Advanced ARM Architectures

Project Report

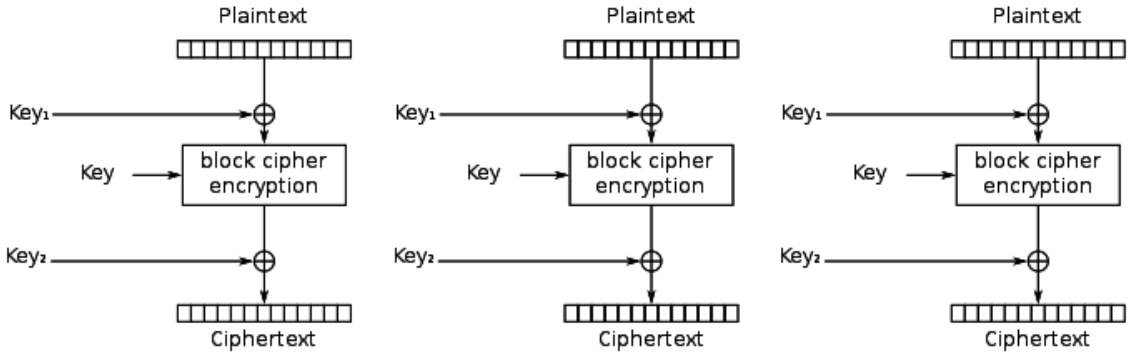
Name:

Goli Akshay Sujith(IMT2017507)

DES Algorithm

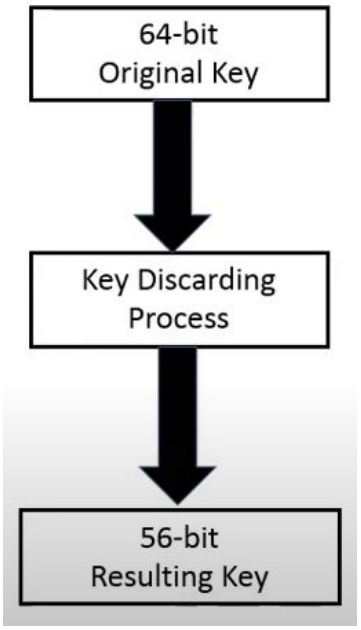
Introduction:

- Developed in early 1970's at IBM and submitted to NBS
- DES is a landmark in cryptographic algorithms.
- DES works based on Feistel Cipher Structure.
- DES is symmetric cipher algorithm and use block cipher method for encryption and decryption



Xor Encrypt Xor (XEX) mode encryption

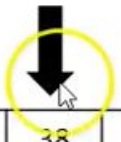
Key discarding Process.



If the key is 64 bit, it is converted to a 56 bit resulting key by removing the every 8th bit of original key. This process is known as the key discarding process.

For Example,

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

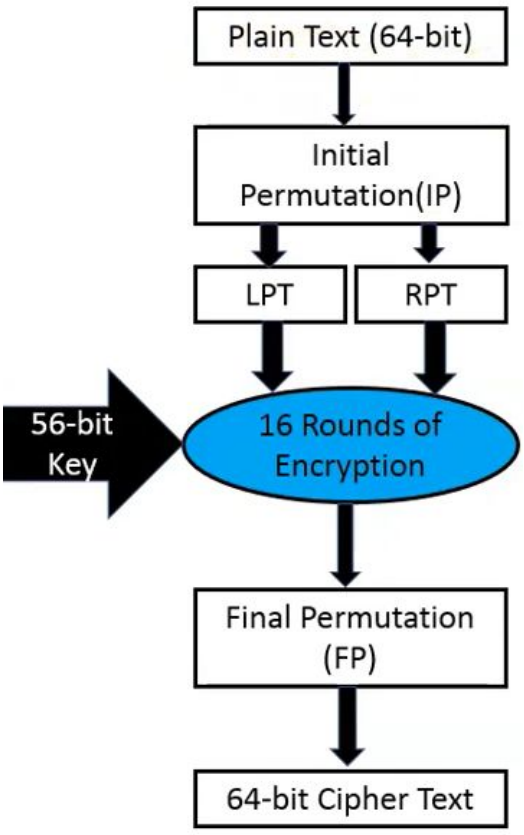


1	2	21	38	58	15	37
22	55	44	3	53	27	11
49	28	14	42	61	48	63
18	39	56	10	64	16	62
45	40	20	54	4	33	34
7	30	47	59	32	5	35
29	12	13	6	24	46	57
17	23	50	31	43	51	9

From the above table we can see that every 8th bit is removed and a 56-bit resulting key is generated.

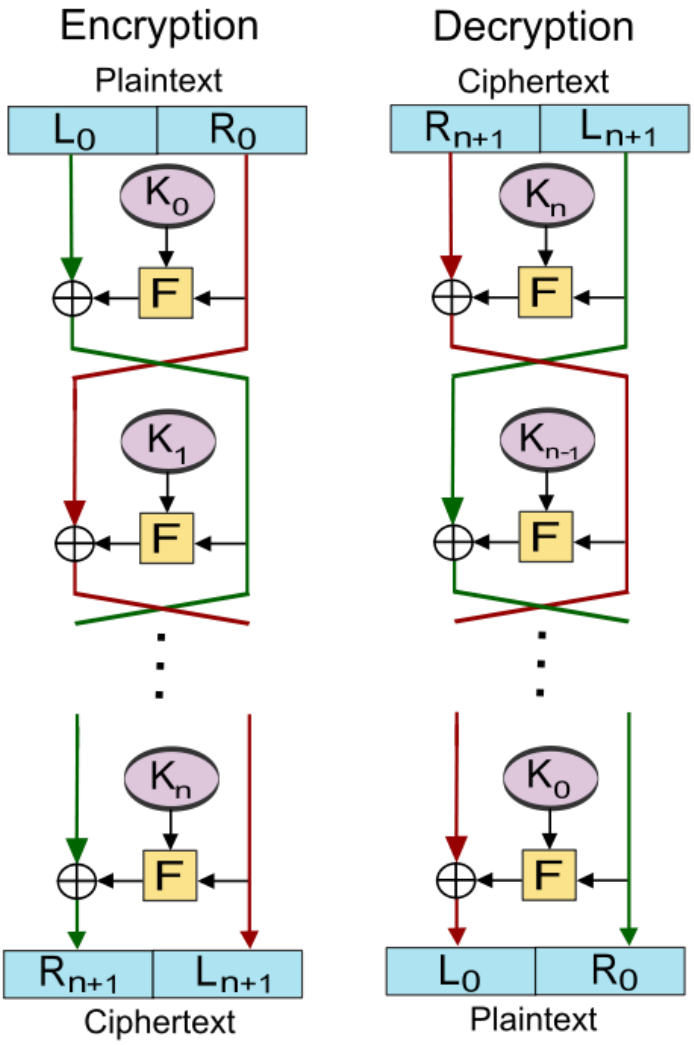
Steps of DES (Data Encryption Standard)

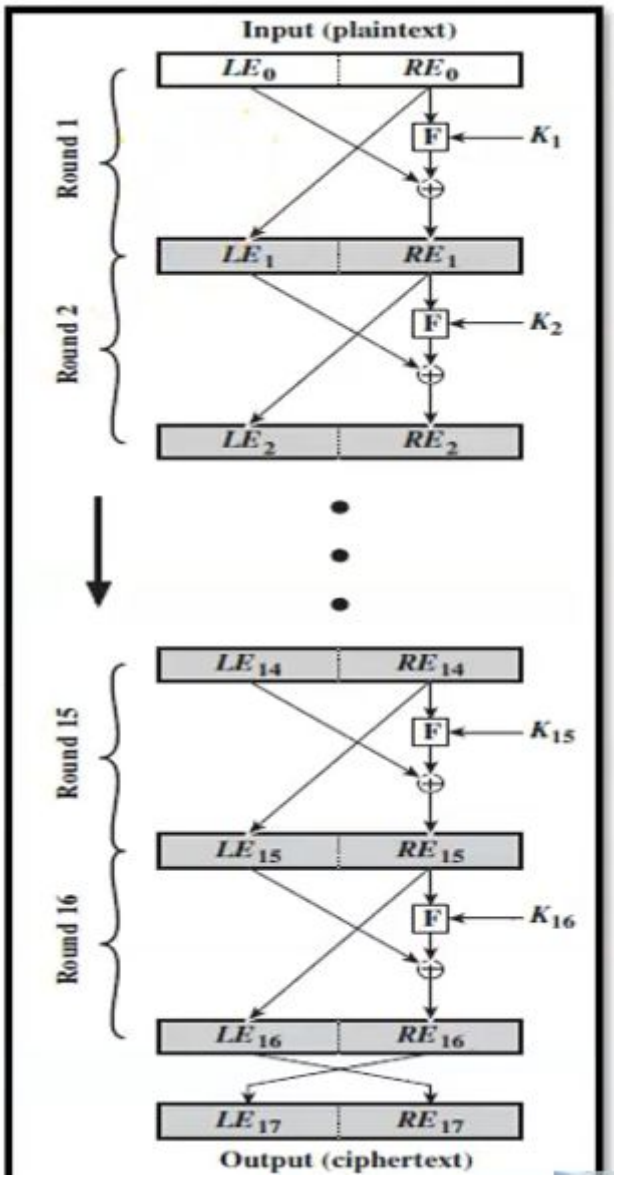
1. 64 bit plain text block is given to Initial permutation (IP) Function
2. IP is performed on 64 bit plain text block
3. After IP, it produces two halves of the permuted block known as left Plain text (LPT) and Right Plain Text (RPT)
4. Once LPT and RPT is generated, each LPT and RPT perform 16-rounds of encryption. The input for 16 rounds of encryption is LPT, RPT and 56-bit Key.
5. After the above process, LPT and RPT is rejoined and final permutation (FP) is performed on the combined block.
6. FInally, a 64-bit Cipher block is generated.



(FIG-1)

Feistel Cipher Structure to perform 16 rounds of Encryption





Initial Permutation (IP) and generation of LPT-RPT

- In the DES process, IP is performed only once (fig-1). The 64 bit text is given to the IP table.
- Bit sequence will be changed as per the IP Table

For Example,
As per the table,

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	63	55	47	39	31	23	15	7

The 58th bit will take first position.
The 1st bit will take 40th position and so on.

Permutation is all about arranging the bits.

Now from figure 1, The output of IP is divided into two equal halves as LPT, RPT (Each of 32 bit)
64 bit (IP) -> 32 bit (LPT) + 32 bit (RPT)

16 Rounds of Encryption function:

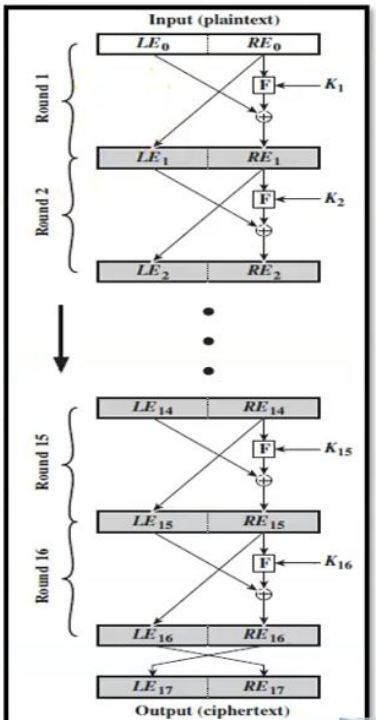
1. Key Transformation (56-bit key)
    a. Key Bit shifted per round (Left shift of the key bits)
    b. Compression permutation
2. Expansion permutation of Plain Text and XOR (Plain text size is 48 bit and Cipher Text size is 48 bit). Here the 32 bit LPT and 32 bit RPT converts into 48 bits because the key in the above is compressed and generated
3. S - Box (Substitution): The 48 bits in the above step needs to be converted to 32 bit back again
4. P - Box (Permutation): Transposition is done. (changing the positions of bits as explained before)
5. XOR and Swap

Key Transformation:

- 56 bit is divided into 2 halves each of 28-bits
- Circular left shift is performed on each half
- Shifting of bit position depends on the round
  For example:

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

For 1,2,9,16 rounds are done by 1 and others by 2.



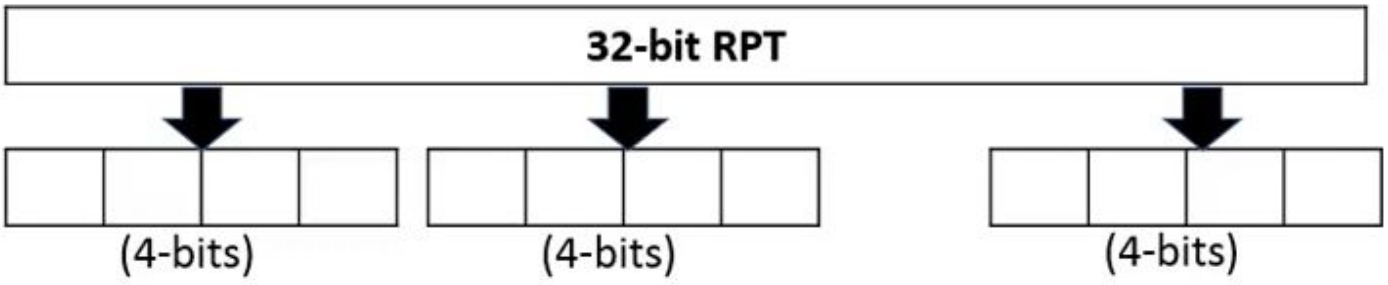
Compression permutation

- 56-bit input with bit shifting position
- Generates 48-bit key (Compression of key bit unit)
- Dropping 9, 18, 22, 25, 35, 38, 43, and 54th bits

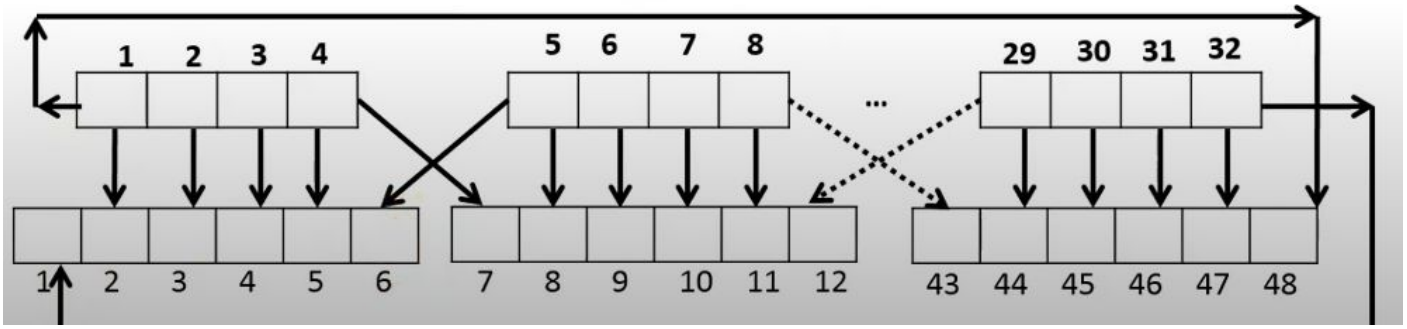
14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Expansion Permutation

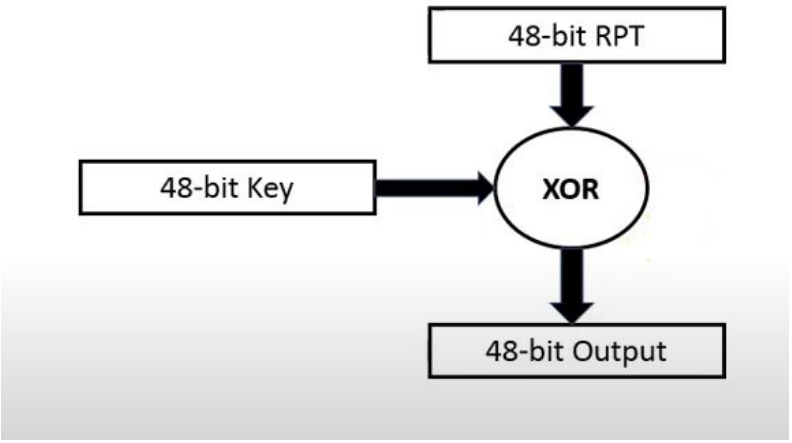
- 32 bit RPT of IP is expanded to 48 bits
- Expansion permutation steps:
    - 32bit RPT is divided into 8 block of 4 bits each



- Each 4-bit block is expanded to 6bit and produce 48-bit output

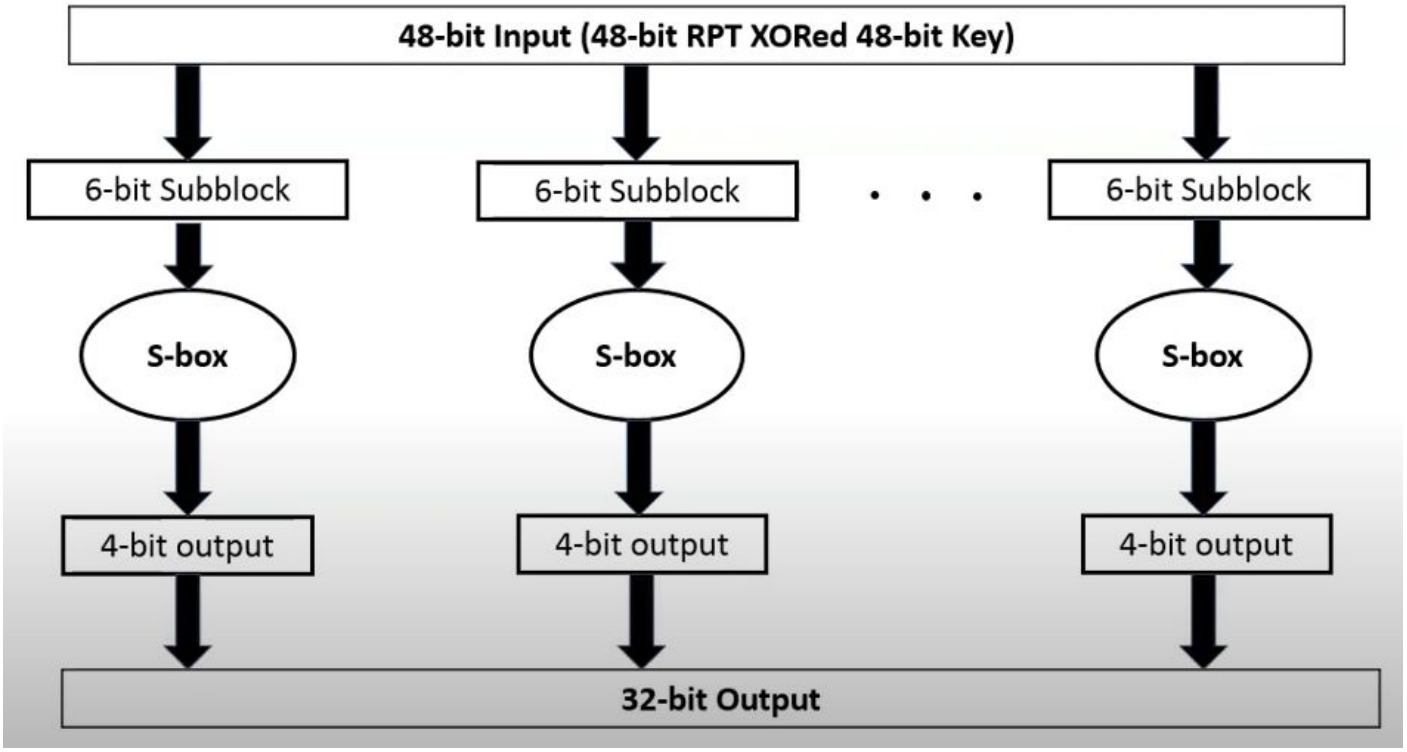


This is how we make it to 48 bits
Then we'll perform,



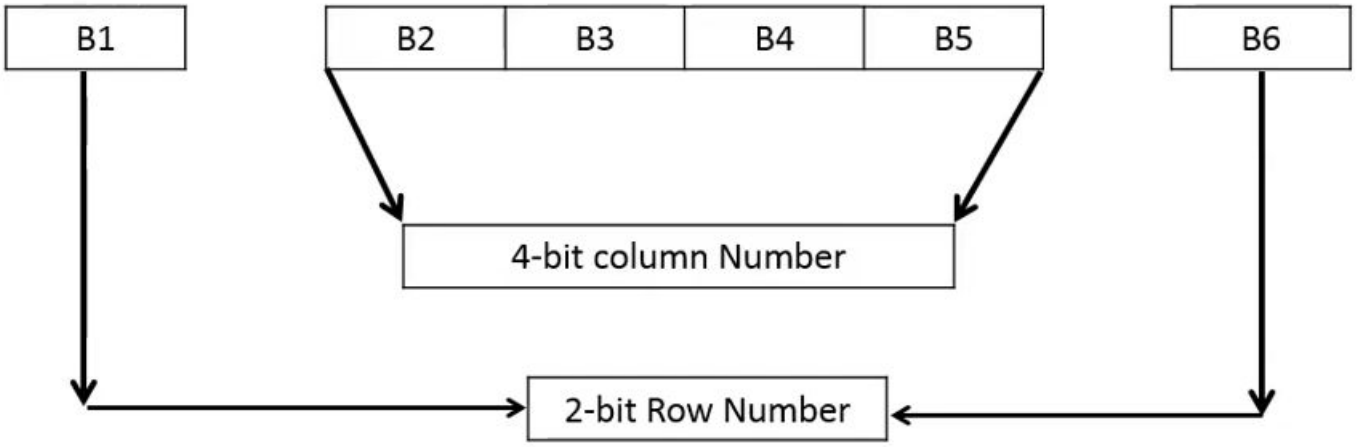
S- box

- Divided into 6-bits 8 sub blocks



Working of S-BOX:

The block is mapped to the S block table



S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Example: This is the block for 01 row and 1101 column.

The Output of S-Box is 32 bit

P-Box (Permutation):

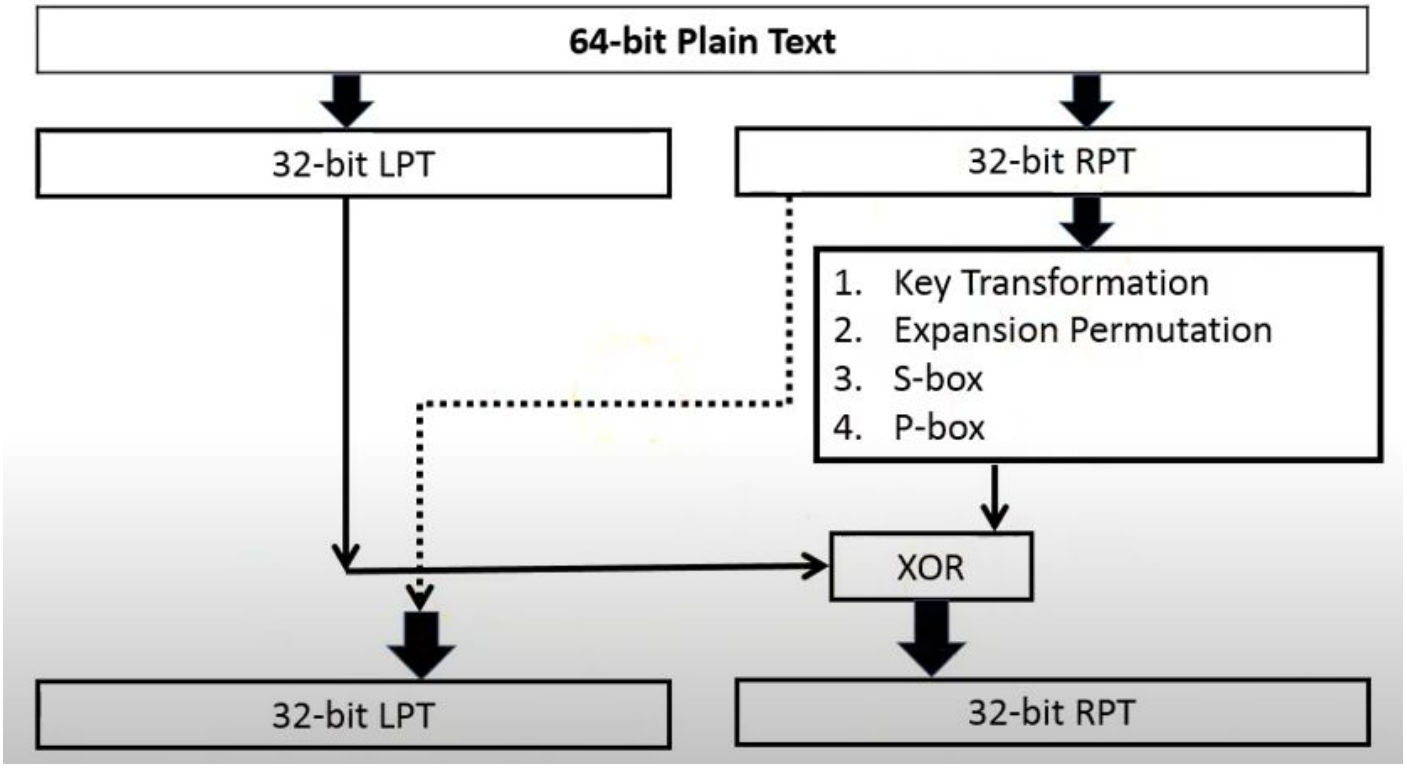
- The output of S-BoX is given to p-box and mapping is done accordingly with the P-Box table

32 bit is permuted with 16*2 permutable table

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

Example: 16th bit of s-box takes 1st position as per the permutation table.

XOR and SWAP:



This is how 16 rounds are done again and again.
After 16 rounds

Final Permutation:

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

Example: 40th bit takes the 1st position as per the table. And the Output is 64-bit Cipher text.

Calculations:

[spreadsheet bit wise calculation]
https://docs.google.com/spreadsheets/d/1EOR-uJIIiSYX4CZ5Y3eA8AXsxwQYziRHkdThYYM2CqU/edit#gid=827703666

Keywords:

CIPHER_TEXT -> Defining Cipher text
Key -> Defining Key
MODEENC -> Defining key for encryption and decryption
SBOX -> Defining S box table
__F1INITPERMRES -> Permutation function
__F4INVPERMUTATION -> Inverse permutation
__F3ROUND -> Rounding function

Test Cases:
Input Key -> (Hex): 9474b8e8c73bca7d
Input text -> (Hex): 9474b8e8c73bca7d
Binary form ->
1001010001110100101110001110100011000111001110111100101001111101

Cipher encryption text-> 8da744e0c94e5e17