Exp No: 4

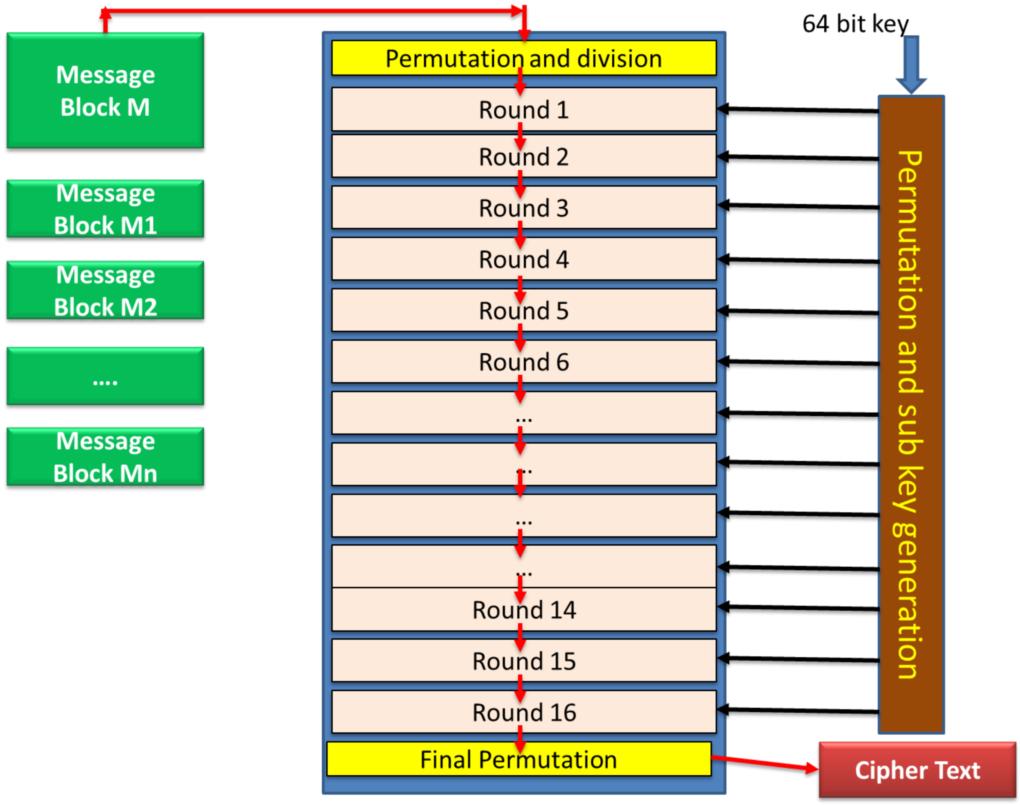
DES algorithm

Aim

To implement DES encryption and decryption

Description to Implement

The general structure of the DES consists of key schedule, round function, Initial and final permutation.



Step1: Plaintext is broken into blocks of length 64 bits.

Step2: The 64-bit block undergoes an initial permutation (IP) using initial permutation IP table, IP(M).

Step3: The 64-bit permuted input is divided into two 32-bit blocks: left (L) and right (R). The initial values of the left and right blocks are denoted L_0 and R_0.

Step4: There are 16 rounds of operations on the L and R blocks. During each round, the following formula is applied:

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

Step5: The function F(.) represents the heart of the DES algorithm. This function implements the following operations:

1. Expansion 2. Key mixing 3. Substitution 4. Permutation

1-Expansion: The right 32-bit half-block is expanded to 48 bits using the expansion permutation (E) table, $E(R_{n-1})$.

2-Key mixing: The expanded result is combined with a subkey using an XOR operation. Sixteen 48-bit subkeys (one for each round) are derived from the main key using the key schedule, $K_n + E(R_{n-1})$.

3-Substitution: After mixing in the subkeys, the block is divided into eight 6-bit pieces and fed into the substitution boxes (S-boxes), which implements nonlinear transformation. Each 6-bit piece uses as an address in the S-boxes where the first and last bits are used to address the i^{th} row and the middle four bits to address the j^{th} column in the S-boxes. The output of each S-box is 4-bit length piece. The output of all eight S-boxes is then combined into 32 bit section.

$K_n + E(R_{n-1}) = B_1B_2B_3B_4B_5B_6B_7B_8$

$S(Kn+E(Rn-1))=S1(B_1)S2(B_2)S3(B_3)S4(B_4)S5(B_5)S6(B_6)S7(B_7)S8(B_8)$

4-Permutation: The 32 bits outputs from the S-boxes are rearranged using the P-box, $F=P(S(K_n + E(R_{n-1})))$

Step6: The results from the final DES round (i.e., L_{16} and R_{16}) are recombined into a 64-bit value and rearranged using an inverse initial permutation (IP^{-1}) table. The output from IP^{-1} is the 64-bit ciphertext block.

PC1 Table

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

Schedule of left shifts

Iter	No. of left shifts
1	1
2	1

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

PC2 Table

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

IP 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

E-bit selection 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

S1-Box

S1

Column Number

Row No.	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

Final stage of permutation 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

Table IP^{-1}

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