Q.7 Explain Hill cipher with example.

→ . Multi-letter cipher
. Developed by Lester Hill in 1929
. Encrypts group of letters : digraph, trigraph / polygraph
. This can be expressed as :

$C = E(k, P) = P \times k \bmod 26$

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

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

encryption

$$\begin{aligned} C_1 &= (P_1 k_{11} + P_2 k_{21} + P_3 k_{31}) \bmod 26 \\ C_2 &= (P_1 k_{12} + P_2 k_{22} + P_3 k_{32}) \bmod 26 \\ C_3 &= (P_1 k_{13} + P_2 k_{23} + P_3 k_{33}) \bmod 26 \end{aligned}$$

ex: Encrypt "pay more money" using Hill cipher with key $\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$.

→

p	a	y	m	o	r	e	m	o	n
15	0	24	12	14	17	4	12	14	13

e	y
4	24

. key = 3x3 matrix
PT = pay mor emo ney

Encrypting Pay

$(C_1\ C_2\ C_3) = (15\ 0\ 24)\begin{pmatrix}17 & 17 & 5\\ 21 & 18 & 21\\ 2 & 2 & 19\end{pmatrix} \bmod 26$

$= (15\times17+0\times21+24\times2 \quad 15\times17+0\times18+24\times2$
$15\times5+0\times21+24\times19) \bmod 26$

$= (803\ \ 303\ \ 531) \bmod 26$
$= (17\ \ 17\ \ 11)$
$(R\ \ R\ \ L)$

Encrypting mor

$(C_1\ C_2\ C_3)\ (12\ 14\ 17)\begin{pmatrix}17 & 17 & 5\\ 21 & 18 & 21\\ 2 & 2 & 19\end{pmatrix} \bmod 26$

$= (12\times17+14\times21+17\times2 \quad 12\times17+14\times18+17\times2$
$12\times5+14\times21+17\times19) \bmod 26$

$= (532\ \ 490\ \ 677) \bmod 26$
$= (12\ \ 22\ \ 1)$
$= (M\ \ W\ \ B)$

Encrypting emo

$(C_1\ C_2\ C_3) = (4\ 12\ 14)\begin{pmatrix}17 & 17 & 5\\ 21 & 18 & 21\\ 2 & 2 & 19\end{pmatrix} \bmod 26$

$= (4\times17+12\times21+14\times2 \quad 4\times17+12\times18+14\times2$
$4\times5+12\times21+14\times19) \bmod 26$
$= (348\ \ 312\ \ 538) \bmod 26$
$= (10\ \ 0\ \ 18)$
$= (K\ \ A\ \ S)$

Encrypting ney

$(C_1\ C_2\ C_3) = (13\ 4\ 24)\begin{pmatrix}17 & 17 & 5\\ 21 & 18 & 21\\ 2 & 2 & 19\end{pmatrix} \bmod 26$

$= (13\times17+4\times21+24\times2 \quad 13\times17+4\times18+24\times2$
$13\times5+4\times21+24\times19) \bmod 26$

$= (348\ 312\ \ 538) \bmod 26$
$= (15\ \ 3\ 7) \quad (P\ D\ H)$

plaintext : pay more money
ciphertext : RRL MWBKASPDH

Decryption requires k^{-1}, inverse matrix k.

$k^{-1} = \frac{1}{\text{Det}\,k} \times \text{Adj}\,k$

To find Det k, Adj k
To find determinant of k $\begin{pmatrix}17 & 17 & 5\\ 21 & 18 & 21\\ 2 & 2 & 19\end{pmatrix}$

$$\det\begin{pmatrix}17 & 17 & 5\\ 21 & 18 & 21\\ 2 & 2 & 19\end{pmatrix} \bmod 26 = 17(18\times19-2\times21) - 17(19\times21-2\times21) + 5(2\times21-2\times18) \bmod 26$$

$= 17(300) - 17(357) + 5(6) \bmod 26$

$= -939 \bmod 26$

$= -3 \bmod 26 = 23$

To find Adjacent k

$$\text{Adj } k = \begin{vmatrix}17 & 17 & 5\\ 21 & 18 & 21\\ 2 & 2 & 19\end{vmatrix}$$

$$= \text{Adj}k = \begin{matrix}17 & 17 & 5 & 17 & 17\\ 21 & 18 & 21 & 21 & 18\\ 2 & 2 & 19 & 2 & 2\\ 17 & 17 & 5 & 17 & 17\\ 21 & 18 & 21 & 21 & 18\end{matrix}$$

$$\text{Adj}k = \begin{bmatrix}18\times19 - 2\times21 & 2\times5-17\times19 & 17\times21-18\times5\\ 21\times2 - 19\times21 & 19\times17 - 5\times2 & 5\times21-21\times17\\ 21\times2 - 2\times18 & 2\times17 - 17\times2 & 17\times18-21\times17\end{bmatrix}$$

$$= \begin{bmatrix}300 & -313 & 267\\ -357 & 313 & -252\\ 6 & 0 & -51\end{bmatrix} \bmod 26$$

$$= \begin{pmatrix}14 & 25 & 7\\ 7 & 1 & 8\\ 6 & 0 & 1\end{pmatrix} \bmod 26$$

$$k^{-1} = \frac{1}{23} \times \begin{pmatrix}14 & 25 & 7\\ 7 & 1 & 8\\ 6 & 0 & 1\end{pmatrix} \bmod 26$$

$$k^{-1} = 23^{-1} \times \begin{pmatrix}14 & 25 & 7\\ 7 & 1 & 8\\ 6 & 0 & 1\end{pmatrix} \bmod 26$$

$$k^{-1} = 17 \times \begin{pmatrix}14 & 25 & 7\\ 7 & 1 & 8\\ 6 & 0 & 1\end{pmatrix} \bmod 26$$

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

Decrypting : RRL

$$(p_1\ p_2\ p_3) = (R\ R\ L)\begin{pmatrix}4 & 9 & 15\\ 15 & 17 & 6\\ 24 & 0 & 17\end{pmatrix} \bmod 26$$

$\leftarrow$ decryption

$$(p_1, p_2, p_3) = (17\ 17\ 11)\begin{pmatrix}4 & 9 & 15\\ 15 & 17 & 6\\ 24 & 0 & 17\end{pmatrix} \bmod 26$$

$= (17\times4 + 17\times15 + 11\times24 \quad 17\times9 + 17\times17 + 11\times0 \quad 17\times15 + 17\times6 + 11\times17) \bmod 26$

$= (15\ 0\ 24) = (P\ A\ Y)$

Decrypting : MWB

$$(P_1\ P_2\ P_3) = (M\ W\ B) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$= (12\ 14\ 17) = (M\ O\ R$

Decrypting KAS

$$(P_1\ P_2\ P_3) = (10\ 0\ 18) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$= (10\times4 + 0\times15 + 18\times24 \quad 10\times9 + 0\times17 + 18\times0$
$10\times15 + 0\times6 + 18\times17) \bmod 26$
$= (4\ 12\ 14) = (E\ M\ O)$

Decrypting PDH

$$(P_1\ P_2\ P_3) = (15\ 3\ 7) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$= (13\ 4\ 24) = (N\ E\ Y)$

Ciphertext: RRL MWB KAS PDH
Plaintext : Pay mor emo ney

Q.8 What are block cipher principles?

→ A block cipher is designed by considering its 3 critical aspects which are listed as below.

1) No. of rounds
2) Design of function F
3) Key Schedule Algorithm

1> No of Rounds

The no of rounds judges strength of block cipher algorithm. It is considered that more is no of rounds, difficult is for cryptoanalysis to break algorithm.

2> Design of function F

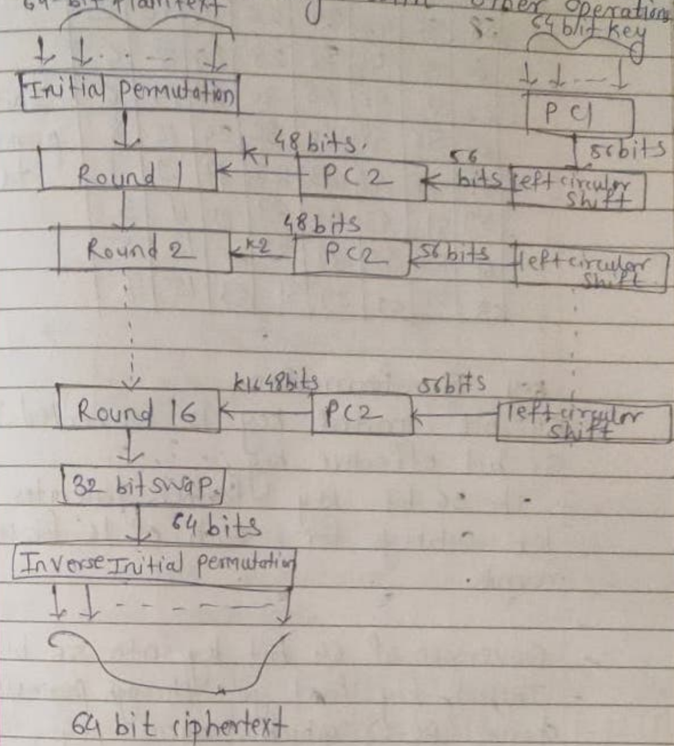
- The fn f of the block cipher must be designed such that it must be impossible for any cryptanalysis to unscramble the substitution.
- More function F is non-linear, more it would be difficult to crack it. Well, while designing function F it should be confirmed that it has good avalanche property which states that a change in one-bit of i/p must reflect the change in many bits of o/p.
- The fun F should be designed such that it possesses bit independence criterion.

which states that o/p bits must change independently if there is any change in i/p bit.

3) key schedule Algo.
- It is suggested that key schedule should confirm the strict avalanche effect & bit independence criterion.

Q.9 Explain DES encryption & decryption.
→ DES is symmetric block cipher. By symmetric we mean that size of i/p text & o/p text is same.
- It is block cipher that encrypts 64 bit blocks.
- It takes 64 bit plaintext i/p & generates corresponding 64 bit ciphertext o/p.
- Main key length is 64 bit which is transformed into 56 bits by skipping every 8 bit in key.
- It encrypts text in 16 rounds where each round uses 48 bit subkey.

Encryption:
- DES is based on 2 attributes of feistel str i.e substitution & Transposition.
- DES consist of 16 steps, each of which is called round.
- Each round perform steps of substitution & transposition along with other operations.



Initial permutation
64-bit PT block is i/p into IP fun that rearranges order of bits. The order of bits is changed using predefined table.

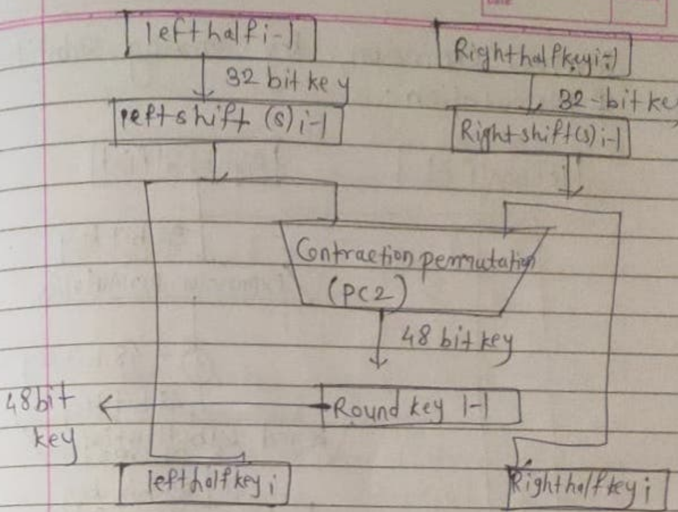
58	50	42	34	26	18	10	2
60	52	44	36	28	28	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Initial permutation table

- key Transformation
- 64-bit Initial key is Converted into 56 bit effective key.
- The 56 bit key further generates 48 bit subkeys for each of 16 feistel rounds.

- Conversion of 64 bit key into 56 bit key
- Initial key first go through permuted choice 1 (pc-1) which reduces key to 56 bits
- In pc-1 every 8th bit in key is discarded. That is bit positions 8, 16, 24, 32, 40, 48, 56 & 64 are discarded.

- Generating 48 bit Round subkey.

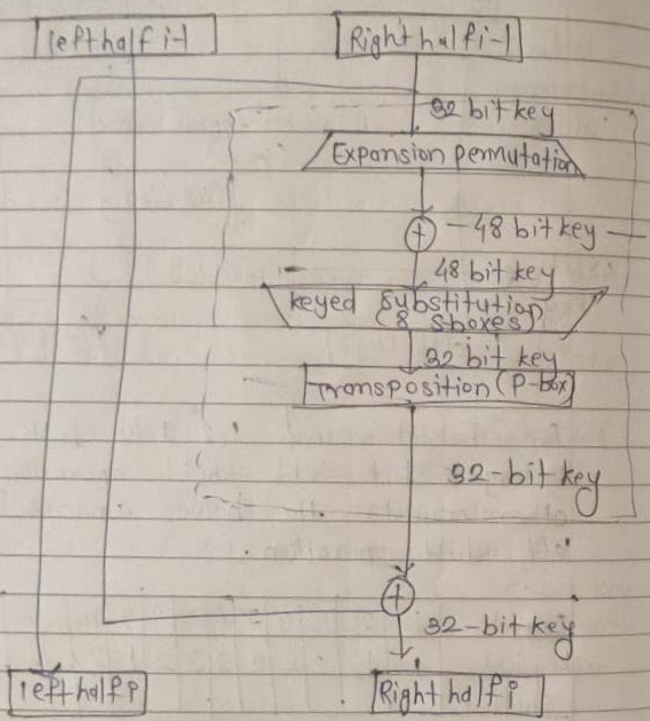


For Feistel round 1, 2, 9, 16 both halves undergo 1-bit left shift operation. for others rounds the halves undergo 2-bit left shift operation.

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

Feistel Rounds (1-16)
- Every round receives 64-bit permuted pt from IP fun & 48 bit transformed key (ki)
- The permuted 64 bit PT is divided into 2 halves called left PT & right PT. Both of these halves are 32 bit in size.
- The right half or RPT is processed using magler (F) fun. Mangler function

involves expansion, key mixing, substitution & permutation.



- 32-bit swap & Inverse IP.
  After 16 rounds we get 2 blocks of 32-bit each.
- The two 32 bit halves are again swapped back, resulting in 64-bit block.
- This step is called 32-bit swap in DES encryption algo.

Decryption

- Decryption in DES follows same process as encryption but in reverse order.
- since DES is symmetric key algo same key is used for both encryption & decryption, but subkeys are applied in reverse order.

- Reverse subkey Application: 16 rounds keys generated during key scheduling are used in reverse order during decryption
- Inverse Feistel function: Feistel n/w str. ensures that decryption mirrors encryption.
- Final permutation: After 16 rounds, o/p undergoes inverse initial permutation, reversing initial shuffling.

Q.10 Explain diffusion & confusion with difference

→ Confusion:

- Mixing up relationship b/w ciphertext & key so that it becomes hard to guess key even if someone knows how CT looks.
- Goal: To make connection b/w CT & encryption key as complex as possible.
- Achieve: usually by using substitution.
- ex: In substitution cipher, each letter is replaced by another. This hides how

key affects ciphertext.

Diffusion:
- Spreading out influence of 1 PT bit over many CT bits.
- Goal: To make sure that small change in PT changes many bits in CT.
- Achieve: usually by using permutation/transposition.
- ex: In transposition cipher, letters are rearranged, so the pattern of PT is hidden.

Feature	Confusion	Diffusion
purpose	Makes relationship b/w key & CT complex	spreads PT inf throughout CT
Main technique	substitution	permutation/transposition
effect	Hides how key affects CT	Hides statistical str. of PT
ex	substitution cipher	Transposition cipher
if missing	key may be guessed easily	Patterns in PT may still appear in CT.

Q.11 Difference b/w Stream & Block cipher

Feature	Stream cipher	Block cipher
Data processing	Encrypts 1 bit / byte at a time	Encrypts a block at a time
Speed	usually faster & simple	slightly slower
Error propagation	Error affects only 1 bit / byte	Error affects whole block.
key usage	uses keystream generator	uses same key for entire block.
Ex. Algo	RC4, A5/1	AES, DES, Blowfish
Best used for	streaming data	static data