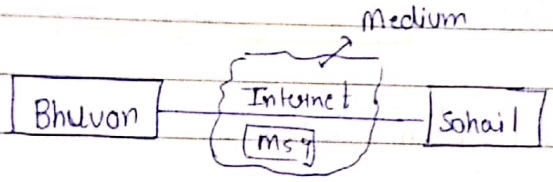
Cryptography

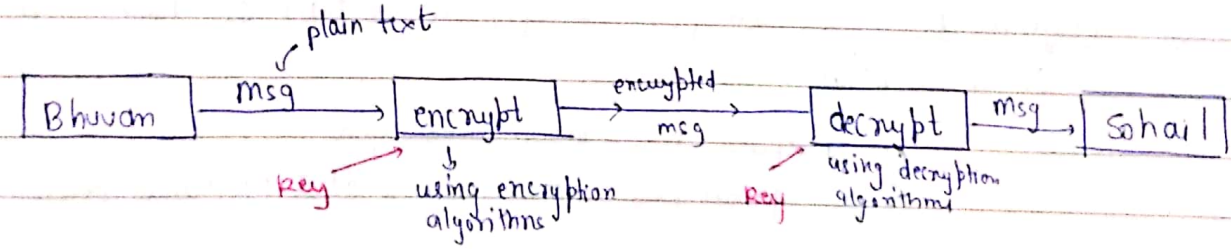
* Cryptography: The art of protecting information by transferring into an unreadable format.
Cryptography is about constructing and analyzing protocols that prevent third properties or the public from reading private messages.

* Why cryptography?



- Is it secure?
No bcz attacker/3rd party can corrupt/change our data and may misuse it also.

Thus, to provide security and protect the valueable info, we can use cryptography.



Case i) if keys are same → Symmetric cryptography
ii) if key are different → Asymmetric cryptography

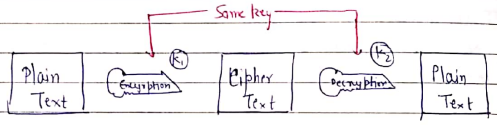
- Encryption → change data from readable to unreadable
- Decryption → Transform data from unreadable to readable.

- Key → String of bits used by cryptographic algorithms to transform plain text to cipher text or vice-versa

* Types of cryptography:

① Symmetric: It is the simplest kind of encryption technique that involves only 1 key to encrypt and decrypt (or cipher & decipher information)

- It is also called Secret key Cryptography / Private key cryptography.
- eg. DES (Data Encryption System)



② Asymmetric: It uses two keys ie a pair of keys for encryption & decryption

- (A) → public key - known to everyone
- (B) private key - known only to that particular person

→ Also called public key cryptography

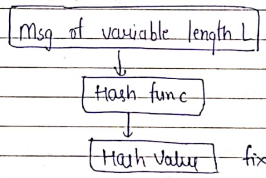
Note: A msg that is encrypted using a public key can only be decrypted using a private key, while also,
A message encrypted using private key can be decrypted using public key.

eg- RSA, DSA, Elliptic curve etc.

③ Hash functions:

- There is no concept of key in this algo.
- → Takes in variable length size message and gives fixed size output
  - ↳ Hash code or Hash value
- Hash code makes it impossible for the contents of plain text to be recovered.

Many OS use hash functions to encrypt passwords



*	Symmetric Crypto	Asymmetric Crypto
1.	Also called private key crypto or secret key crypto	1. Also called public key crypto.
2.	Only 1 key is used for encryption & decryption	2. Diff keys (public & private) is used for encryption & decryption
3.	Performance → These are faster in execution	3. Performance → Slower in execution
4.	Less complex & less computational power required	4. More complex & more computational power required.

5. Used for transfer of bulk data (bc executes faster)	5. Used for securely exchanging the secret key
6. Problem - Sharing the key in between sender & receiver is not safe	6. Problem - No problem of key sharing bcz of private key concept
7. Commonly used symmetric key algos - DES, AES, RC4 2 DES, 3 DES	7. Algos - RSA, DSA, Deffie Hellman

* Security Goals:

① CONFIDENTIALITY - It is the most common aspect of information security

- It allows authorized users to access sensitive & protected data.
- The data sent over the network should not be accessed by unauthorized users.

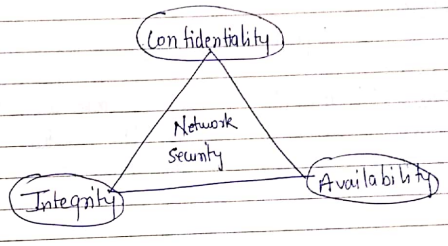
Attackers will try to capture data. To avoid this, various encryption techniques are used to safeguard our data so that even if attacker gains access, he/she will be able to decrypt it

② INTEGRITY - It means that changes need to be done only by the authorized entities and through authorized mechanisms & nobody else should be able to modify our data.
eg - In a bank, when we deposit / withdraw money the balance needs to be maintained.

③ AVAILABILITY :- Data must be available to the authorized user.
Info is useless if we can not access it.

[Eg] - what would happen if we can't access our bank, accounts for transactions

CIA Triad in cryptography
& confidentiality
Integrity
Availability



* Security Services:

① Data Confidentiality - Protect data from attackers.

② Data Integrity - Protecting data from modification

③ Authentication - Verifying actual person hai ya nahi

④ Non-Repudiation - Assurance that someone can't deny the validity of something.

- NR is a service which provides proof of the origin of data and the integrity of data.

eg → A gives ₹1000 check to B & later B deny it!!
It can't happen bcz A will have its proof.

⑤ Access Control → To whom the access can be given can be decided.
or
The prevention of unauthorized use of a resource (i.e. this service controls who can have access to our info, under what conditions)

* Security Attacks → Passive Attacks
→ Active Attacks

① Passive Attack - It attempts to learn or make use of the info from the system but do not affect the system resources i.e. the attacker will only see the data he will not modify it

- We can prevent it by using better encryption
- Two types of passive attacks :-

1. Release of message contents - The attacker / hacker will easily be able to understand the data/info.

2. Traffic Analysis - If we have encryption protection, hacker might still be able to observe the pattern of these messages

The attacker would determine the location and the identity of communication hosts and could observe the frequency & length of messages being exchanged

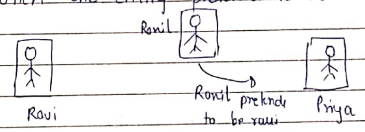
This info might be helpful in guessing the nature of communication that was taking place

- Passive attacks are difficult to detect bcz they don't involve any alteration of data.
So, the sender & receiver will not be able to know whether a third person is reading their message or not.

② Active Attacks - It attempts to alter system resources / info
↳ see + modify msg

- Masquerade
↳ when one entity pretends to be another entity

eg

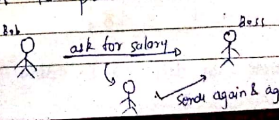


- Modification of messages
↳ some portion of the message is altered or the message is d
or reordered to produce an unauthorized effect

eg → Give 100 ₹ to John
↓
Give ₹500 to Gaurav

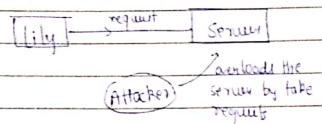
- Replay
↳ Involves passive capture of a message & its subsequen
retransmission to produce an unauthorized effect

eg)



- Denial of Services
  - It prevents normal use of communication facilities

eg) disruption of an entire network either by disabling the network or by overloading it by messages so as to degrade performance



* Security Mechanisms - These are used to provide security.

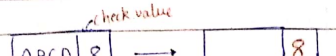
① Encipherment - The use of mathematical algo's to transform data into a form that is not readable

Plain text → Cipher text

② Digital Signature - Means by which a sender can electronically sign the data & the receiver can electronically verify the signature

③ Data Integrity - This mechanisms appends to the data a short check value that has been created by a specific process from the data itself. The receiver creates a new check value from the received data & compares the newly created check-value with the one received

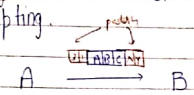
If both values are same, the integrity of data has been preserved



check value made from data like by adding the value of digits or to alphabet

④ Authentication Exchange - Two entities exchange some messages to prove their identity to each other.

⑤ Traffic padding - We add some extra/dummy bits with data while encrypting.



⑥ Routing Control - Means selecting & continously changing different available routes b/w the sender & the receiver to prevent the attacker from eavesdropping on a particular route

↓

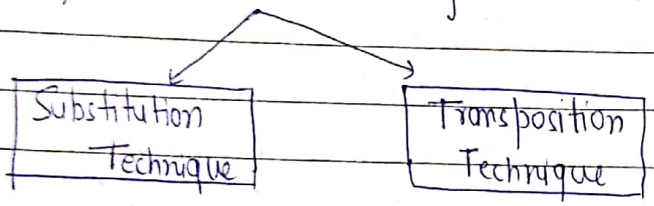
जासूसी

⑦ Access Control - These methods prove that a user has access right to the data

⑧ Notarization - Means selecting a third trusted party to control the communicating b/w two entities This can be done to prevent repudiation

ENCRYPTION TECHNIQUES

* Classical Encryption Techniques:
- Symmetric encryption also referred to as conventional encryption is of 2 types. or we can say it has 2 techniques

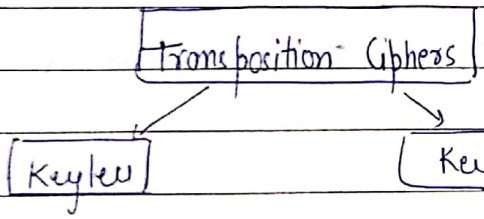


① Substitution Techniques - It is the one in which the letters of the plain text are replace by other letters or by numbers or symbols

eg→ Name → I w P x

② Transposition Techniques - Performing some sort of permutations on the plain text letters i.e. reordering the symbols.

eg N A M E → E A M N or
A E M N or M N E A etc.



Transposition Ciphers
- Rail fence
- Columnar transposition
- Double transposition

* Substitution Techniques: → poly alphabetic cipher
→ mono alphabetic cipher

① Mono Alphabetic Substitution Cipher - A single cipher alphabet for each plain text alphabet is used throughout the proces i.e. fixed substitution hoga

if 'N' → I use 'x', then I will always use 'x' in place of 'N'

eg MY NAME → NP OBNZ

② Polyalphabetic substitution -
- There is no fixed substitution
- Each occurance of a character may have a different substitute ie we can use more than 1 substitution for the same letter

eg MY NAME → NP OB XZ

- The relationship b/w a character in the plain text to a character in cipher text is one to many

eg
a → d & later a → m

Transposition Techniques:

① Rail Fence Technique -
- In this, plain text is written down as a sequence of diagonals & then read off as a sequence of rows

eg.- "all the best for exams" - plain text
encrypting with rail fence technique (depth = 2)

a	l	l	h	b	e	s	t	f	o	r	e	x	a	m	s
		l	t	e	e	t	o		e		a				s

encrypted msg is → ALLHBESTFORXAMSLTEETOEAS

- Used for short messages
- Easy to break by the attacker

② Row transposition Cipher (Columnar Transposition) -
- We write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of column

(Key) - integer value (unique digits from 0 to 9)
eg- 453 12, 4321 etc

CRYPTO → 14 6 3 5 2
1 4 6 3 5 2

- Example- text → "attack postponed until two am"

Key →	4	3	1	2	5	6	7
	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

extra dummy bits

Cipher → TINA APTM TSUO AODW COIX KNLY PETZ

- Explaination - i) To encrypt start with the column labelled as in our case column 3 write down all the letters of column
ii) Now proceed to column no 4 which is labelled as 2 t
2, 1, 5, 6 & 7th column.

- Can easily be understood by the attacker / 3rd party
- Used for short msgs only.
-

③ Double Transposition -

- Columnar / Row transposition cipher applied twice
- The key in case 2 can be same or different also
- This technique was used in world war I by german military & also by world war II.

Cipher text from prev. example as plain text & applying row transpositn again

Plain - "TTNA APTM TSUO AODW COIX KNLY PETZ"

Key →	4	3	1	2	5	6	7
	t	t	n	a	a	p	t
	m	t	s	u	o	a	o
	d	w	c	o	i	x	k
	n	l	y	p	e	t	z

Cipher - NSCY AUOP TTWL TMDN AOIE PAXT TOKZ

NOTE: key → STRIPE
564231

eg key → ZEBRA
53241

NOTE: Transposition ciphers are of two types ① Keyed (which uses key) eg Row transposition ② Keyless (which don't use key). eg Rail fence

* Caesar cipher:

- It is also called ~~swift~~ shift cipher / additive cipher
- Each letter in the plaintext is replaced by a letter corr. to a no. of shifts in the alphabet

- It is monoalphabetic caesar cipher
- Oldest & simplest method of encryption technique

NOTE: Julius caesar used an additive cipher to communicate with his officers. for this reason, additive ciphers are sometimes know as caesar cipher.

He used a key of 3 for communication.

eg- Plain → Meet me
Cipher → PHHW PH

Cipher Text

$$C = E(K, P) = (P + K) \bmod 26$$ // encryption

(K: key, P: Plain, message)

Plain Text

$$P = D(K, P) = (C - K) \bmod 26$$ // decryption

(C: Cipher Text)

if (C-K) is -ve, then add 26 to it

- Numerical value is assigned to each letter

a	b	c	d	e	f	.....	x	y	z
0	1	2	3	4	5		23	24	25

- If the cryptanalyst / attacker knows a cipher text, then he can apply a brute-force technique to find the plain text by using all the possible 25 keys.
- Since it is a part of symmetric encryption, same key is used for encryption & decryption.

$$1 \leq K \leq 25$$

eg- Plain text → "HELLO"

Encryption

let key = 4

$C(H) = (P+K) \bmod 26$
$= (7+4) \% 26 = 11$ (L)

$C(E) = (P+K) \bmod 26 = (4+4) \% 26 = 8$ (I)

$C(L) = (11+4) \% 26 = 15$ (P)
$C(O) = (14+4) \% 26 = 18$ (S)

∴ Cipher → LIPPS

Decryption

Cipher → "LIPPS" key = 4

$P(L) = (C-K) \bmod 26$
$= (11-4) \% 26 = 7$ (H)
$P(I) = (8-4) \% 26 = 4$ (E)
$P(P) = (15-4) \% 26 = 11$ (L)
$P(S) = (18-4) \% 26 = 14$ (O)

∴ Plain → HELLO

* Playfair Cipher Algorithm:

- Invented in 1854 by charles wheatstone. but was named after Lord Playfair, who promoted the use of cipher.
- Algorithm

① Create a 5×5 matrix of that is called grid of letters.
② The matrix is made by inserting the values of key and remaining alphabets into the matrix (row wise from left to right), where letter I & J will be combined together.

③ Convert the texts into pair of alphabets
eg Hey or → He yo rx
→ if pair can't be made with some letters. Break the letter in single and add 'x' to the previous letter.

eg Hello → He lx lo
Helloe → He lx lo e
above two problems

→ if the letter is standing alone in the process of pairing, then add 'z' with the letter.

eg → Helloe → He lx lo ez
e was alone so we add z here

Hexxoe → He xz xo ez
x was already here, so we took z

④ Code will be formed using 3 rules:
i) if both the alphabets are in the same row, replace them with alphabets to their immediate right

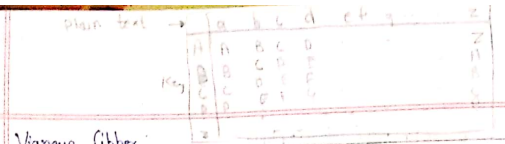
ii) if both the alphabets are in the same column, replace them with alphabets immediately below them.

iii) if not in same row/column, replace them with alphabets in the same row respectively, but at other pair of corners.

key → Abhi

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

Cipher
Plain text
same column: BM → ER, RW → WB
same row: FG → GK, UQ → QR
Horizontal: QW → RV, KS → FU, KL → DP



† Vigenere Cipher:

- Designed by Blaise de Vigenere (16th century french mathematician).
- It is a polyalphabetic substitution cipher.
- The encryption is done using a (26 x 26) matrix or table.

- Method ① → **Vigenere Table**

eg:- Plain Text - GIVE MONEY
Key - LOCK

Sol:- (P) G I V E M O N E Y
(K) L O C K L O C K L

repeat the letters of the key so that the no. of letters in P & K ie. plain txt and key becomes equal

cipher → R W X O X C P O J

For decryption,

(C) R W X O X C P O J
(K) L O C K L O C K L

Plain→ G I V E M O N E Y

- Method ② → **Without table**

→ Encryption

$E_i = (P_i + K_i) \bmod 26$

→ Decryption

$D_i = (E_i - K_i) \bmod 26$

Plain text → "She is listening"
key → "PASCAL"
∴ key stream (15, 0, 18, 2, 0, 11) The key stream is the repetition of this initial key stream

Plain	S	h	e	i	s	l	i	s	t	e	n	i	n	g
P's value	18	7	4	8	18	11	8	18	19	4	13	8	13	6
Key stream	15	0	18	2	0	11	15	0	18	2	0	11	15	0
c's value	7	7	22	10	18	22	23	18	11	6	13	19	2	6
Cipher	H	H	W	K	S	W	X	S	L	G	N	T	C	G

Plain value	18	7	4	8	18	11	8	18	19	4	13	8	13	6
Plain Text	S	H	E	I	S	L	I	S	T	E	N	I	N	G

† Vernam Cipher:

- Used for encrypting alphabetic text, it is a type of substitution cipher.
- In this, we assign a number to each character of plain-text like (a=0, b=1, c=2, ... , z=25)
- length of key used for encryption = length of plain text

Plain Text - "RAMSWARUPK" Key → "RANCHOBABA"

Plain Text	17	0	12	18	22	0	17	20	15	10
Key	17	0	13	2	7	14	1	0	1	0
PT+key	34	0	25	20	29	14	18	20	16	10
Subt.	18	0	25	20	3	14	18	20	16	10
Cipher	I	A	Z	U	D	O	S	U	Q	K
Cipher Text	8	0	25	20	3	14	18	20	16	10
Key	17	0	13	2	7	14	1	0	1	0
CT-Key	-9	0	12	18	-4	0	17	20	15	10
Plain	17	0	12	18	22	0	17	20	15	10
	R	A	M	S	W	A	R	U	P	K

* Hill Cipher

1. Developed by lester Hill in 1929
2. Encrypts a group of letters called polygraph (like in playfair cipher, we saw it was encrypting a pair of letters which was called as a digraph). So, here it can be a polygraphs (digraph, trigraph, etc.)
3. This method make use of mathematics

To encrypt,

$$C = KP \bmod 26$$

Step 1 → Choose a key (key matrix must be a square matrix)

We can take any key

eg. VIEW → $\begin{bmatrix} V & I \\ E & W \end{bmatrix} = \begin{bmatrix} 21 & 8 \\ 4 & 22 \end{bmatrix}_{2\times2}$

key = QUICKNESS → $\begin{bmatrix} Q & U & I \\ C & K & N \\ E & S & S \end{bmatrix} = \begin{bmatrix} 16 & 20 & 8 \\ 2 & 10 & 13 \\ 4 & 18 & 18 \end{bmatrix}_{3\times3}$

eg. plain text = ATTACK

let key = $\begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix}$

Since, the key is a 2×2 matrix, plain text should be converted into vector of length 2

So, $\begin{bmatrix} A \\ T \end{bmatrix} \begin{bmatrix} T \\ A \end{bmatrix} \begin{bmatrix} C \\ K \end{bmatrix}$

Now encryption begins,

① So, 1st vector → $\begin{bmatrix} A \\ T \end{bmatrix} = \begin{bmatrix} 0 \\ 19 \end{bmatrix}$, key = $\begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix}$

$$C = KP \bmod 26$$

$$C = \begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 0 \\ 19 \end{bmatrix} \bmod 26 = \begin{bmatrix} 2\times0 + 3(19) \\ 3(0) + 6(19) \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 57 \\ 114 \end{bmatrix} \bmod 26 = \begin{bmatrix} 5 \\ 10 \end{bmatrix}$$

corresponding elements (alphabet) = $\begin{bmatrix} F \\ K \end{bmatrix}$

∴ Plain text $\begin{bmatrix} A \\ T \end{bmatrix}$ becomes $\begin{bmatrix} F \\ K \end{bmatrix}$ i.e. AT → FK

② Now, 2nd vector is $\begin{bmatrix} T \\ A \end{bmatrix} = \begin{bmatrix} 19 \\ 0 \end{bmatrix}$

$$C = KP \bmod 26$$

$$= \begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 19 \\ 0 \end{bmatrix} \bmod 26 = \begin{bmatrix} 2(19) + 3(0) \\ 3(19) + 6(0) \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 38 \\ 57 \end{bmatrix} \bmod 26 = \begin{bmatrix} 12 \\ 5 \end{bmatrix}$$

Now, corresponding alphabets will be $\begin{bmatrix} M \\ F \end{bmatrix}$

So, TA → MF

③ Next is $\begin{bmatrix} C \\ K \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 10 \end{bmatrix}$

$$C = KP \bmod 26 = \begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 2 \\ 10 \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 34 \\ 66 \end{bmatrix} \bmod 26 = \begin{bmatrix} 8 \\ 14 \end{bmatrix} = \begin{bmatrix} I \\ O \end{bmatrix}$$

So, CK → IO

PLAIN → "AT TA CK"

Cipher → "FK MF IO"

* Hill Cipher Decryption

To encrypt, $C = KP \bmod 26$

To decrypt, Find inverse of key matrix K^{-1}

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

plain → ATTACK , key $K = \begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix}$

cipher → FKMFIO

$K^{-1} = \frac{1}{|K|} \text{ adj } (K)$

$|K| = \begin{vmatrix} 2 & 3 \\ 3 & 6 \end{vmatrix} = |12 - 9| = 3$ ∴ determinant value, d=3

Now find multiplicative inverse of determinant

i.e $d\,d^{-1} = 1 \bmod 26$ (identity matrix)

$3 * d^{-1} = 1 \bmod 26$

so, $d^{-1} = 9$

- Use Hit & Trial method

$1 \bmod 26 = 1$

$3 * d^{-1} \bmod 26 = 1$

$3 * 9 \bmod 26 = 1$

$27 \bmod 26 = 1$

another example.

$5 * d^{-1} \equiv 1 \bmod 26$

↳ multiplicative inverse of 5

$5 * n \bmod 26 = 1$

let $n = 5$

$5 * 5 \bmod 26 = 25 \bmod 26 = 25$

let $n = 11$

$5 * 11 \bmod 26 = 55 \bmod 26 = 3$

let $n = 21$

$5 * 21 \bmod 26 = 105 \bmod 26 = 1$

Multiplicative inverse of 5 = 21

So till now, determinant $d = 3$

$d^{-1} = 9$

Now we will find adjoint of the matrix

Let $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ then $\text{adj}(A) = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

$K = \begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix}$, $\text{adj}(K) = \begin{bmatrix} 6 & -3 \\ -3 & 2 \end{bmatrix}$

Before decryption, we have to remove -ve values (add 26 to -ve)

$\therefore \text{adj}(K) = \begin{bmatrix} 6 & -3+26 \\ -3+26 & 2 \end{bmatrix} = \begin{bmatrix} 6 & 23 \\ 23 & 2 \end{bmatrix}$

$\text{adj}(K) = \begin{bmatrix} 6 & 23 \\ 23 & 2 \end{bmatrix}$ and $d^{-1} = 9$

$K^{-1} = \frac{1}{|K|} \text{adj}(K) = |K^{-1}| \text{adj}(K) = d^{-1} \text{adj}(K)$

$K^{-1} = \begin{bmatrix} 54 & 207 \\ 207 & 18 \end{bmatrix} \bmod 26 = \begin{bmatrix} 2 & 25 \\ 25 & 18 \end{bmatrix}$

$K^{-1} = \begin{bmatrix} 2 & 25 \\ 25 & 18 \end{bmatrix}$

Now, we will decrypt

Cipher = FK MF IO

① $C = \begin{bmatrix} F \\ K \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \end{bmatrix}$

Plain text, $P = K^{-1} C \bmod 26 = \begin{bmatrix} 2 & 25 \\ 25 & 18 \end{bmatrix} \begin{bmatrix} 5 \\ 10 \end{bmatrix} \bmod 26$

$P = \begin{bmatrix} 260 \\ 305 \end{bmatrix} \bmod 26 = \begin{bmatrix} 0 \\ 19 \end{bmatrix} = \begin{bmatrix} A \\ T \end{bmatrix}$

③ Similarly, $C = \begin{bmatrix} M \\ F \end{bmatrix} = \begin{bmatrix} 12 \\ 5 \end{bmatrix}$

$P = \begin{bmatrix} 2 & 25 \\ 25 & 18 \end{bmatrix} \begin{bmatrix} 12 \\ 5 \end{bmatrix} \bmod 26 = K^{-1} C \bmod 26$

$= \begin{bmatrix} 149 \\ 390 \end{bmatrix} \bmod 26 = \begin{bmatrix} 17 \\ 0 \end{bmatrix} = \begin{bmatrix} T \\ A \end{bmatrix}$

④ $C = \begin{bmatrix} T \\ O \end{bmatrix} = \begin{bmatrix} 8 \\ 14 \end{bmatrix}$

$P = K^{-1} C \bmod 26 = \begin{bmatrix} 2 & 25 \\ 25 & 18 \end{bmatrix} \begin{bmatrix} 8 \\ 14 \end{bmatrix} \bmod 26$

$= \begin{bmatrix} 366 \\ 452 \end{bmatrix} \bmod 26 = \begin{bmatrix} 2 \\ 10 \end{bmatrix} = \begin{bmatrix} C \\ K \end{bmatrix}$

Plain text → ATTACK

* Hill Cipher (3x3) matrix

let plain text → "SAFE MESSAGES"

let key → "CIPHERING"

$\begin{bmatrix} C & I & P \\ H & E & R \\ I & N & G \end{bmatrix} = \begin{bmatrix} 2 & 8 & 15 \\ 7 & 4 & 17 \\ 8 & 13 & 6 \end{bmatrix}$

Since, key is a 3x3 matrix, plain text should be converted into column vectors of length 3

i.e (n x 1) matrices

↓

(3 x 1) matrices

So, we get SAF EMF SSA GES

$\begin{bmatrix} S \\ A \\ F \end{bmatrix} \begin{bmatrix} E \\ M \\ E \end{bmatrix} \begin{bmatrix} S \\ S \\ A \end{bmatrix} \begin{bmatrix} G \\ E \\ S \end{bmatrix}$

• Encryption $C = KP \bmod 26$

① $P = \begin{bmatrix} S \\ A \\ F \end{bmatrix} \rightarrow \begin{bmatrix} 18 \\ 0 \\ 5 \end{bmatrix}$

$C = \begin{bmatrix} 2 & 8 & 15 \\ 7 & 4 & 17 \\ 8 & 3 & 6 \end{bmatrix} \begin{bmatrix} 18 \\ 0 \\ 5 \end{bmatrix} \bmod 26$

$C = \begin{bmatrix} 2(18) + 15(5) \\ 7(18) + 4(0) + 17(5) \\ 8(18) + 3(0) + 6(5) \end{bmatrix} = \begin{bmatrix} 111 \\ 211 \\ 274 \end{bmatrix} \bmod 26 = \begin{bmatrix} 7 \\ 3 \\ 18 \end{bmatrix}$

Now the corresponding alphabets are $\begin{bmatrix} H \\ D \\ S \end{bmatrix}$

② $P = \begin{bmatrix} G \\ M \\ E \end{bmatrix} \rightarrow \begin{bmatrix} 6 \\ 12 \\ 4 \end{bmatrix}$

$C = KP \bmod 26 = \begin{bmatrix} 2 & 8 & 15 \\ 7 & 4 & 17 \\ 8 & 3 & 6 \end{bmatrix} \begin{bmatrix} 6 \\ 12 \\ 4 \end{bmatrix} \bmod 26 = \begin{bmatrix} 164 \\ 194 \\ 212 \end{bmatrix} \bmod 26$

$= \begin{bmatrix} 8 \\ 19 \\ 4 \end{bmatrix} \rightarrow \begin{bmatrix} I \\ O \\ E \end{bmatrix}$

③ $P = \begin{bmatrix} S \\ S \\ A \end{bmatrix} \rightarrow \begin{bmatrix} 18 \\ 18 \\ 0 \end{bmatrix}$

$C = KP \bmod 26 = \begin{bmatrix} 2 & 8 & 15 \\ 7 & 4 & 17 \\ 8 & 3 & 6 \end{bmatrix} \begin{bmatrix} 18 \\ 18 \\ 0 \end{bmatrix} \bmod 26 = \begin{bmatrix} 180 \\ 198 \\ 378 \end{bmatrix} \bmod 26 = \begin{bmatrix} 24 \\ 16 \\ 14 \end{bmatrix}$

$C = \begin{bmatrix} 24 \\ 16 \\ 14 \end{bmatrix} \rightarrow \begin{bmatrix} Y \\ Q \\ O \end{bmatrix}$

(f) Again, $P = \begin{bmatrix} G \\ E \\ S \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \\ 18 \end{bmatrix}$

$C = KP \bmod 26 = \begin{bmatrix} 2 & 8 & 15 \\ 7 & 4 & 17 \\ 8 & 13 & 6 \end{bmatrix} \begin{bmatrix} 6 \\ 4 \\ 18 \end{bmatrix} \bmod 26 = \begin{bmatrix} 319 \\ 364 \\ 208 \end{bmatrix} \bmod 26$

$C = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} C \\ A \\ A \end{bmatrix}$

∴ SAFEMESSAGES = HDSIOEYQOCAA

† Hill-Cipher Decryption (3×3 matrix):

Encrypt → $C = KP \bmod 26$
Decrypt → $P = K^{-1}C \bmod 26$

Cipher = "HDSIOEYQOCAA"
Key K = CIPHERING $\begin{bmatrix} 2 & 8 & 15 \\ 7 & 4 & 17 \\ 8 & 13 & 6 \end{bmatrix}$

Now, we need K^{-1}

i) finding determinant value

$d = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a(ei - fh) - b(di - gf) + c(dh - ge)$
$= 2(24 - 13(17)) - 8(7(6) - 8(17))$
$+ 15(7(13) - 8(4))$

$|d| = 1243$

Now, we will find multiplicative inverse of the determinant

i.e. $d \cdot d^{-1} \equiv 1 \bmod 26$
$1243\,(d^{-1}) \equiv 1 \bmod 26$
so, $d^{-1} = 5$

∵ $d^{-1} = 5$
→ $1243 \times 5 \bmod 26$
→ $6215 \bmod 26$
→ 1

Now, we will find adjoint (K)

$K = \begin{bmatrix} 2 & 8 & 15 \\ 7 & 4 & 17 \\ 8 & 13 & 6 \end{bmatrix}$ →

for 1st element $4(6) - 13(17)$
$= -197$

for 2nd element $7(6) - 8(17) = -94$
for 3rd element $7(13) - 8(4) = 59$

And so on, a $\begin{bmatrix} -197 & 94 & 59 \\ 197 & -108 & 38 \\ 76 & 71 & -98 \end{bmatrix}$ ← cofactor matrix

Now, we do transpose

$adj(K) = \begin{bmatrix} -197 & 197 & 76 \\ 94 & -108 & 71 \\ 59 & 38 & -98 \end{bmatrix}$ ← adjoint matrix

Removing -ve signs

$adj(K) = \begin{bmatrix} -197 + 26(n) & 197 & 76 \\ 94 & -108 + 26(n) & 71 \\ 59 & 38 & -98 + 2(26) \end{bmatrix} = \begin{bmatrix} 11 & 197 & 76 \\ 94 & 22 & 71 \\ 59 & 38 & 4 \end{bmatrix}$

Now,

$K^{-1} = |d^{-1}|\, adj(K)$
$K^{-1} = 5 * \begin{bmatrix} 11 & 197 & 76 \\ 94 & 22 & 71 \\ 59 & 38 & 4 \end{bmatrix} = \begin{bmatrix} 55 & 985 & 380 \\ 470 & 110 & 355 \\ 295 & 190 & 20 \end{bmatrix} \bmod 26$

Now, modulo by 26

$K^{-1} = \begin{bmatrix} 3 & 7 & 16 \\ 2 & 6 & 17 \\ 9 & 8 & 20 \end{bmatrix}$

Decrypting using formula $P = K^{-1}C \bmod 26$

Cipher : "HDS IOE YQO CAA"

Now, for HDS,

$$P = K^{-1}C \bmod 26 = \begin{bmatrix} 3 & 7 & 16 \\ 2 & 6 & 17 \\ 9 & 8 & 20 \end{bmatrix} \begin{bmatrix} 7 \\ 3 \\ 18 \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 336 \\ 338 \\ 441 \end{bmatrix} \bmod 26 = \begin{bmatrix} 18 \\ 0 \\ 5 \end{bmatrix} = \begin{bmatrix} S \\ A \\ F \end{bmatrix}$$

Similarly for IDE $\rightarrow \begin{bmatrix} 8 \\ 14 \\ 4 \end{bmatrix}$

$$P = \begin{bmatrix} 3 & 7 & 16 \\ 2 & 6 & 17 \\ 9 & 8 & 20 \end{bmatrix} \begin{bmatrix} 8 \\ 14 \\ 4 \end{bmatrix} \bmod 26 = \begin{bmatrix} 186 \\ 168 \\ 264 \end{bmatrix} \bmod 26 = \begin{bmatrix} 4 \\ 12 \\ 4 \end{bmatrix}$$

$$P = \begin{bmatrix} E \\ M \\ E \end{bmatrix}$$

Again, for $\begin{bmatrix} Y \\ Q \\ O \end{bmatrix} \rightarrow \begin{bmatrix} 24 \\ 16 \\ 14 \end{bmatrix}$

Corresponding to that,

$$P = K^{-1}C \bmod 26 = \begin{bmatrix} 3 & 7 & 16 \\ 2 & 6 & 17 \\ 9 & 8 & 20 \end{bmatrix} \begin{bmatrix} 24 \\ 16 \\ 14 \end{bmatrix} = \begin{bmatrix} 18 \\ 18 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} S \\ S \\ A \end{bmatrix}$$

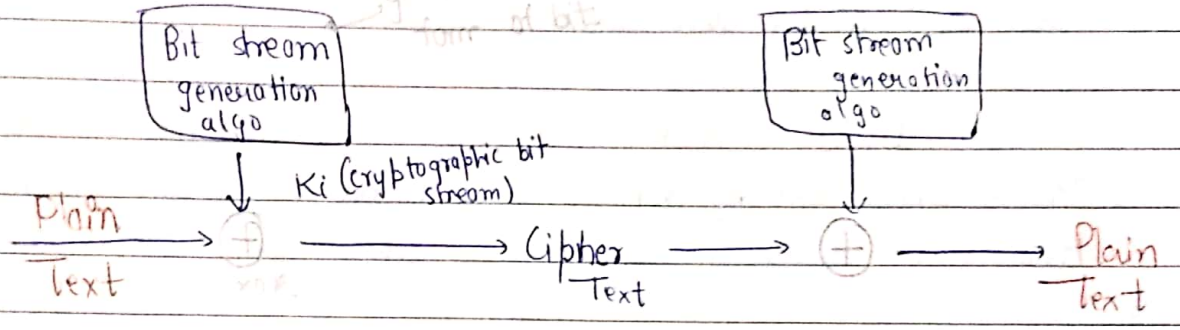
and similarly for $\begin{bmatrix} C \\ A \\ A \end{bmatrix}$ we will get $\begin{bmatrix} G \\ E \\ S \end{bmatrix}$ after solving.

$\therefore$ HDSIOEYQOCAA becomes SAFEMESSAGES.

† Stream & Block Cipher:

① Stream Cipher

- It is the one that converts (encrypts) a digital data stream one bit or 1 byte at a time.
- It is a symmetric key cipher



eg.

```
  1  0  1  1  0  1  1  0   ← message at sender side
⊕ 0  1  0  1  0  1  0  1   ← key
  1  1  1  1  0  0  0  1  1 ← cipher
```

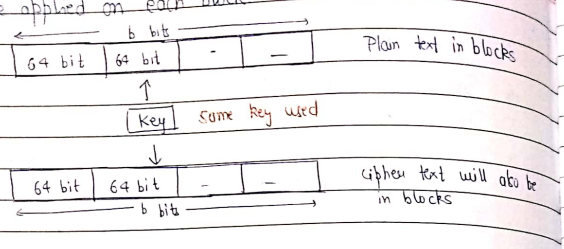
To decrypt,

```
  1  1  1  1  0  0  0  1  1 ← cipher
⊕ 0  1  0  1  0  1  0  1   ← key
  1  0  1  1  0  1  1  0   ← plain text at receiver side
```

② Block Cipher

- In this, a block of ~~cipher~~ plain text is treated as a whole & used to produce the ciphertext of equal length.
- Typically, block size of 64 & 128 bits is used
- Symmetric Cipher (Only 1 key used)

- Key will be applied on each block



eg. DES (64 bit block size)

	BLOCK CIPHER	STREAM CIPHER
1.	Plain → cipher text by taking plain text's block at a time	1. 1 bit or 1 byte of plain text to cipher text
2.	It uses 64 bits or more	2. Stream cipher uses 8 bits
3.	Complexity of block cipher is simple	2. While stream cipher is more complex
4.	Uses confusion as well as diffusion concept	4 Uses only confusion concept
5.	In this reverse encrypted text is hard	5 Reverse encrypted text is easy (we have to do XOR again)
6.	ECB (electronic code block) CBC (cipher block chaining) algorithms modes are used	6. CFB (cipher Feedback) OFB (output feedback) algorithm modes used

* Shannons theory of confusion & diffusion:

① The terms confusion & diffusion were introduced by Claude Shannon.

② Shannon's concern was to prevent crypt analysis, based on statistical analysis.

The reason is –

Assume attacker has some knowledge of the statistical characteristic of the plain text (eg in a msg, the frequency distribution of various letters may be known).

If theseee statistics are in any way reflected in the cipher text, the crypt analyst i.e. attacker may be able to deduce the encryption key.

- Thus shannon suggested 2 methods for frustating the attackers.
  1. Confusion } Properties for creating a
  2. Diffusion } secure cipher

* **DIFFUSION** → In simple words, if a symbol in the plaintext is changed, several or all symbols of in the cipher text will also change.

→ The idea of diffusion is to hide the relationship b/w the ciphertext & plaintext.

Acc. to wikipedia, Diffusion means that if we change a single bit of the plain text, then (statically) half of the bits in the ciphertext should change & vice-versa.

* **CONFUSION** → It hides the relationship between cipher text & the key.

↳ is maintained as complex as possible

→ If a single bit of in the key is changed then most/all bit of the ciphertext will also be changed.

Acc to wikipedia, confusion means that each bit of the ciphertext should depend upon several parts of the key, obscuring the connection b/w two.

make unclear or
difficult to understand

Summary: Diffusion - Make statistical relation b/w plaintext and ciphertext as complex as possible

Confusion - Makes relation b/w key & ciphertext as complex as possible

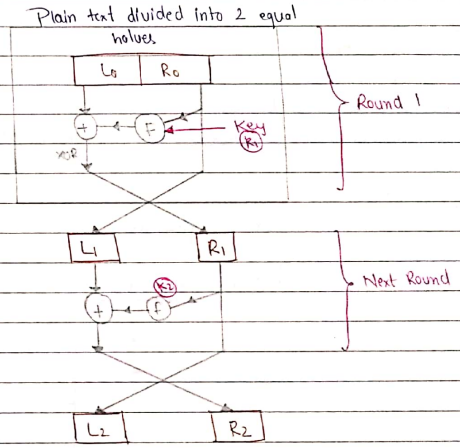
* Feistel Cipher Structure:

- Most of the block cipher technique follows this structure

(i) The plain text is divided into 2 equal halves L_0 and R_0.
→ The 2 halves of the data pass through n rounds of processing and then combined to produce the ciphertext block
→ On the right half we apply a function and in the function we will use a subkey generated from the master key.
The O/P of this is XOR ed with the left half & then their O/P will be swapped.
This is one single round.

→ We will have n rounds ← depends on Algo
All rounds will have same structure

In any algo, we divide the plaintext in 2 halves & apply the function on RHS & XOR it with LHS and the O/P is swapped then, that algo follows feistel structure.



Now,

① Block Size – Larger block size, more security
② Key Size - Larger key size means more security but may decrease the speed of encryption/decryption.
③ No. of rounds - More rounds, more secure
④ Subkey Generation Algo - More complex algo, harder for attacker to steal data.
⑤ Function/Round function - More complex function, harder for the crypt analyst to attack.

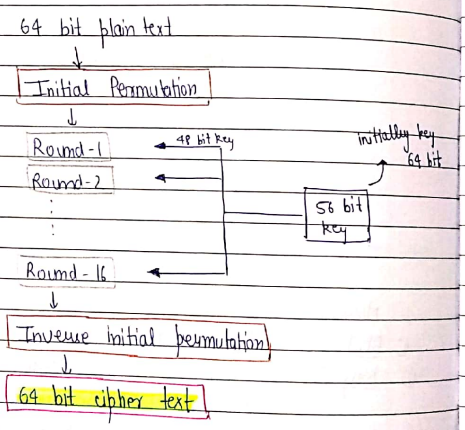
* Data Encryption Standard (DES):

1. Block Cipher
2. Symmetric cipher
3. 64 bit plain text block (it encrypts data in blocks of size 64 bit)
4. 16 rounds of fiestal round

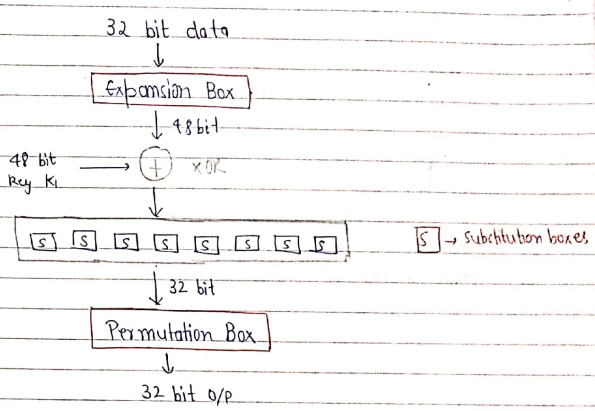
- Steps -

i) Initial permutation
ii) 16 fiestel rounds
iii) Swapping / left right swap
iv) Final permutation / Inverse initial permutation

- Basic Structure -

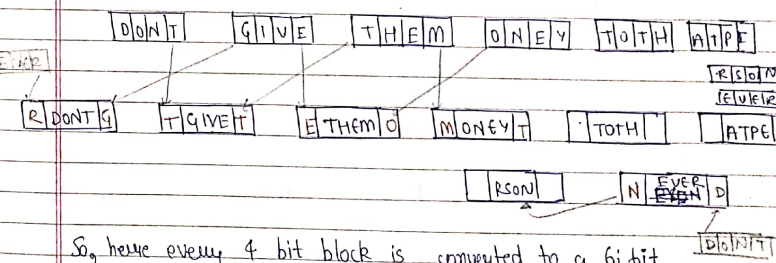


* Function definition:



* What happens in expansion box?

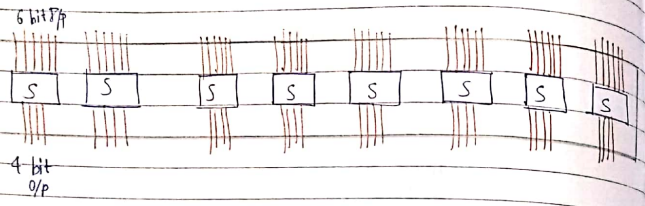
32 bit data will be → 16 1's & 0's
but for explaination let us consider a text



So, here every 4 bit block is converted to a 6 bit block.

There were 8 blocks of 4 bit each = 32 bit
Now, there are 8 blocks of 6 bit each = 48 bit
Now, these 48 bits XOR with 48 bit key & given/sent to S-boxes

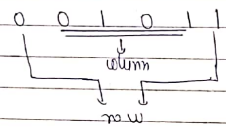
What happens in S boxes



O/P → 4 × 8 = 32 bits

- How 6 bit converted to 4 bit ?!

eg



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	3	5	7	0	1	4								1	4	5
1	4	2	1	0	9	7								2	3	6
2	16													3	4	0
3														11	10	3

number will be filled

Each S box will have a diff table.

How 16 subkeys are generated ?!

→ Actually, we have a 64 bit key in starting which go as a I/P into PC-1 (Permuted choice-1) and we get O/P as 56 bit key.

- Inside PC-1

64 bit key divided into 8 parts each of 8 bit [8 * 8 = 64 bit]

1 2 3 4 5 6 7 (8) — 1st part

9 10 11 12 13 14 15 (16) — 2nd part

57, 59, (64) — 8th part

From each part, last bit is discarded

i.e. bit → 8, 16, 24, 32, ... 64 are discarded

Hence, we have 8 parts of 7 bit each = 8 * 7 = 56 bits

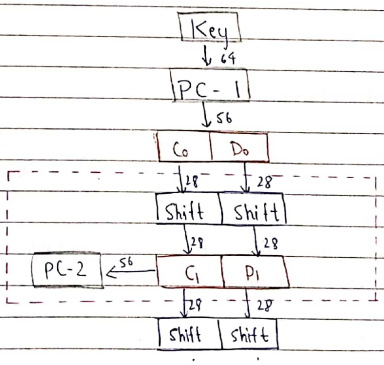
→ O/P of PC-1 is 56 bits which is divided into 2 parts of 28 bits each → C_0, D_0

→ Now, these bits are shifted with left shift in each round.

in Rounds i = 1, 2, 9, 16 → 1 shift i.e. rotated left by 1 bit

in other round i = 3, 4, ... 14, 15 → two halves rotated left by 2 bits

→ After shifting, we get (C_1, D_1) which goes as I/P in PC-2



Inside DES → 56 bit → 48 bit
using a predefined table
Then we get our 1st key
for round 1

C_1 → 28 bits → (1-28)
D_1 → 28 bits → (29-56)
Now ^56 bits, 48 bits are selected !!

Left half of C_1 (9, 18, 22, 25 position bits are missing)

Right half D_1 (35, 38, 43, 54 position bits are missing i.e discarded)

† DES Analysis

Properties

① Avalanche effect - It means a small change in plaintext (or key) should create a significant change in the ciphertext

DES has been proved to be strong with regard to this property.

eg: Plain → 0000 0000 0000 0000
Cipher → 4789 FD47 6E82 A5F1

Key used is same say
Key = 2223 4512 987A BB23

Plain → 0000 0000 0000 0001
Cipher → 0A4E D5C1 5A63 FEA3

Although, the two plain texts differ only by 1 bit, cipher text block differs a lot/significantly

② Completeness effect - It means that each bit of the ciphertext needs to be depend on many bits of the plain text

The confusion & diffusion produced by D-boxes & S-boxes in DES, show a very strong completeness effect

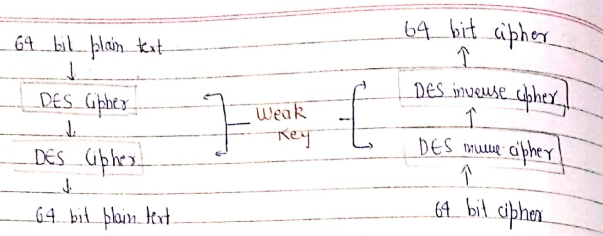
† DES Weakness: ① Key Size

- Critics believes that the most serious weakness of DES is key size of 56 bits
- Because, with todays technology (like parallel processing & very powerful processor) it can be easily cracked.
  ∴ 2^{56} keys (brute force attack)
- Thus, we use triple DES (3DES) with two keys (112 bits) or triple DES with three keys (168 bits).

② Weak keys:

- Four out of 2^{56} keys are called weak keys.
- A weak key is a key which after parity drop consists of operation all 0's, all 1's or half 0's & half 1's.
- The disadvantage of using weak key is that if we encrypt a block with a weak key and subsequently encrypt the result the same weak key, we get the original block
- The process creates the same original block if we decrypt the block twice.

So, if after 2 decryptions, if the result is same, then attacker is successful.



③ Semi-Weak Keys: Six
- Six key-pairs are called semi-weak keys.
- A semi weak key creates only two different round keys, and thus each of them is repeated 8 times.

④ Possible Weak Keys:
- There are 48 keys that are called possible weak keys.
- A possible weak key is a key that creates only 4 different round keys, in other words, the 16 round keys are divided into 4 groups & each group is made of 4 equal keys.

⑤ Key-Clustering:
- Means 2 or more diff keys can create the same cipher-text from the plain text.

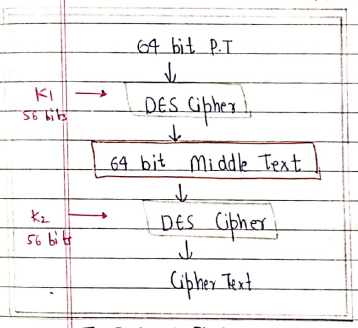
⑥ Weakness in Cipher Design:
- Two specifically choosen IP's to s-box array can create the same O/P.

* Multiple DES: Since DES attack was vulnerable to brute force attack, variation of DES called multiple DES were introduced.

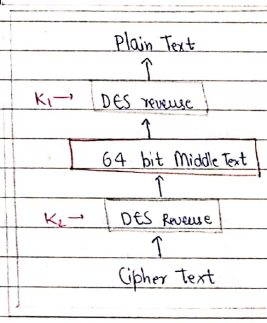
① Double DES ② Triple DES

* Double DES:
- Uses 2 different keys
  (56 + 56) = 112 bit key
- Double encryption occurs as follows:

$$P \rightarrow E(K_1, P)$$
$$\downarrow$$
$$E(K_2, E(K_1, P)) = \text{Cipher}$$



ENCRYPTION BLOCK



DECRYPTION BLOCK

Keys are applied in reverse order

Plain = D(K1, D(K2, C))

- For decryption,
  1st decrypted using key K_2 which produces single encrypted cipher text.
  This 64 bit middle text / temp. cipher text is then decrypted using the key K_1 to get plain text.

* Drawbacks of Double DES:

MEET-IN-THE-MIDDLE attack

This attack involves encryption from 1 end and decryption from the other end and then "Matching the results in the middle" and hence the name.

Q. Explain meet in the middle (MIM) attack.

The attack requires knowing some plaintext / ciphertext pairs. Lets assume,

Cipher Text → C , Plain Text → P

The attack proceeds as follows -

i) Encrypt P for all 2^{56} possible values of K_1 and store the results in a table and sort it.

ii) Now, decrypt C using all 2^{56} possible values of K_2. As each decryption is produced, check against the table for a match.

iii) When there is a match, we have located a possibly correct pair of keys.

Note: Now, more than 1 pair of keys may result in a match, but the number of pairs will be small we should try each possible pair of keys.

Some pairs of plain text known → encrypt pairs for all 2^{56} possible values of K_1

plain	open	middle

Cipher text known → decrypt pairs for all 2^{56} possible values of K_2

cipher	plain	middle

Sort the results in table 1 (K_1, K_2') (K_2', K_2)

→ We will compare these values with the values of the 1st table computed earlier.

Decrypt (K_2, C) = Encrypt (K_1, P)

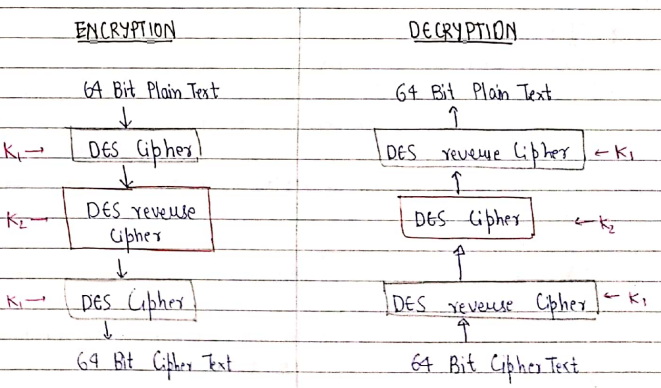
$\therefore (K_1, K_2)$ is key pair used.

→ So, it takes twice as long to break double DES using brute force.

Because DES has 2^{56} bit security, double DES has $2 \times 2^{56} = 2^{57}$ security.

* Triple DES:

① Using 2 Keys



② Using 3 Keys

ENCRYPTION	DECRYPTION
64 Bit PT	64 Bit PT
↓	↑
K_1 → DES Cipher	K_1 → DES reverse Cipher
↓	↑
K_2 → DES reverse Cipher	K_2 → DES Cipher
↓	↑
K_3 → DES Cipher	K_3 → DES reverse cipher
↓	↑
64 Bit Cipher	64 Bit Cipher

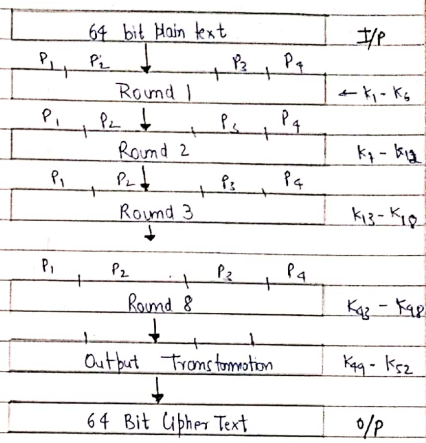
* IDEA Algorithm :

- IDEA stands International Datat Encryption Algorithm.
- Originally called IPES (Improved Proposed Encryption standard)
- Symmetric key block cipher (designed by James Massey & Xuejia Lai) in 1991
- It was intended as a replacement for the DES. It is reversible like DES.

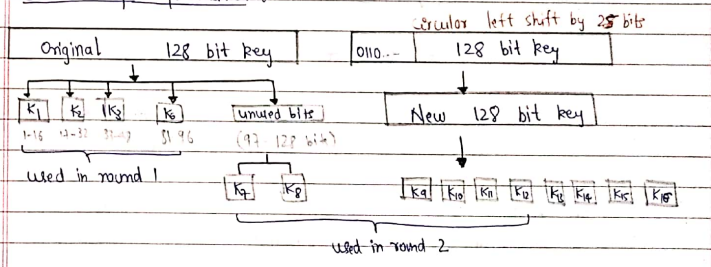
Key Size → 128 bits	→ from which we will generate 52 subkeys
Block Size → 64 bits	→ In each round, block divided into 4 portions/parts (16 bit each)
8 Identical Transformation Rounds	→ In each round, 6 subkeys are used (16 bit each)
One half round i.e the output transformation	→ It use 4 subkeys (16 bit each) → O/P after this round gives cipher text (64 bit)

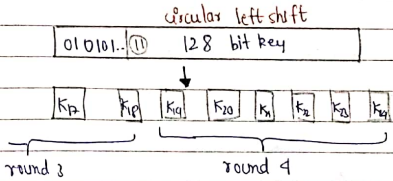
→ I/P divided into 4 portions (P_1 to P_4) 16 bits each.

- There are 8 similar rounds
- Each round uses 6 subkeys (16 bits each)
- Last round i.e output transformation produces the cipher text & uses 4 subkeys (16 bit each).



→ 52 Subkeys Generation :





→ Single Round Details :-

$S_1 = P_1 \times K_1$
$S_2 = P_2 + K_2$
$S_3 = P_3 + K_3$
$S_4 = P_4 \times K_4$

$S_5 = S_1 \oplus S_3$
$S_6 = S_2 \oplus S_4$

$S_7 = S_5 \times K_5$
$S_8 = S_6 + S_7$
$S_9 = S_8 \times K_6$
$S_{10} = S_7 + S_9$

$S_{11} = S_1 \oplus S_9$ → new P_1
$S_{12} = S_3 \oplus S_9$ → new P_2 (swap)
$S_{13} = S_2 \oplus S_{10}$ → new P_3 (swap)
$S_{14} = S_4 \oplus S_{10}$ → new P_4

In short,
6 subkeys used in 1 round
6 times XOR

→ Output Transformation (one half round)

$P_1 \times K_{49} \rightarrow C_1$
$P_2 + K_{50} \rightarrow C_2$
$P_3 + K_{51} \rightarrow C_3$
$P_4 \times K_{52} \rightarrow C_4$

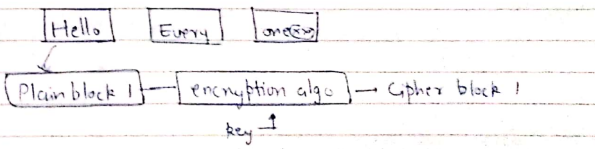
- Takes place at the end of 8th round
- I/P to this block is a 64 bit value divided into 4 sub blocks (say P_1, P_2, P_3 & P_4).

† Block Cipher Modes of Operation :
For different types of messages, we need different modes of operations.

5 modes of operations are :
i) ECB - Electronic Codebook Mode
ii) CBC - Cipher Block Chaining Mode
iii) CFB - Cipher feedBack Mode
iv) OFB - Output feed Back Mode
v) CTR - Counter Mode

† ECB (Electronic Code Book) :
- Simplest mode of operation
- Plain text is divided into a no. of fixed size block.
- If message is not a multiple of block size, then padding is done.
- Take one block at a time & encrypt it
- Same key used for encryption & decryption.

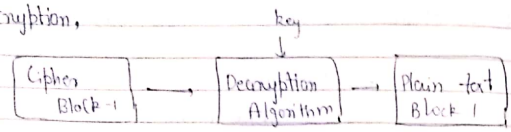
eg. Let block size = 5
Plain text → Hello Everyone



This will happen for all the blocks.

Note: ① Best for short amount of data, such as a key
② Not secure for lengthy data.
③ If identical blocks appear, then this mode produces same cipher

For decryption,

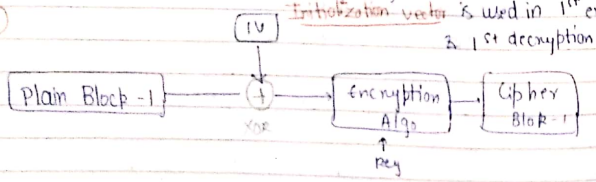


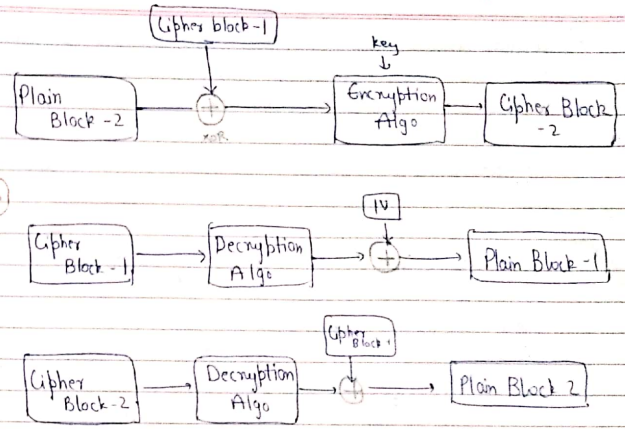
This will happen for every block

CBC (Cipher Block Chaining):

- To overcome security issues of ECB code (b/c in ECB if same block appears twice then cipher text produced will be same)
- I/P to the encryption algo is XOR of the current plain text block and the preceding ciphertext block. So, repeating patterns not exposed.
- Same key for encryption & decryption.

Encryption





- IV should be known to both parties, but should be unpredictable by the 3rd party.
- So, we can use ECB encryption to encrypt IV to ensure max security

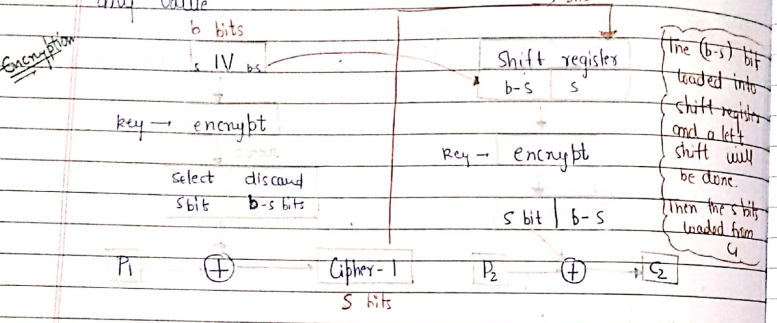
Note: Now, if we have 2 diff blocks it will produce different ciphers
Limitation - If we have 2 identical msgs & if we use same IV, ciphers will be same.

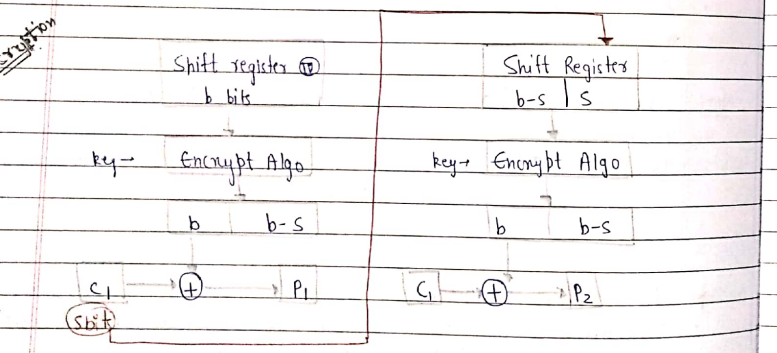
CFB (Cipher Feed Back)

- In this mode cipher is given a feedback to the next block of encryption with some new specifications.
- I^st an initial vector IV (b bits) is used for 1st encryption & the O/P bits are divided as set of 's' and 'b-s' bits.

- The left hand side 's' bits are selected & applied an XOR operation with the plain text bits. This result is given to a shift register & the process continues.

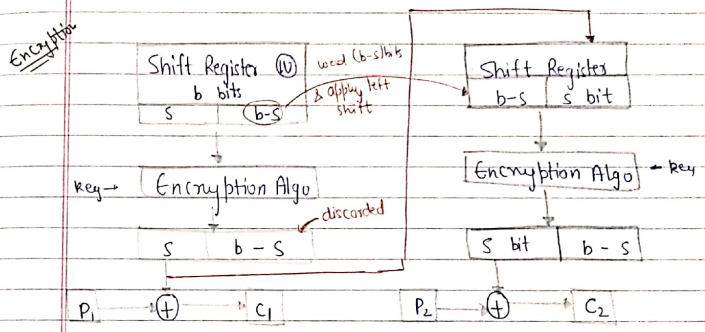
The plaintext is divided into segment of 's' bits. 's' can have any value





* OFB (Output feed Back):

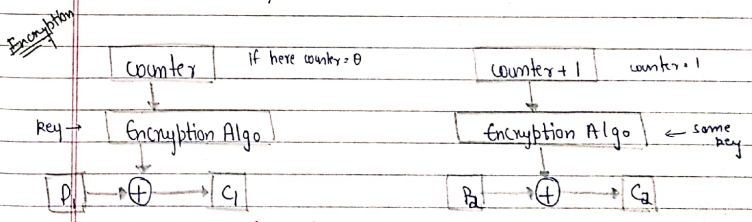
- Similar in structure to CFB
- The O/P of encryption function that is feedback to the shift register in OFB, whereas in CFB the cipher text unit if feed back to the shift register.



For decryption, ci & pj will be exchanged.

* Counter Mode (CTR):

- Simple & fast
- A counter, equal to the plain text block size is used.
- Counter is initialized to some value & then incremented by 1 for each subsequent block.



For decryption, change P_i & C_i