SUBHAPREET PATRO
2211CS010547
GROUP-3

CNS ASSIGNMENT - 1

1) (a) Explain the concepts of CIA principle of security with examples.

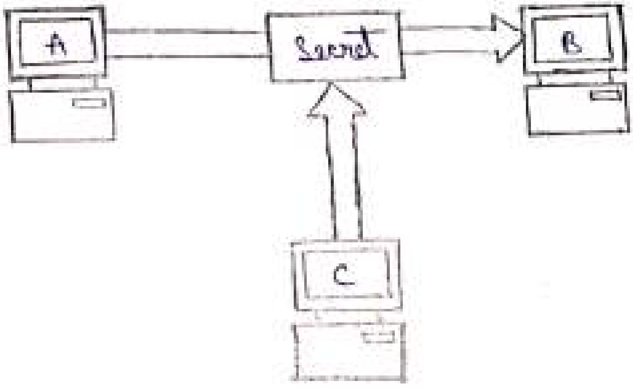
Ans:- Principles of Security:-

The principles of security can be classified as follows:-

• Confidentiality:-

→ The degree of confidentiality determines the secrecy of the information. The principle specifies that only sender & receiver will be able to access the information shared between them. Confidentiality compromises if an unauthorized person is able to access a message

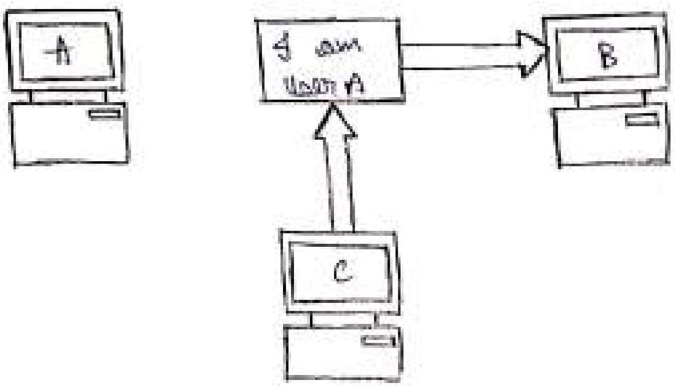
→ For example, let us consider sender A wants to share some confidential information with receiver B & the information gets intercepted by the attacker C. Now the confidential information is in the hands of an intruder C.



3

• Authentication:-

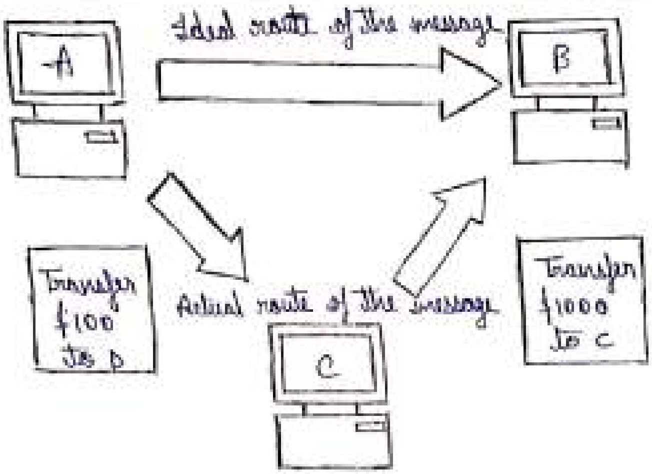
Authentication is the mechanism of identifying the user or system or entity. It assures the identity of the person trying to access the information. The authentication is mostly secured by using username & password. The authorized person whose identity is preregistered can prove his/her identity & can access the sensitive information.



• Integrity-

→ Integrity gives the assurance that the information received is exact & accurate. If the content of the message is changed after the sender sends it but before reaching the intended receiver, then it is said that the integrity of the message is lost.

→ For example, suppose A writes a check for $100 to pay for goods bought from the US. However, when A see in his next account statement, he will be startled to see that the check resulted in a payment of $1000! This is the case for loss of message integrity. Here, user C tampers with a message originally sent by user A, which is actually destined for user B. User C somehow manages to access it, change its contents, & send the changed message to user B. User B has no way of knowing that the contents of the message were changed after user A had sent it. User A also does not know about this change. This type of attack is called modification. Modification causes loss of message integrity.



(b) Demonstrate the substitution techniques in classical cryptography with examples.

Ans. Substitution technique is a classical encryption technique where the characters present in the original message are replaced by the other characters or numbers or by symbols. If plaintext is considered as the string of bits, then the substitution technique would replace bit pattern of plain text with the bit pattern of cipher text.

Types of Substitution Techniques:-

1. Caesar Cipher
2. Monoalphabetic Cipher
3. Playfair Cipher
4. Hill Cipher
5. Polyalphabetic Cipher
6. One-Time Pad.

(i) Caesar Cipher:-

This is the simplest substitution cipher by Julius Caesar. In this substitution technique, to encrypt the plain text, each alphabet of the plain text is replaced by the alphabet three places further it. And to decrypt the cipher text each alphabet of cipher text is replaced by the alphabet three places before it.

Let us take an example:-

Plain Text:- meet me tomorrow

Cipher Text:- phhw ph wrpruurz

Look at the example above, we have replaced 'm' & 'p' which occur three places after 'm'. Similarly, 'e' is replaced with 'h' which occurs in three places after 'e'.

Note:- If we have to replace the letter 'z' then the next three alphabets counted after 'z' will be 'a', 'b', 'c'. So, while counting further three alphabets, if 'z' occurs it circularly follows 'a'.

There are also some drawbacks of this simple substitution technique. If the hacker knows that Caesar Cipher is used then to perform brute force cryptanalysis, he has only to try 25 possible keys to decrypt the plain text.

(ii) Monoalphabetic cipher:-

Monoalphabetic cipher is a substitution cipher, where the cipher alphabet for each plain text alphabet is fixed, for entire encryption.

In simple words, if the alphabet 'p' in the plain text is replaced by the cipher alphabet 'd'. Then in the entire plain text wherever alphabet 'p' is used, it will be replaced by the alphabet 'd' to form the ciphertext.

(iii) Playfair Cipher :-

Playfair Cipher is a substitution cipher which involves a 5 X 5 matrix.

Plain text :- meet me tomorrow

Key :- KEYWORD

Now, we have to convert this plain text to cipher text using the given key.

Step 1 :- Create a 5×5 matrix & place the key in that matrix row-wise from left to right. Then fill the remaining alphabet in the blank space.

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

If a key has duplicate alphabets, fill those alphabets only once in the matrix and I & J should be kept together in the matrix even though they ~~seem~~ occur in the key.

Step 2 :- Now, you have to break the plain text into a pair of alphabets.

Plain text :- meet me tomorrow

pair :- me et me to mo rx ~~ow~~ ro wz

Note :-

→ Pair of alphabets must not contain the same letter. In case, pair has the same letter then break it & add 'x' to the previous letter.

Like in our example, 'rr' occurs in pair, so we have broken that pair & added 'x' to the first 'r'.

→ In case while making pair, the last pair has only one alphabet left then we add 'z' to that alphabet to form a pair as in our example, we have added 'z' to 'w' because 'w' was left alone at last.

→ If a pair has 'xx' then we break it and add 'z' to the first 'x' i.e 'xz' & 'x_'.

Step 3:- In this step, we will convert plain text into cipher text. For that, take the first pair of plain text and check for cipher alphabets for the corresponding in the matrix. To find cipher alphabets follow the rules below:-

① If both the alphabets of the pair occur in the same row replace them with the alphabet to their immediate right. If an alphabet of the pair occurs at extreme right then replace it with the first element of that row, i.e, the last element of the row in the matrix circularly follows the first element of the same row.

② If the alphabets in the pair occur in the same column, then replace them with the alphabet immediately below them. Here also, the last element of the column circularly follows the first element of the same column.

③ If the alphabets in the pair are neither in the same column nor in the same row, then the alphabet is replaced by the element in its own row & the corresponding column of the other alphabet of the pair.

Pair:- me et me to mo rx ro wz

Cipher text:- ken kee ksn kez kso ta ke yo

(iv) Hill Cipher:-

Hill Cipher is a polyalphabetic cipher introduced by Lester Hill in 1929.

Plain text:- Binary

Key:- HILL

Choose the key in such a way that it always forms a square matrix. With HILL as the key, we can form a 2×2 matrix.

Now, of plain text, you have to form a column vector of length similar to the key matrix. In our case, the key matrix is 2×2 then the column vectors of plain text would be 2×1.

The general equation to find cipher text using hill cipher is

$$\boxed{C = KP \bmod 26}$$

$$[c_1\ c_2] = \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \bmod 26$$

For our example, our key matrix would be:-

$$\begin{bmatrix} H & I \\ L & L \end{bmatrix}$$

And our plain text matrices of 2×1 will be as follows:-

$$\begin{bmatrix} B \\ I \end{bmatrix} \begin{bmatrix} N \\ A \end{bmatrix} \begin{bmatrix} R \\ Y \end{bmatrix}$$

Now, we have to convert the key matrix & plain text matrices into numeric matrices. For that, number the alphabets such as A=0, B=1, C=2, - - -, Z=25. So, considering the alphabet numbering:-

Key matrix will be:-

$$K = \begin{bmatrix} 7 & 8 \\ 11 & 11 \end{bmatrix}$$

Plain text matrices ~~two~~ would be:-

$$\begin{bmatrix} 1 \\ 8 \end{bmatrix} \begin{bmatrix} 13 \\ 0 \end{bmatrix} \begin{bmatrix} 17 \\ 24 \end{bmatrix}$$

In the first calculation, we would ~~have~~ get two cipher alphabets for plain text alphabet 'B' & 'I'

$$[c_1 \ c_2] = \begin{bmatrix} 7 & 8 \\ 11 & 11 \end{bmatrix} \begin{bmatrix} 1 \\ 8 \end{bmatrix} \text{ mod } 26$$

$$= \begin{bmatrix} 71 \\ 99 \end{bmatrix} \text{ mod } 26 = \begin{bmatrix} 19 \\ 21 \end{bmatrix} = \begin{bmatrix} T \\ V \end{bmatrix}$$

So, the cipher alphabet for plain text alphabet 'B' & 'I' is 'T' & 'V'. Similarly we have to calculate the ciphertext for the remaining plain text.

The calculated ciphertext for 'BINARY' using hill cipher is 'TVNNZJ

(v) <u>Polyalphabetic Cipher:-</u>

Polyalphabetic cipher is far more secure than a monoalphabetic cipher. As monoalphabetic cipher maps a plain text symbol or alphabet to a ciphertext symbol & uses the same ciphertext symbol wherever that plaintext occurs in the message, but, polyalphabetic cipher replaces the plain text with a different ciphertext.

(vi) <u>One Time Pad</u> :-

The one-time pad cipher suggests that the key length should be as long as the plaintext to prevent the repetition of key. Along with that, the key should be used only once to encrypt & decrypt the single message after that the key should be discarded.

Onetime pad suggests a new key for each new message & of the same length as a new message. ~~Now, let~~ :-

Plain text :- Binary

Key :- Cipher

Number the alphabets such as A=0, B=1, C=2, ---, Z=25. So, our plain text & key in numeric form would be :-

Plain text :- ~~[illegible]~~ 1 8 13 0 17 24

Key :- 2 8 15 7 4 17

Now, you have to add the number of the plain text alphabet, to the number of its corresponding key alphabet.

B + C = 1 + 2 = 3

I + I = 8 + 8 = 16

N + P = 13 + 15 = 28

A + H = 0 + 7 = 7

R + E = 17 + 4 = 21

Y + R = 24 + 17 = 41

The resultant ciphertext numbers we get are (3, 16, 28, 7, 21, 41)

If the addition of any plain text number & the key number is >26 then subtract only that particular number from 26.

N+P = 28 - 26 = 2

Y+R = 41 - 26 = 15

So, the final cipher text numbers are (3,16,2,7,21,15). Now convert this number to alphabets assuming A=0, B=1, ..., Z=25

Cipher Text:- dqchvp.

2) Let message = "INDIAN ARMY", Ignore the space between words. Keyword = "MRUH", find cipher-text using playfair cipher.

Ans:- The playfair cipher, is also called playfair square, is a cryptographic technique used for manual encryption of data. This scheme was invented by Charles Wheatstone in 1854. The playfair cipher was used by the British Army in WW-I and by Australians in WW-II.

The playfair cipher encryption scheme uses two main processes:-

① Creation & Population of matrix

② Encryption process.

Step 1:- Creation & Population of matrix:-

The playfair cipher makes use of 5x5 matrix (table) which is used to store a keyword or phrase that becomes the key for encryption & decryption. The way this is entered into the 5x5 matrix is based on some simple rules:-

① Enter the keyword in the matrix row-wise: left-to-right, & then top-to-bottom.

② Drop duplicates.

③ Fill the remaining spaces in the matrix with the rest of English alphabets (A-Z) that were not a part of our keyword. While doing so, combine I & J in the same cell of the table. In other words, if I or J is a part of the keyword, disregard both I and J while filling the remaining slots.

Steps in Encryption Process:-

To encrypt a message using the Playfair Cipher, the plaintext message is first divided into pairs of letters. If there is an odd number of letters, a "dummy" letter such as 'x' is added at the end ~~of~~ to make the message even. Each pair of letters is then encrypted using following steps:-

① If the two letters are the same, add a 'dummy' letter such as 'x' between them.

② Locate the two letters in the grid & find their positions (row & column).

③ If the two letters are in same row, replace each letter with the letter to its right (wrapping around to the beginning of row, if necessary).

④ If the two letters are in the same column, replace each letter with the letter below it (wrapping around to the top of column, if necessary).

⑤ If the two letters are not in the same row / column, replace each letter with the letter in the same row but in the column of other letter.

To decrypt a message, with playfair cipher, the reverse process is used.

Given keyword : MRUH

Given plain text : INDIAN ARMY

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

(1) First, we break the original text into pairs of two alphabets each. this means that our original text would now look like this

IN DI AN AR MY

(2) Now, we apply our playfair cipher algorithm to this text, the first pair alphabets is IN. In this case, this text is KG, which is our first cipher text applying step #3

(3) Our next text block is DI. Step #5 will apply. As we can see, our second block of cipher text is CK.

(4) Now, our third plain text block is AN. Step #4 will apply. Cipher text block would be ~~MU~~, FT.

(5) Now fourth block of plain text which is AR. Step #3 will apply. Cipher text block would be MU

(6) Fifth block of plain text is MY. Step #5 will apply. Cipher text block would be HV

→ Thus our plain text blocks IN DI AN AR MY becomes KG CK FT MU HV

→ To decrypt a message encrypted with the playfair cipher, the reverse process is used.

KG CK FT MU HV becomes IN DI AN AR MY

3) What do you mean by DES? Diagramatically illustrate the structure of DES & describe the steps in DES encryption process with example.

Ans. Input : 64 bits

Output : 64 bits

Main key : 64 bits

Sub key : 56 bits

Round key : 48 bits

No. of rounds : 16 rounds.

→ DES stands for Data Encryption Standard & the algorithm is referred to as the Data Encryption Algorithm (DEA)

→ In DES, data is encrypted in 64 bit blocks using 56 bit key.

DES encryption algorithm:-

Structure of DES
- Key ~~Scher~~ Schedule
- Round Function
- Initial & Final Permutation

① Plain text is broken into blocks of length 64 bits

② 64 bit block undergoes an initial permutation (IP) using IP table, IP(M)

③ 64 bit permuted input is divided into two 32 bit blocks.

$$\underbrace{\underset{(L_0)}{\text{Left}} \qquad \underset{(R_0)}{\text{Right}}}_{\text{Initial Blocks.}}$$

④ Here, 16 rounds are performed on both L and R blocks.

$L_n = R_{n-1}$

$R_n = L_{n-1}$ XOR(R_{n-1}, K_n)

⑤ The function F(·) represents the heart of the DES algorithm.

- Expansion:-

The right 32 bit half block is expanded to 48 bits using the expansion permutation (E) table

$E(R_{n-1})$

• Key Mixing:-

The expanded result is combined with a subkey using an XOR operation.

$$K_n + E(R_{n-1})$$

• Substitution :-

→ After mixing the subkeys, the block is divided into eight 6 bit pieces & fed into the substitution boxes (S boxes). Each 6 bit piece is used as an address in S boxes.

→ Here in S boxes, first & last bit is used ~~&~~ as address bit & middle four bits are used as main bits. Hence, S box is 4 bit length piece.

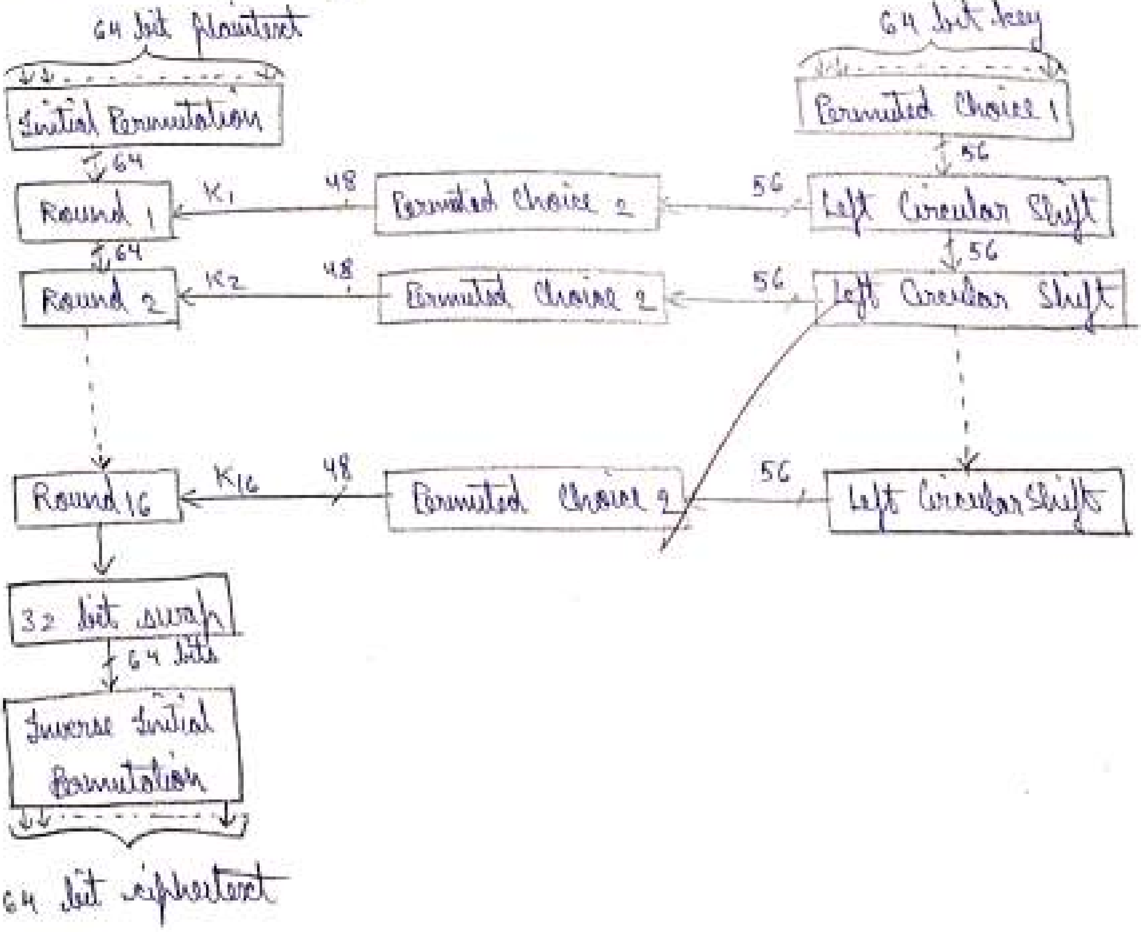
→ Now 8 S-boxes output is combined & 32 bit section is obtained.

$$S(K_n + E(R_{n-1})) = S_1(B_1)\, S_2(B_2)\, S_3(B_3)\, S_4(B_4)\, S_5(B_5)\, S_6(B_6)\, S_7(B_7)\, S_8(B_8)$$

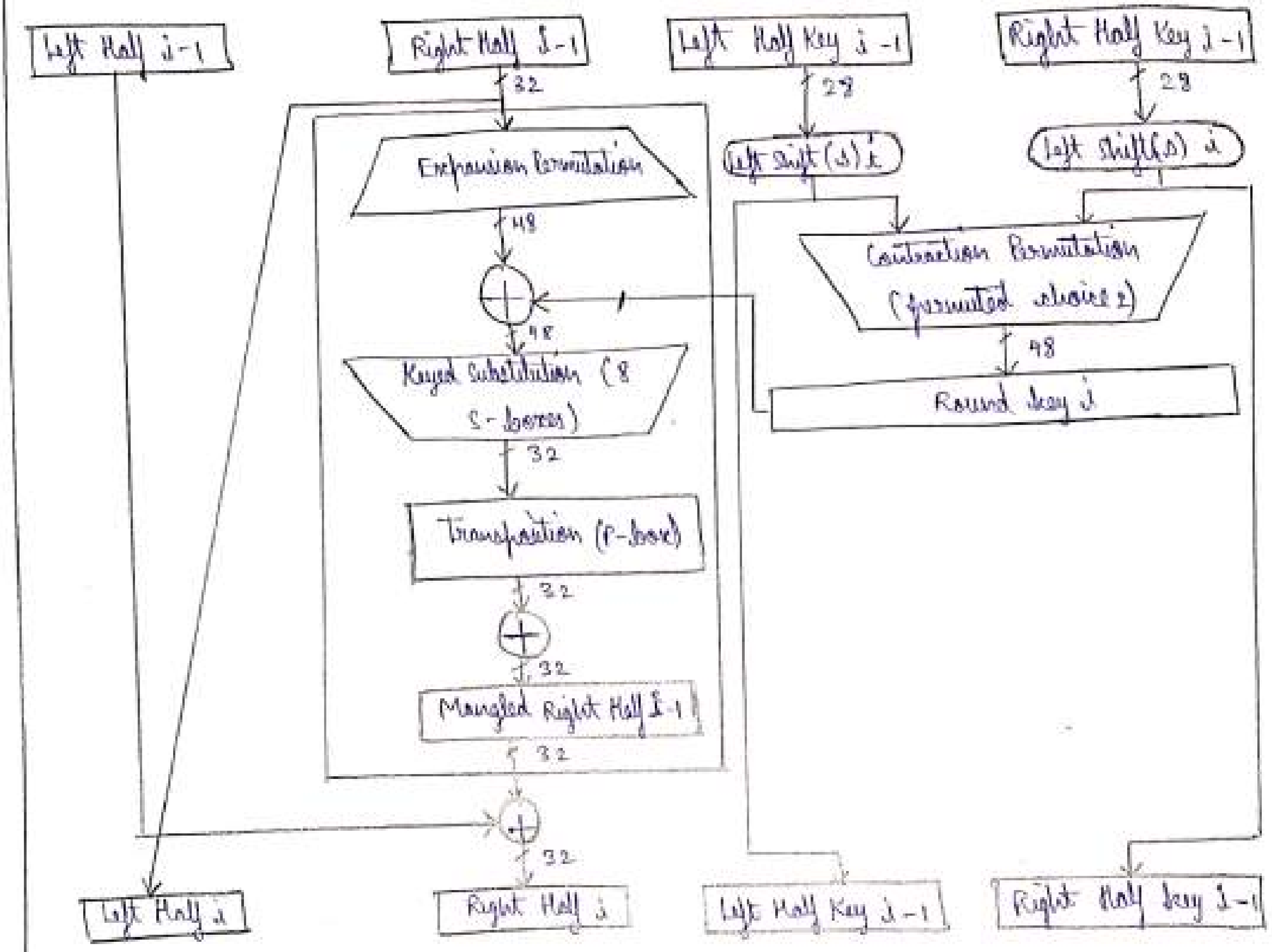
• Permutation:-

32 bits outputs ~~as~~ from S-boxes are rearranged using P-box

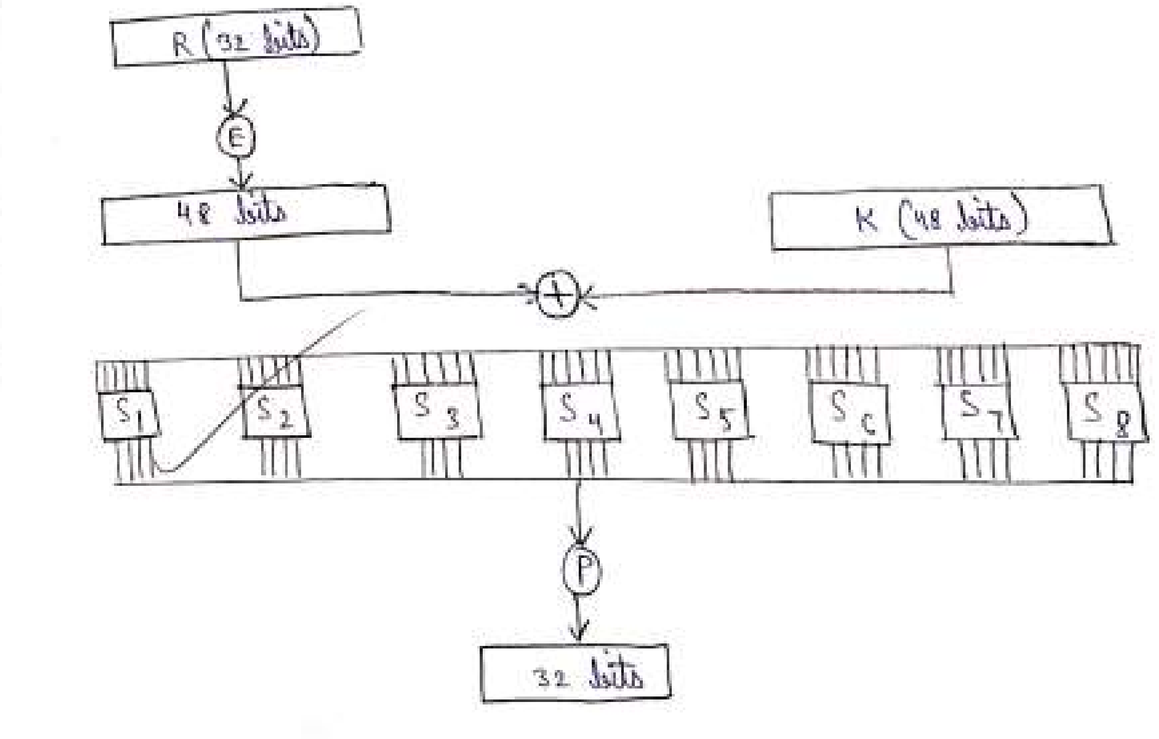
⑥ Now the results obtained after 16 rounds L_{16} & R_{16} are recombined into 64 bit and rearranged using an IP^{-1} Table.



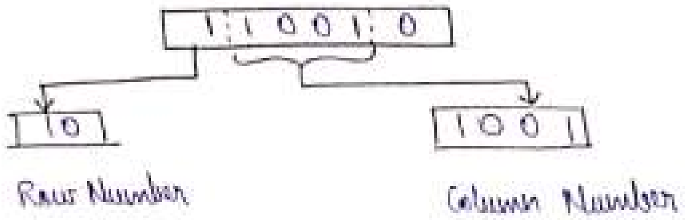
DES Round Structure:



Mangler Function:



S box



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

1100

Key Schedule (generator):-

This algorithm generates the subkeys (K→K₁, K₂,..,K₁₆)..

① The 56 bits of the key are selected from the initial 64 by Permuted Choice 1 (PC1) Table

② The 56 bits are divided into two 28 bit halves

③ In each round, both halves are rotated left by one or two bits (specific for each round)

④ The 48 subkey bits are selected by Permuted Choice 2 (PC2) table (24 bits from the left half, & 24 from the right) & used in each round.

4) Describe International Data Encryption (IDEA) Algorithm with its key generation, encryption & its applications in Cyber security world.

Ans → The International Data Encryption algorithm (IDEA) is perceived as one of the strongest cryptographic algorithms.

Rounds: 8
Input text: 64 bit
Key Size: 128 bits
No. of keys: 52
No. of data blocks: 4

→ IDEA is reversible like DES, that is the same algorithm is used for encryption & decryption.

→ IDEA is a block cipher algorithm which also works on 64 bit plain text blocks. The key is longer and consists of 128 bits.

→ 64 bit input plaintext block is divided into 4 portions of plain text P_1 to P_4

→ Thus P_1 to P_4 are the inputs to the first round of the algorithm. There are eight such rounds.

→ In each round, six subkeys are generated from the original key, each of the subkeys consists of 16 bits.

Rounds:

In IDEA there are 8 rounds. Each round involves a series of operations on the 4 data blocks using 6 keys.

In rounds add, multiply, XOR, modulo operations are required.

Step1:- Multiply P_1 and K_1

↓

Step2:- Add P_2 and K_2

↓

Step3:- Add P_3 and K_3

↓

Step4:- Multiply P_4 and K_4

↓

Step5:- XOR the results of S_1 and S_3

↓

Step6:- XOR the results of S_2 and S_4

↓

Step7:- Multiply the results of S_5 with K_5

↓

Step8:- Add the results of S_6 and S_7

↓

Step9:- Multiply the results of S_8 with K_6

↓

Step10:- Add the results of S_7 and S_9

↓

Step11:- XOR the results of S_1 and S_9

↓

Step12:- XOR the results of S_3 and S_3

↓

Step13:- XOR the results of S_2 and S_{10}

↓

Step14:- XOR the results of S_4 and S_{10}

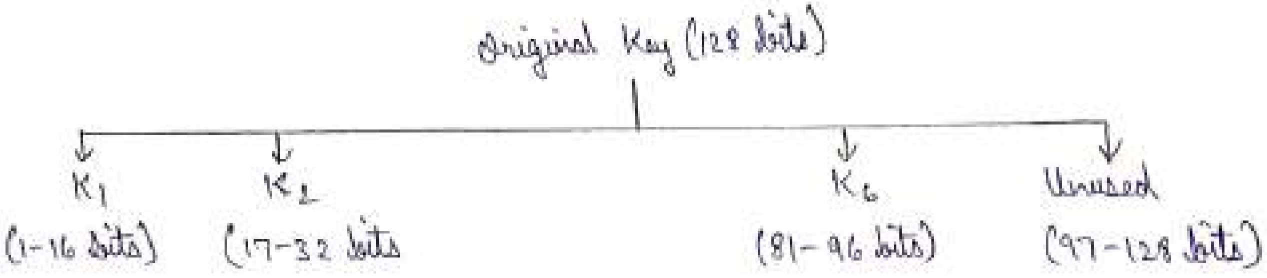
→ Input blocks are shown as P_1 and P_4, the subkeys are denoted by K_1 to K_6 and the output of this step is denoted by R_1 to R_4.

- **Subkey generation for a Round:-**

Each round uses 6 subkeys. So 8 rounds uses 48 subkeys and for final output transformation 4 subkeys are used. From 128 bits these 52 ~~&~~ subkeys

First Round:-

Here the 128 bit initial key is used to generate 6 subkeys which are K_1 to K_6



Second Round:-

In round 1, 1-96 bits are adequate to produce subkeys K_1 to K_6. For second round, we can utilize the unused 32 bits which would produce 2 more subkeys.

- **Output Transformation:-**

It is a one time operation. It takes place at the end of the 8th round. The input to output transformation is of course the output of 8th round.

Process of output Transformation

Step 1:- Multiply R_1 & K_1

↓

Step 2:- Add R_2 and K_2

↓

Step 3:- Add R_3 and K_3

↓

Step 4:- Multiply R_4 and K_4

- **Subkey generation for the output transformation :-**

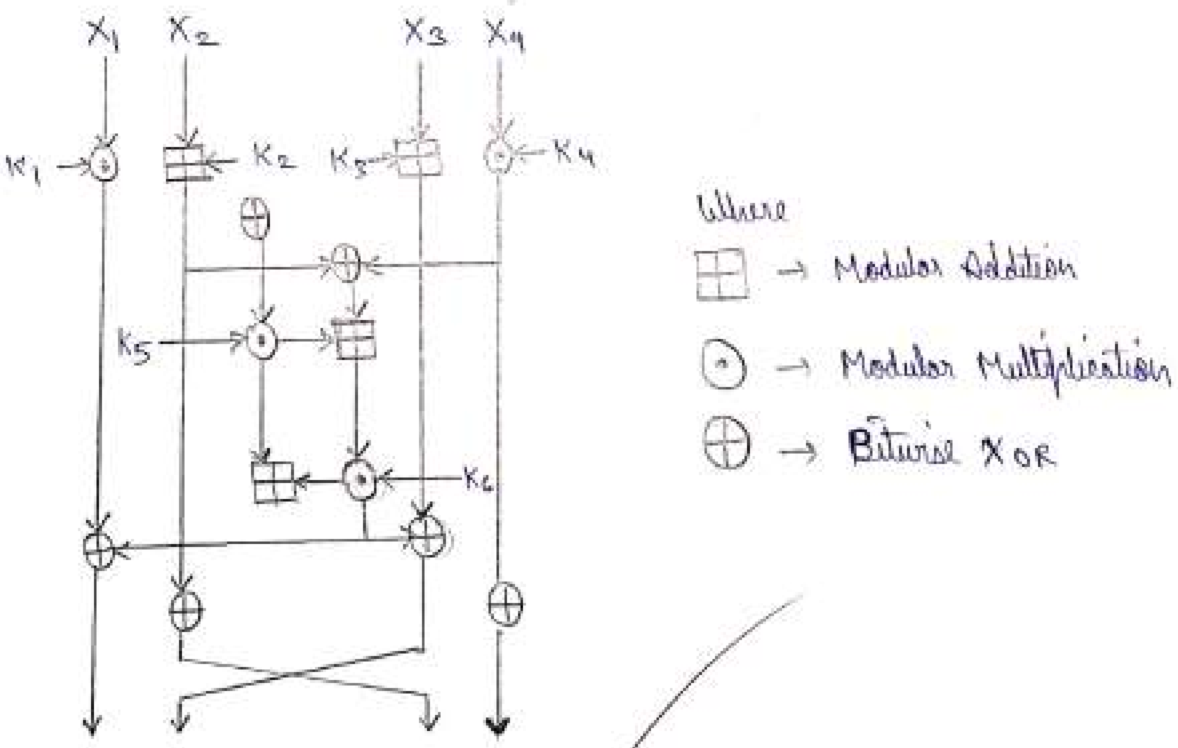
At the end of the 8th and final round, the key is exhausted & shifted. ∴ In this round, the first 64 bits make up subkeys K_1 to K_4 which are used as the 4 subkeys for this round.

- **IDEA Decryption :-**

The decryption process is exactly the same as the encryption process. There are some alterations in the generation & pattern of subkeys. The decryption subkeys are actually an inverse of the encryption subkeys.

- **Where is IDEA used :-**

IDEA is used in secure communications, financial transactions, electronic voting, file encryption & legacy systems.



5) Explain about Blowfish Algorithm with focus on Key Generation & Encryption & Decryption Process.

Ans. Blowfish was developed by Bruce Schneier, and has the reputation of being a very strong symmetric cryptographic algorithm. Blowfish was designed with the following objectives in mind:-

① Fast:- Blowfish encryption rate on 32 bit microprocessors is 26 clock cycles per byte

② Compact:- Blowfish can execute in less than 5KB memory.

③ Simple- Blowfish uses only primitive operations, such as addition, XOR & the table look-up, making its design & implementation simple.

④ Secure:- Blowfish has a variable key length up to maximum of 448 bits long, making it both flexible & secure.

→ Blowfish encrypts 64 bit blocks with a variable length key. It contains 2 parts as follows:-

① Subkey generation
② Data Encryption.

Rounds:- 18

Subkeys:- 18

Subkey size: 32 bit

Input Plain Text Size : 64 bits

Key size :- 32 - 448 bits.

• Subkey generation:-

(a) Initialize the P array first, followed by 4 S boxes with a fixed string

$P_1, ----, P_{18}$

(b) Perform bitwise XOR of P_1 with K_1 .... this is performed till P_{14} & K_{14}. The key array (K) is exhausted. Hence for P_{15} to P_{18} reuse K_1 to K_4.

$$P_1 = P_1 \text{ XOR } K_1$$
$$P_2 = P_2 \text{ XOR } K_2$$
$$\vdots$$
$$P_{15} = P_{15} \text{ XOR } K_1$$
$$P_{16} = P_{16} \text{ XOR } K_2$$
$$P_{17} = P_{17} \text{ XOR } K_3$$
$$P_{18} = P_{18} \text{ XOR } K_4$$

(c) Now take 64 bit block. Use P arrays and S boxes above to run the Blowfish encryption process on 64 bit all zero block. This step produces a 64 bit cipher text. Divide this into two 32 bit block values.

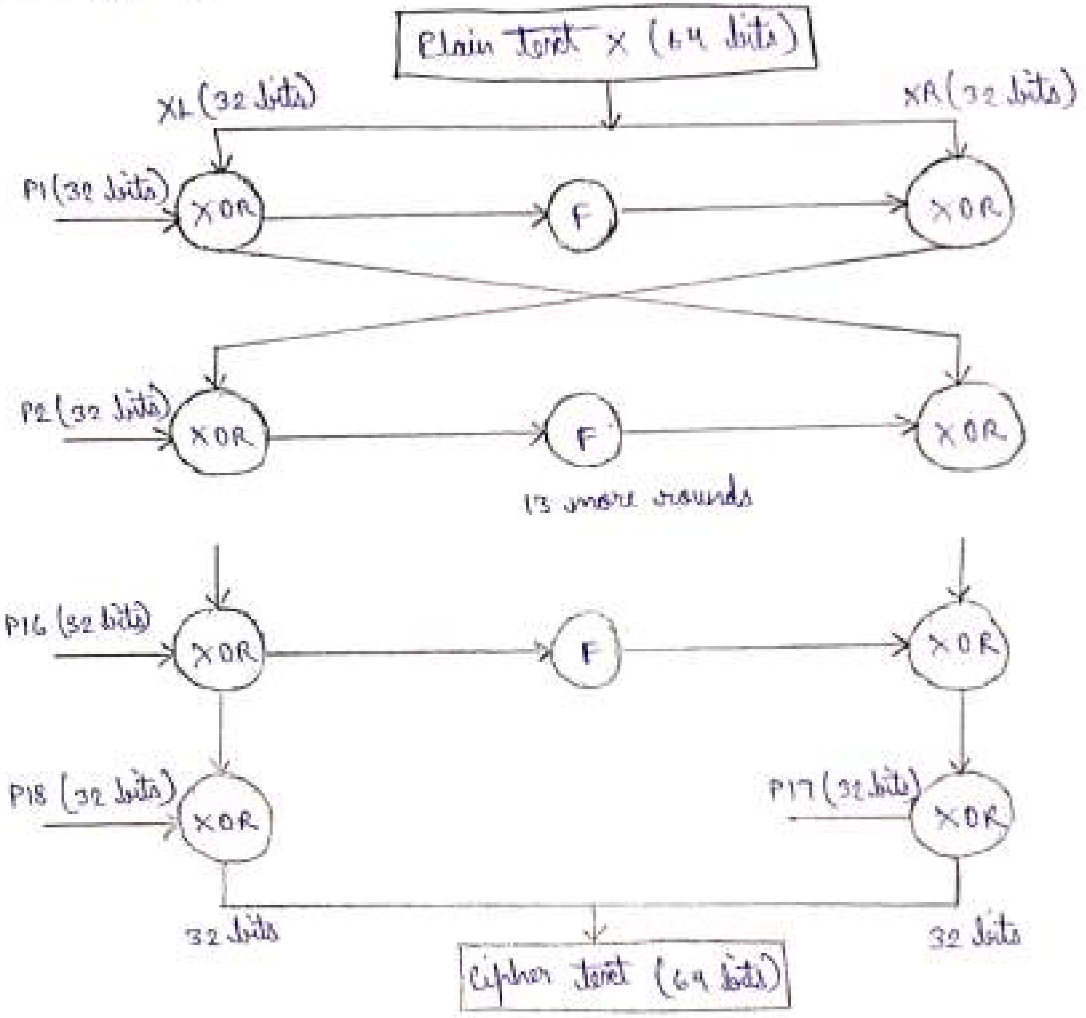
Data Encryption and Decryption:-

The encryption of a 64 bit plaintext input x is shown in an algorithmic fashion.

① Divide x into two blocks: xL and xR, of equal sizes. Thus, both xL and xR will consist of 32 bits each.

```
② For i=1 to 16
        XL = XL XOR Pi
        XR = F (XL) XOR XR
        Swap XL, XR
   Next i
③ Swap XL, XR (i.e undo last swap)
④ XL = XL XOR P18
⑤ Combine XL & XR back into x.
```

Blowfish encryption :-

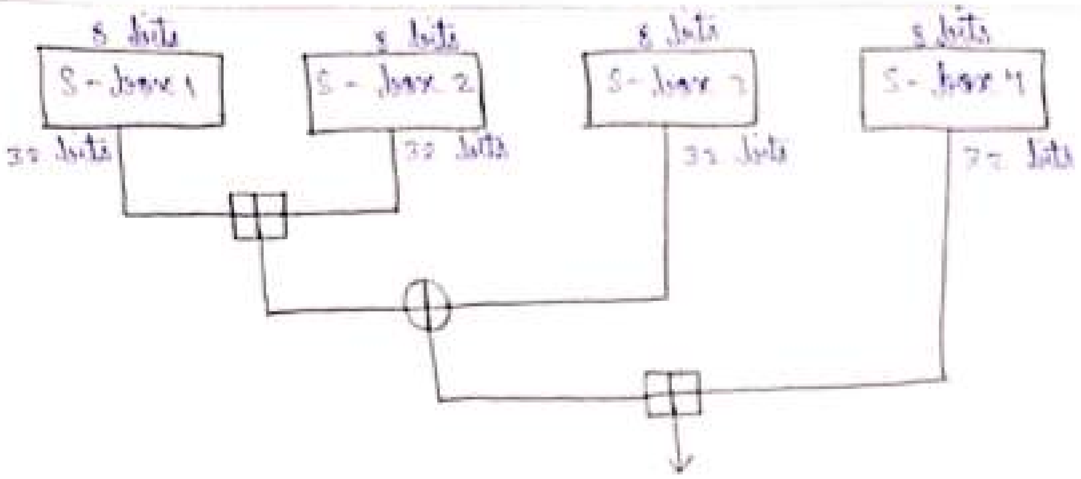


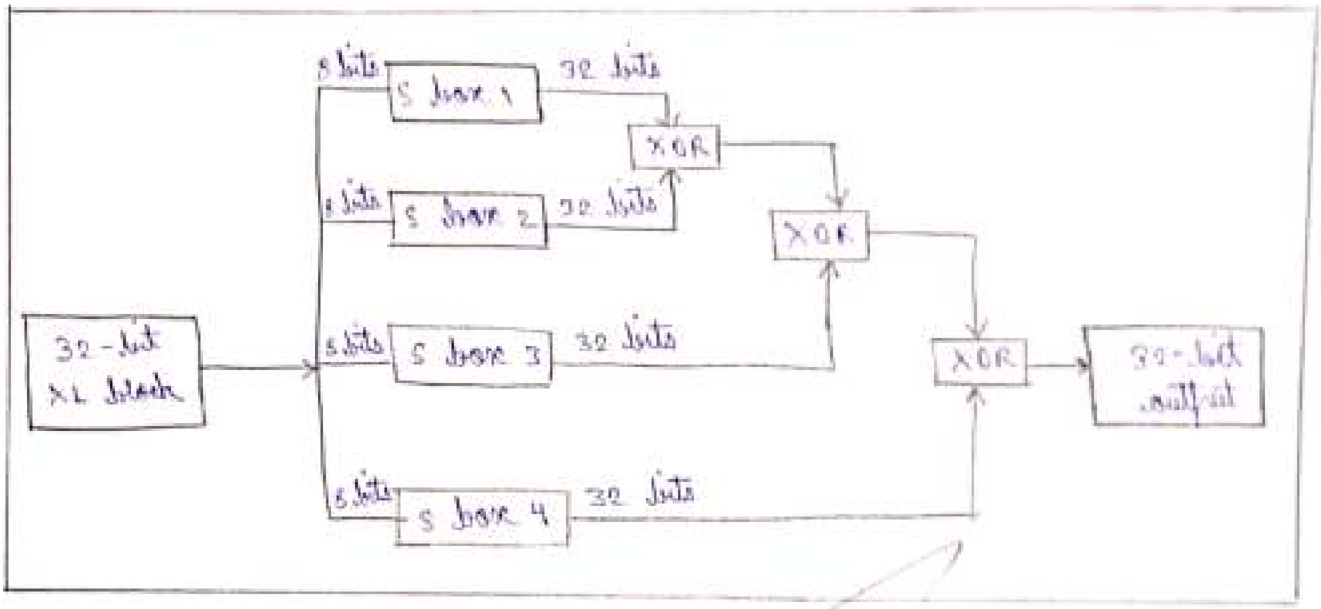
The function F is as follows:-

(a) Divide the 32 bit XL block into four 8 bit sub blocks, named a, b, c & d

(b) compute F[a,b,c,d] = ((S1,a + S2,b) XOR S3,c) + S4,d. For example, if a=10, b=95, c=37 & d=191, then the computation of F will be

F[a,b,c,d] = ((S1,10 + S2,95) XOR S3,37) + S4,191


8 bits
S - box 1
8 bits
S - box 2
8 bits
S - box 3
8 bits
S - box 4
32 bits
32 bits
32 bits
32 bits


32 - bit
XL block
8 bits
S box 1
32 bits
8 bits
S box 2
32 bits
8 bits
S box 3
32 bits
8 bits
S box 4
32 bits
XOR
XOR
XOR
32 - bit
output

13/2/25