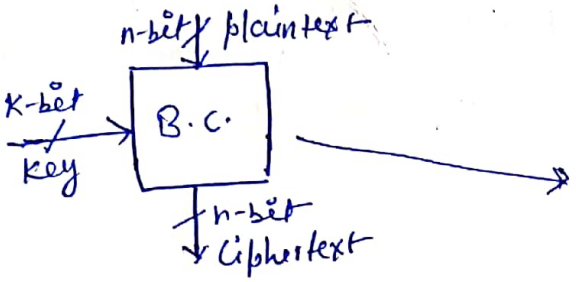
Block Cipher

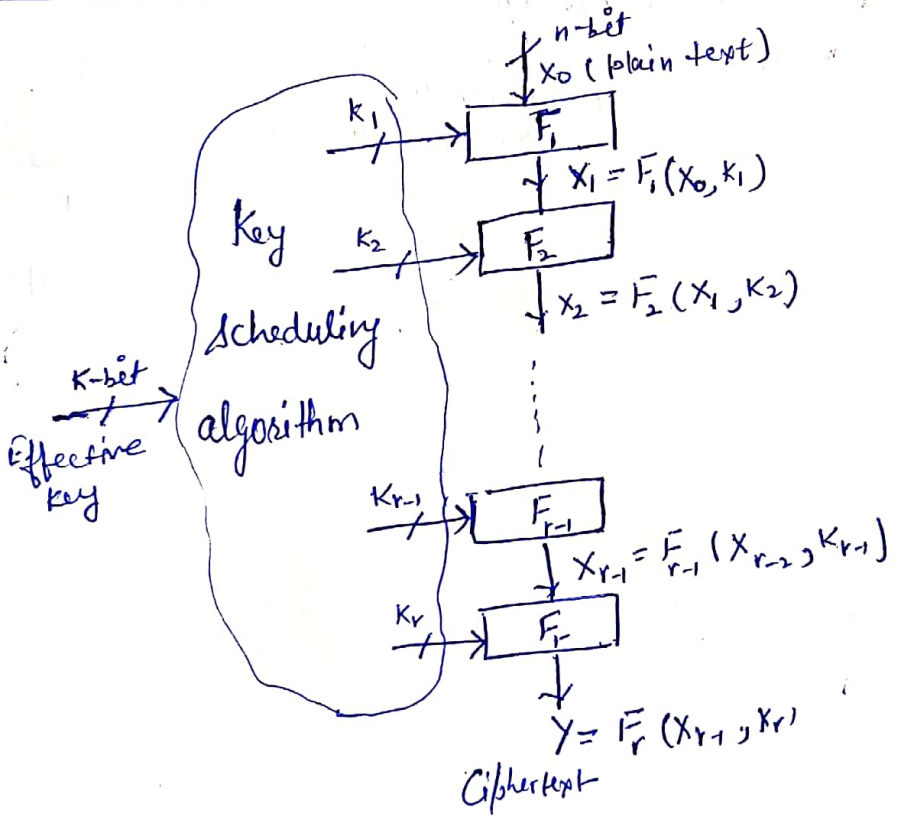
General structure of Block Cipher (B.C.)



Block cipher is a fn.

$E: \{0,1\}^n \times \{0,1\}^K \longrightarrow \{0,1\}^n$

r-round n-bit block cipher : There are r-round inside it



$F_1, F_2, \ldots, F_r$ are r-block ciphers

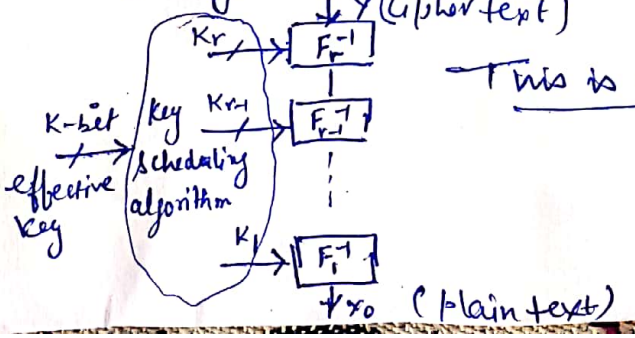
This figure shows the encryption scheme of r-round n-bit block cipher.

Here $K_1, K_2, \ldots, K_r$ are round keys which are generated by some key scheduling algorithm

The size of effective key is K-bit.

Here X_0 is plaintext of size n-bit and Y is the ciphertext of size n-bit.

The decryption of block cipher of r-round n-bit is same as encryption but in reverse order.



This is the decryption of r-round n-bit block cipher

Example of block cipher

(1) Substitution permutation network (SPN)

(2) Data encryption standard (DES) : It is 16-round 64-bit blocks cipher and size of effective key is 56-bit.

Here $r = 16$
$n = 64$
$K = 56$

(3) Advanced encryption ~~scheme~~ standard (AES)

$n = 128$-bit (plain text size)

$r = 10/12/14$ there are three version of AES based on rounds.

- 10-rounds 128 bit
- 12-rounds 128 bit
- 14-rounds 128 bit

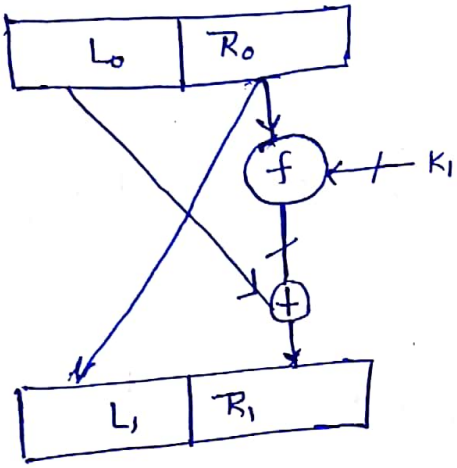
} depends on key size

$K =$ 128 bit
192 bit
256 bit
respectively

AES is

Data encryption standard (DES)

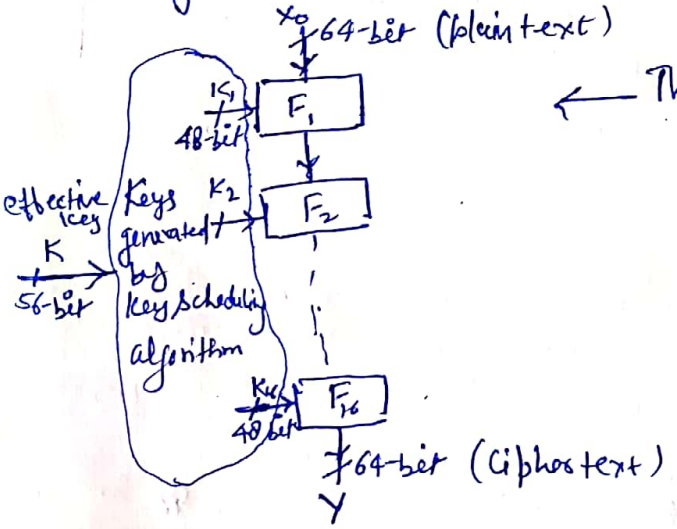
DES is based on Feistel structure. The Feistel structure is shown below



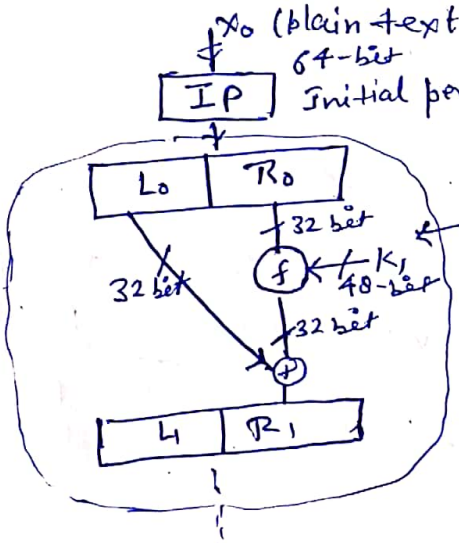
$$L_1 = R_0$$
$$R_1 = f(R_0, K_1) \oplus L_0$$

In Feistel Cipher the plaintext is divided into two half parts. Left part is denoted by L_0 and right part of plaintext is denoted by R_0. f is some fn. whose input is R_0 and key K_1 then output of f is XOR with L_0. The cipher text is $L_1 R_1$ where L_1 is left half part and R_1 is right part of ciphertext

- DES is 16-round, 64-bit block cipher and size of effective key is 56-bit and size of each round key is 48-bit



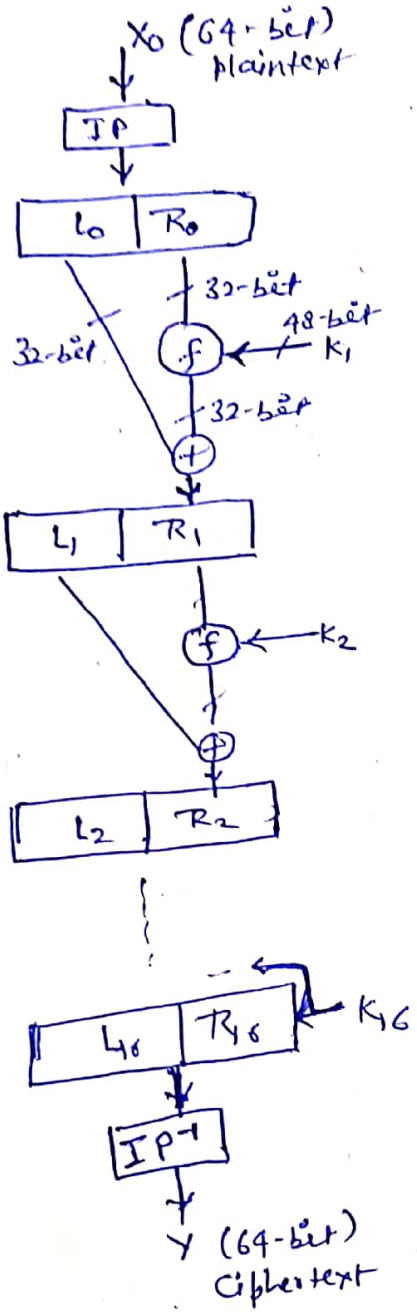
← This is DES cipher in which DES rounds $F_1, F_2, \ldots, F_{16}$ are Feistel structures.



← This is F_1 (one round) where f is some fn. described in next page

This is one round of encryption Total 16 rounds

So, the DES encryption is shown below



Here IP (initial permutation) is a permutation of 64-bits this permutation is fixed and given by the following table

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
63	55	47	39	31	23	15	7

The inverse of this permutation IP^{-1} which is used in last step is given by following 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

If $x_1 x_2 x_3 \ldots x_{64}$ is 64-bit plaintext then out of IP is below

$x_1 x_2 x_3 \ldots x_{64}$ → IP → $x_{16} x_{50} x_{42} \ldots x_7$

→ **f-function in DES** : The f-function in each round of DES takes two inputs one 32-bit right half of text and other is 48-bit round key

The components of f-function is

→ Expansion function — E (which is fixed given by a table)

→ S-boxes — there are 8, S-boxes $S_1, S_2, \ldots, S_8$ in each round these 8 S-boxes are called Vectorial Boolean function

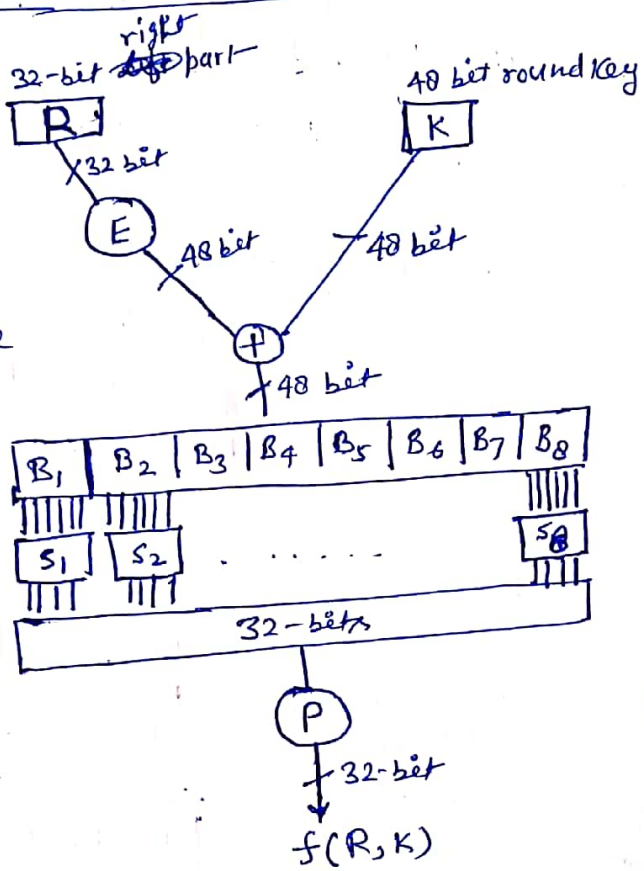
∴ $S_i : \{0,1\}^6 \longrightarrow \{0,1\}^4$ $i = 1, 2, \ldots, 8$

S_i are vectorial Boolean fn. where input is 6-bit and output is 4-bit

S-box is given in table form

and → permutation fn. P is fixed given in table form

f - function of DES is shown below



Here E is expansion fn. given by a table

B_i 6-bit

$S_1, S_2, \ldots, S_8$ are 8 S-boxes

$S_i : \{0,1\}^6 \longrightarrow \{0,1\}^4$

$i = 1, 2, \ldots, 8$

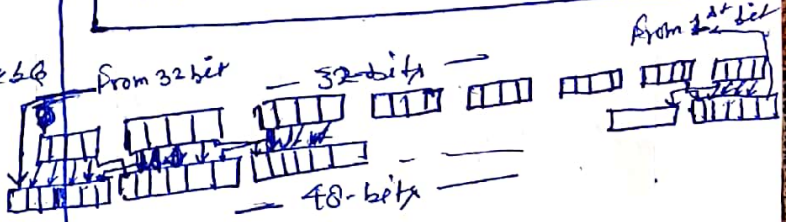
Here P is a permutation of 32-bits given by a table.

The expansion fn. E is given by

E					
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 expands 32-bits into 48-bits

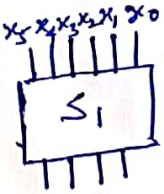
This table can be shown by this figure



The permutation fn. P is given by the following table

P							
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

S box S_1



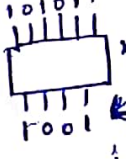
S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	04	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

$x_5 x_0$ ← give us row number 0 to 3

$x_4 x_3 x_2 x_1$ ← gives us column number 0 to 15

Output of S-box is a no. which is intersection of row & Column & then convert in binary

For exm



$x_5 x_0 = 11 \leftarrow 3$

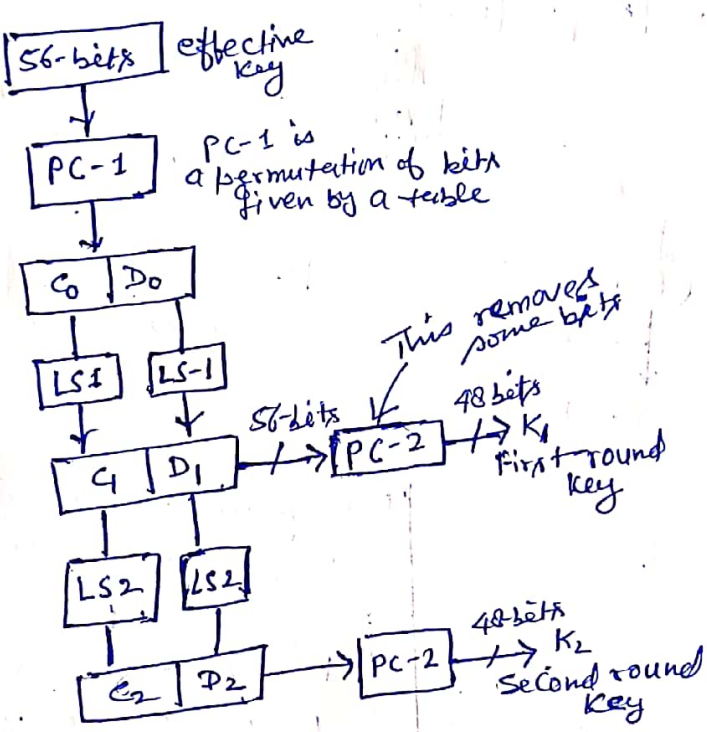
$x_4 x_3 x_2 x_1 = 0101 \leftarrow 5$

output is intersection of 3rd row & 5th column = 9 in binary 1001

Key scheduling algorithm for DES :

Key scheduling algorithm is an algorithm which generates round keys from an effective key. In DES Key scheduling algorithm generates 16 round keys $K_1, K_2, \ldots, K_{16}$ of size 48 bit from an effective key of size 56-bit.

The key scheduling algorithm is as follows



PC-1 (is a permutation of 56-bits) and PC-2 is a (permutation choice 1 & 2) for which reduces 56-bits into 48 bits These two functions PC-1 & PC-2 are given by table PC-1 & PC-2 are given by following tables

PC-2							
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

56 → (PC-2) → 48 → K_i

PC-1						
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	64	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

56 → (PC-1) → 56

LS is left circular shift either one or two circular shift in each round according to the table given below

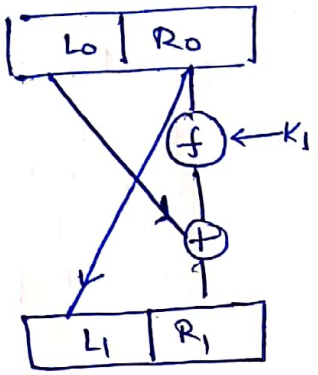
From table we can see 1 bit rotation in first round 2 bit rotation in 3rd round etc.

round number	1	2	3	4	5	6	7	8	9	10	11	13	14	15	16
bits rotation	1	1	2	2	2	2	2	2	1	2	2	2	2	2	1

Decryption of DES : DES is based on Feistel structure We can see that decryption is reverse of encryption

If we take only one round

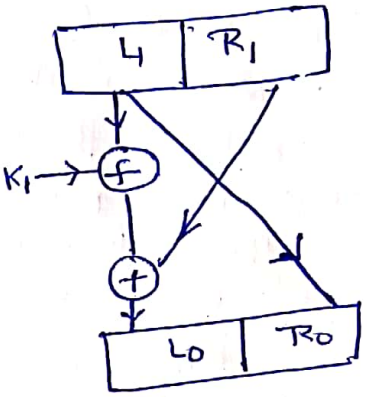
Encryption



$L_1 = R_0$

$R_1 = f(R_0, K_1) \oplus L_0$

Decryption



$R_0 = L_1$

$L_0 = R_1 \oplus f(R_0, K_1)$

$= R_1 \oplus f(L_1, K_1)$

All components of f are not invertible as we know S-box converts 6-bits into 4-bits But overall f is invertible

DES Weakness

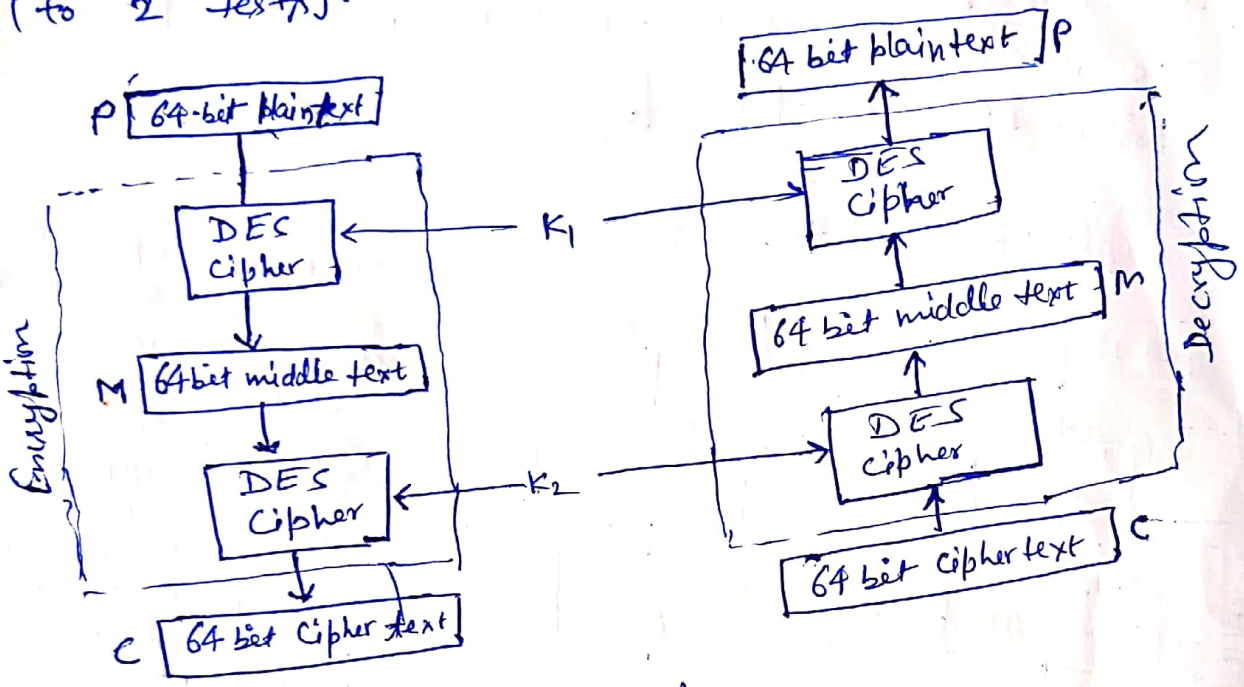
- There are some weaknesses in the cipher design of its S-boxes and P-boxes.
- Several weaknesses have been found in the cipher key
- The most serious weakness of DES is in its key size (56 bits)
- To do a brute-force attack on a given ciphertext block the adversary needs to check 2^{56} keys.
  - → With available technology, it is possible to check one million keys per second. This means that we need more than two thousand years to do brute-force attacks on DES using only a computer with one processor
  - → If we can make a computer with one million chips (parallel processing), then we can test the whole key domain in approx. 20 hours.
  - → Computer networks can simulate parallel processing. In 1977 a team of researchers used 3500 computers attached to the internet to find a key challanged by RSA Lab in 120 days.
  - → If 3500 networked computers can find the key in 120 days, a secret society with 43,000 members can find the key in 10 days.
- The above discussion shows that DES with key size 56 bits is not safe.

→ **Multiple DES** :- From above discussion we now 56 bits key size is not safe. One solution is to use multiple (cascaded) instances of DES with multiple keys : this does not require an investment in new software & hardware.

* **Double DES (2DES)** : use two instances of DES for en/dec. Each instance uses a different key which means that the key size is now doubled (112 bits). However, double DES is vulnerable to a known-plaintext attack as discussed in next part

Meet-in-the middle attack

It looks like that double DES increases the no. of tests for key search from 2^{56} (in single DES) to 2^{112} (in double DES). However, using known-plaintext attack called meet-in-the middle attack proves that double DES improves this vulnerability slightly (to 2^{57} tests).



Here $M = E_{K_1}(P)$ and $M = D_{K_2}(C)$

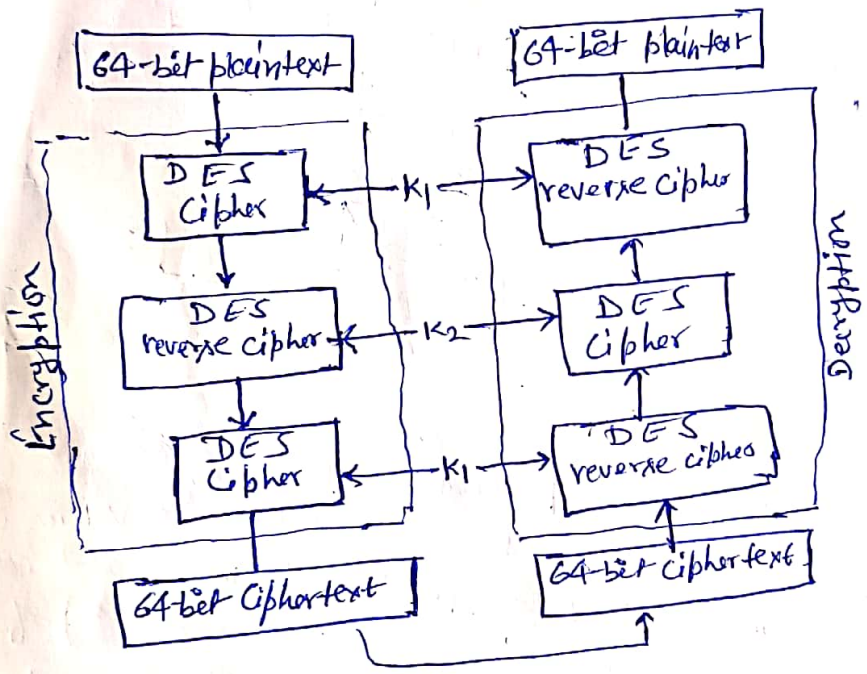
→ Assume that Eve has intercepted a previous pair P and C (known plaintext attack)
Based on first relationship above, Eve encrypts P using all possible values (2^{56}) of K_1 and records all values obtained for M. Based on second relationship above, Eve decrypt C using all possible values (2^{56}) of K_2 and record all values obtained for M. Eve creates two tables sorted by M values. She then compares the value of M until she finds those pairs of K_1 and K_2 for which the value of M is the same in both tables. Note there must be at least one pair because she is doing exhaustive search on the combination of two keys.

→ This means that instead of using 2^{112} key-search tests, Eve uses 2^{56} key search tests two times. In other words, moving from single DES to double DES, we have increased the strength from 2^{56} to 2^{57} (not to 2^{112}).

Triple DES

To improve the security of DES, triple DES (3DES) was proposed. This uses three stages of DES for encryption and decryption. Two version of triple DES are in use today: triple DES with two keys and triple DES with three keys.

→ Triple DES with two keys



To make triple DES compitable with single DES, the middle stage uses decryption (reverse cipher) in the encryption site and encryption (cipher) in the decryption site.

In this way message encrypted with single DES key K can be decrypted with triple DES if $K_1 = K_2 = K$.

It is much stronger than double DES. It has been adopted by the banking industry.