Page No.
Date :

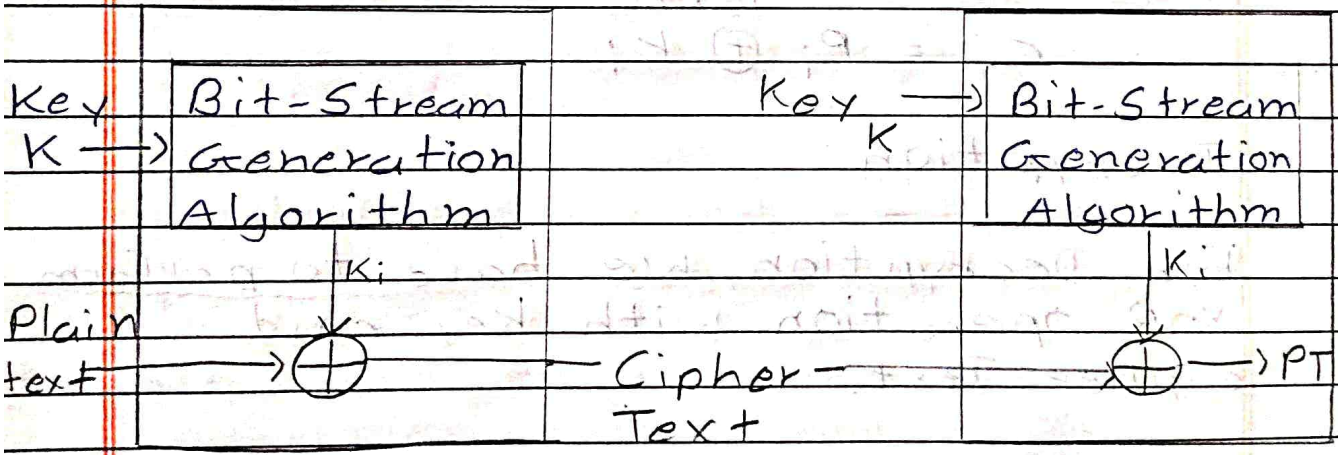
Symmetric and Asymmetric Cryptographic Techniques

* Explain Structure of Stream Cipher.

=> Stream Cipher is a type of Symmetric Cryptography.

In Stream Cipher, one byte is encrypted at a time while we have to perform encryption and decryption.

Stream Cipher encrypts a given plain text to the cipher text using a key.



Encryption

Decryption

In Stream Cipher, Input is given as One bit and we will get one bit output.

In Stream Cipher, Encryption and Decryption is perform bit by bit or byte by byte.

- Encryption :

For Encryption, we have to perform XOR operation with key and plain text.

Using XOR operation, we get the cipher text.

Cipher text = Plain Text $\oplus$ Key

$$C_i = P_i \oplus K_i$$

- Decryption :

For Decryption, we have to perform XOR operation with key and cipher Text.

Using XOR operation, we get the plain text.

Page No.
Date :

Plain Text = Cipher Text ⊕ Key

$P_i = C_i \oplus K_i$

* Explain Feistel Structure.

=> Feistel structure is also know as Block Cipher structure.

Feistel Cipher Structure is use Substitution and Permutation Model.

In Feistel cipher, Input Plain Text block of length is 64-bits.

There are 16 Rounds are perform for encryption.

All the rounds have the same structure for encryption and decryption.

In Feistel Cipher Structure, we have to perform XOR operation with Left-Half of the plain text.

=> Encryption:

In Encryption, The 64-bits Input Plaintext is divided into the two part.

Two Parts → L.H.S. → LE_0 → 32 bits
→ R.H.S → RE_0 → 32 bits.

In this Process, we have to use one Function and Key for the encryption.

-> Process of Round 0 to 16:

For every Round, RE_0 is pass to the Function F which contain the Key.

After that, we have to perform XOR operation with LE_0 and Previous value.

In the Next Round, RE_0 value become the LE_1.

This Process is continues until the Round 16.

$RE_0 = LE_1$

$RE_1 = F(RE_0, K_1) \oplus LE_0$

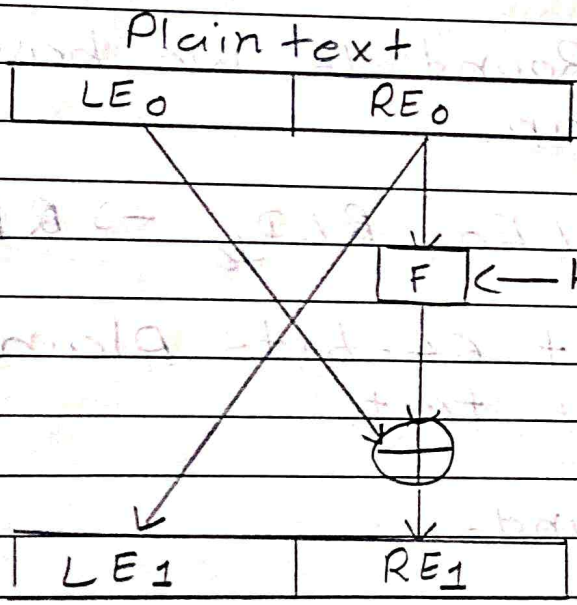
Teacher's Signature

After the Round-16, we have to swap the two part value.

Here, $RE_{16} \rightarrow LE_{17}$ and
$LE_{16} \rightarrow RE_{17}$.

Here, LE_{17} and RE_{17} is become Cipher text.

=> One Round Diagram:



For every, Round This Process is perform For encryption.

=> Decryption:

For Decryption, this same process is used to get Cipher text to Plain text.

But on this side,

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

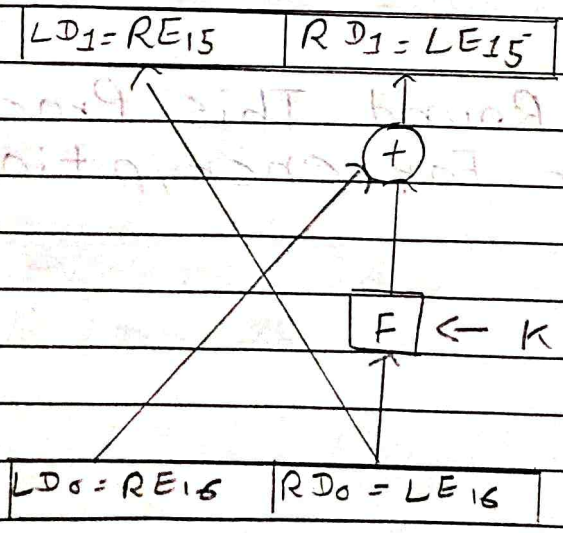
$$RD_1 = RE_{16} \oplus F(RE_{15}, K_{16})$$

After the Round 16, we have to perform swap.

So, $RD_{16} \rightarrow LE_0$, $LD_{16} \rightarrow RE_0$

Here, we get 64-bits plain text using Cipher text.

-> For One Round:



* Explain DES encryption and decryption method.

=> DES stands For Data Encryption Standard which is type of Symmetric cryptography.

In DES, Same key is used for performing encryption and decryption.

There are 64-bits Input block and 64-bit key is used for encryption.

There are 16 Round is used for encryption.

=> Step of DES Encryption.

1 64-bits Plain text is given to the Input and perform initial permutation and gives input in round.

2 In every Round 64 bits are given as Input and Output of every round is also going to be 64 bits until 16 rounds.

3 After completed 16th round with 64-bits output is given to the 32-bits Swap Function.

Do Partition and swap it.

64 bits → 32-bits → L.H.S.
→ 32-bits → R.H.S.

4 After Swap, Again we will get 64-bits output.

After that, we have to perform Inverse Initial Permutation which is Final cipher text.

=> Single Round Process In DES.

Single Round Algorithm:

1 Key Transformation: Permutation of selection of Sub-key From Original Key.

2 Expansion of Permutation (E-Table): Right half is expanded From 32-bits to 48-bits.

3 S-Box Substitution: Accepts 48-bits From XOR operation and produce 32-bits.

4 P-Box Permutation

5 XOR and Swap.

=> Key Transformation:

After Initial Permutation, 64 bits input is divided into two part.

64 bits -> 32-bits -> L.H.S.
64 bits -> 32-bits -> R.H.S.

This 32 bits are going to the Expansion Table.

=> E - Table (Expansion Permutation):

This 32-bits are are expand to the 48-bits using the E-Table.

E - Table:

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

This 48-bits, is used to perform the XOR operation.

=> S-Box (Substitution Box):

S-Box is used to convert 48-bits to 32-bits, after the performing the XOR operation.

For Produce 32 bits output, we have to use 8 S-Box.

In every S-Box, we have to give 6-bits input and we

will get 4-bits output in one S-Box.

In S-Box, 6-bits is convert into the 4-bits.

=> P-Box: After the convert into 32-bits, we have to change the position of every bits.

13	1	16	3	6	9	10	12
19	5	2	21	18	24	25	28
20	15	4	17	8	31	32	11
14	22	26	7	23	27	30	29

=> XOR and Swap:

After the Permutation, we have to perform XOR operation and perform swap operation.

=> Key Scheduling Process in DES.

For every, Round, we have to schedule the key.

Steps: Permuted Choice: 1
Left Circular Shift
Permuted Choice: 2

-> Permuted Choice 1 :

In this step, 64 - bits are convert into the 56 - bits.

8 - bits are dropped to get 56 - bits (8, 16, 24, 32, 40, 56, 64) bits.

This 56 - bits are divided into two parts -> 28 - bits -> C_0
-> 28 - bits -> D_0

-> Left Circular shift : In this step. We have to perform shifting operation.

1 shift for i = 1, 2, 9, 16 Rounds

2 shift for i = other Rounds.

-> Permuted Choice - 2

In this step, 56 bits are convert into the 48 - bits.

We have to dropped the 8 - bits randomly.

After that, we get 48 - bits key.

=> Intial Permutation: We have to change 64-bits position.

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	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	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
64	56	48	40	32	24	16	8

=> Inverse Intial Permutation:

After the 16th Round, we have to change the 64-bits position.

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

=> Decryption in DES:

It is similar to the encryption, the only difference is the subkeys are applied in the reverse.

The data goes through the initial permutation Input and the block is divided into two part.

The right half is carried through the Function F, but 16 th subkey is registered with this part.

The Left side of block is XOR once the output through the F Function is received.