LECTURE 4

MODERN SYMMETRIC BLOCK CIPHERS

- It encrypts n bit block of plaintext.
- The encryption algorithm uses a k-bit key.
- If the message has fewer than n bits ,padding should be done to make it n bits.
- Modern block ciphers are designed as substitution ciphers
- Why ?

MODERN BLOCK CIPHER PROPERTIES

- Diffusion
  - Hiding relationship between PT and CT
  - It implies that each symbol in the cipher text is dependent on some or all the symbols of plaintext
  - Changing a single bit in the PT will result in changing each bit of CT .
- Confusion
  - It hides the relationship between the cipher text and key
  - Changing a single bit in key results in changing most or all bits in the ciphertext .

COMPONENTS OF A BLOCK CIPHER

- Most important components are S-Boxes and P-Boxes
- **P – box** (permutation) is keyless fixed transposition cipher
- In modern block ciphers there are 3 types of P-boxes

- **Straight P-Box**
- A P-box with n input bits and n output bits

- **Compression P-Box**
- A P-box with n input bits and m output bits where m < n

- **Expansion P-Box**
- A P-Box with n input bits and m output bits where m> n
- **S –Box** (Substitution Box)
- S –Box it’s a keyless fixed substitution cipher
- S – Box can have different number of inputs and outputs

- S-Box provides confusion.
- P-Box provides diffusion.
- Block ciphers use a combination of substitution and permutation techniques for encryption.

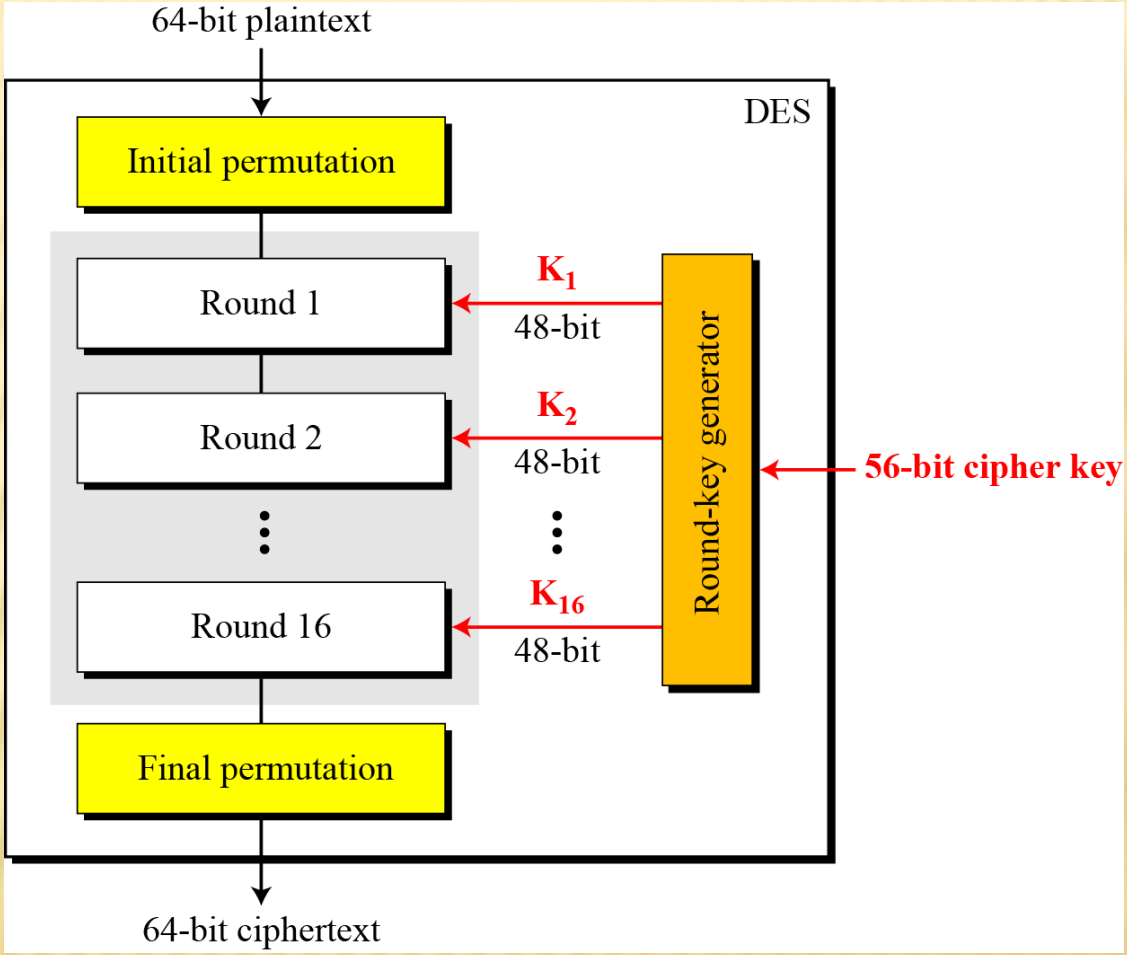
DES - DATA ENCRYPTION STANDARD

- Developed by IBM for the US Govt.
- Encrypts 64-bit data using 56-bit key.
- Encryption is done through a combination of substitution & permutation operations.

DES HISTORY

- IBM developed Lucifer cipher which encrypts 64-bit data using 128-bit key.
- Modified it as a commercial cipher which uses only 56-bit key.
- In 1972 U.S National Bureau of Standards (NBS) issued request for proposals for a national cipher standard.
- IBM submitted their revised Lucifer which was eventually accepted as the DES.

Fig : General Structure of DES



- Encryption is made of two permutations and 16 Feistel rounds.
- Each round uses a 48 bit round key.
- 64 bit plaintext passes through an Initial permutation function.
- The permuted data will go through 16 rounds of the same function.

- The 56 bit key is passed through a permutation function.
- For each round a subkey is produced.
- Its produced by the combination of left circular shift and permutation.
- The permutation function is the same for each round.
- But different subkeys are produced.

- **Initial and Final permutations**
- Each of these P-Boxes take 64 bit input and permutes them according to a predefined rule.
- These permutations are keyless and inverses of each other.
- These P-Boxes have no cryptographic significance.

Initial Permutation	*Final Permutation*
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11 03	35 03 43 11 51 19 59 27
61 53 45 37 29 21 13 05	34 02 42 10 50 18 58 26
63 55 47 39 31 23 15 07	33 01 41 09 49 17 57 25

- DES uses 16 rounds. Each round of DES is a Feistel cipher.

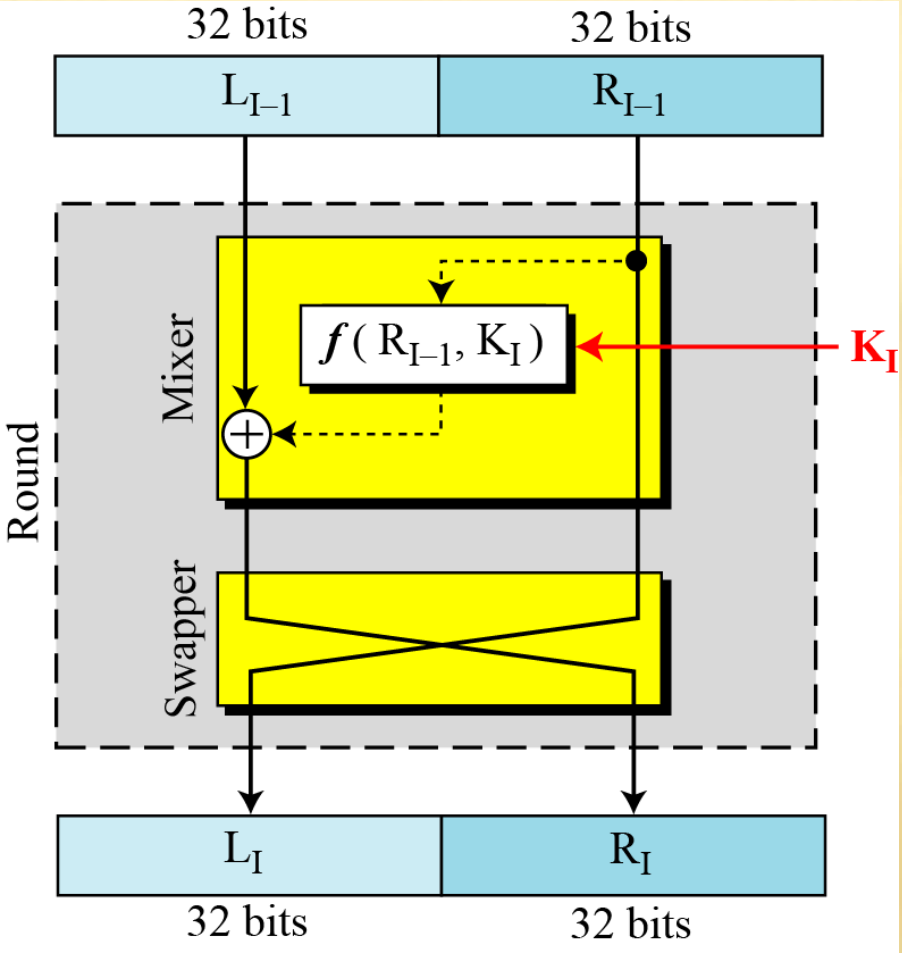
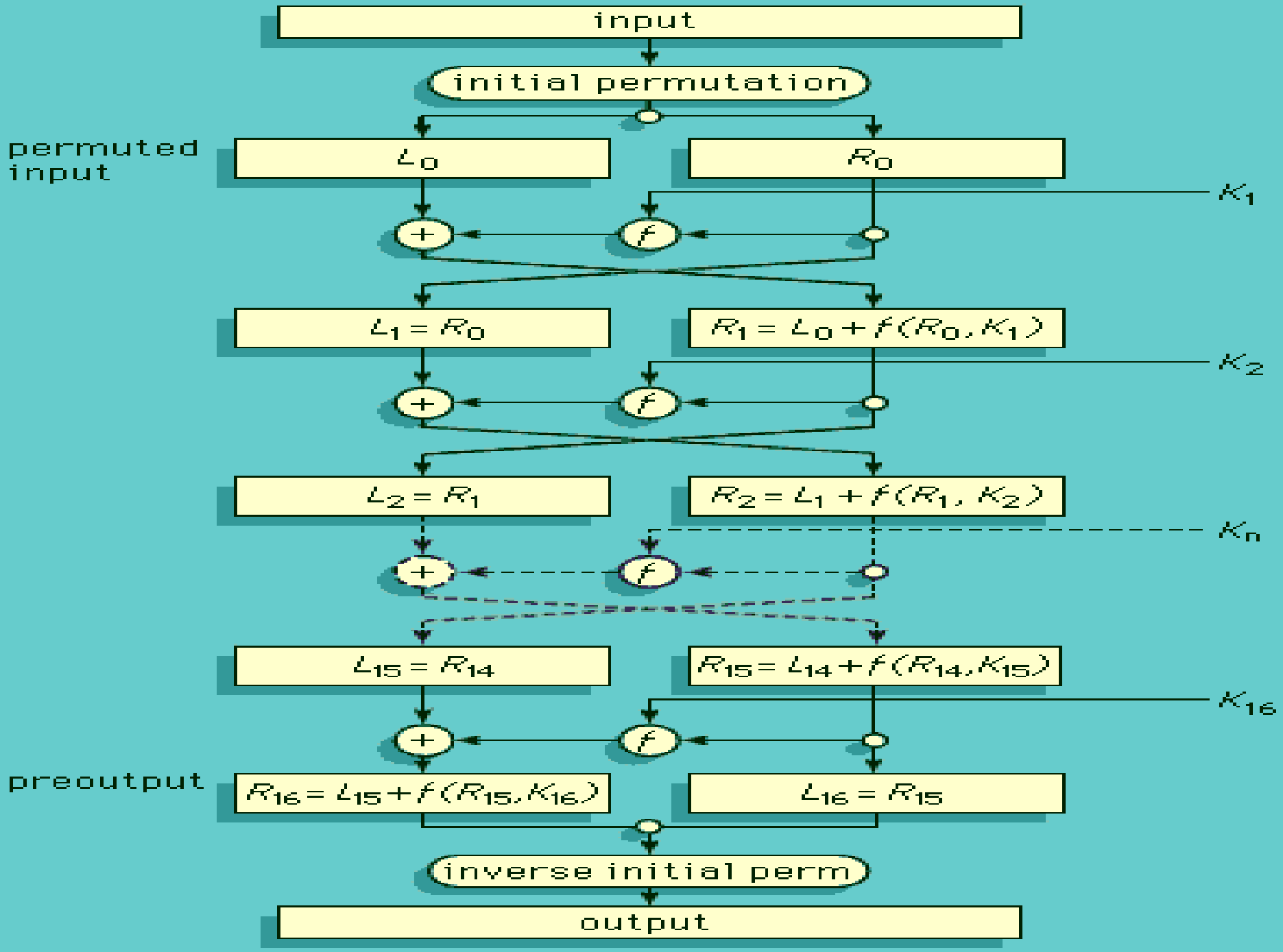


Fig : A round in DES

- The left and right half of each 64 bit intermediate value are treated as separate 32 bit quantities.
- Overall processing at each round

- $L_i = R_{i-1}$
- $R_i = L_{i-1}$ (XOR) $F (R_{i-1} , K_i)$
- Key K_i is 48 bits.
- The R input is 32 bits.
- The R input is first expanded to 48 bits by using expansion permutation table.
- It involves duplication of 16 bits of 32 bit Right half .


input
initial permutation
permuted input
L_0
R_0
K_1
+
f
$L_1 = R_0$
$R_1 = L_0 + f(R_0, K_1)$
K_2
$L_2 = R_1$
$R_2 = L_1 + f(R_1, K_2)$
K_n
$L_{15} = R_{14}$
$R_{15} = L_{14} + f(R_{14}, K_{15})$
K_{16}
preoutput
$R_{16} = L_{15} + f(R_{15}, K_{16})$
$L_{16} = R_{15}$
inverse initial perm
output

DES FUNCTION

- DES function applies a 48 bit key to the rightmost 32 bits to produce a 32 bit output.

- This function is made up of 4 components
- An expansion P-Box
- A whitener
- A group of S- Boxes
- A straight P-Box

- Expansion P-Box
- We need to expand 32 bit right half of data to operate it with a 48 bit key.

32	01	02	03	04	05
04	05	06	07	08	09
08	09	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	31	31	32	01

- **Whitener (XOR)**
- DES uses it after the expansion permutation.
- The inputs are the expanded right half and 48 bit round key.

- **S- Box**es
- DES uses 8 S-Boxes each of which takes a 6 bit input and 4 bit output.
- Each S- box is having 4 rows and 16 columns.
- The 48 bit output from the whitener is divided into eight 6 bit chunks.

- Each of these will be fed into the box, and the result will be 4 bits.
- The first and last bits of the input identify the row and the middle 4 bits identify the column.
- If the S- box consist of decimal value it will be converted to its binary equivalent.

	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	10	03	06	12	11	09	05	03	08
2	04	01	14	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

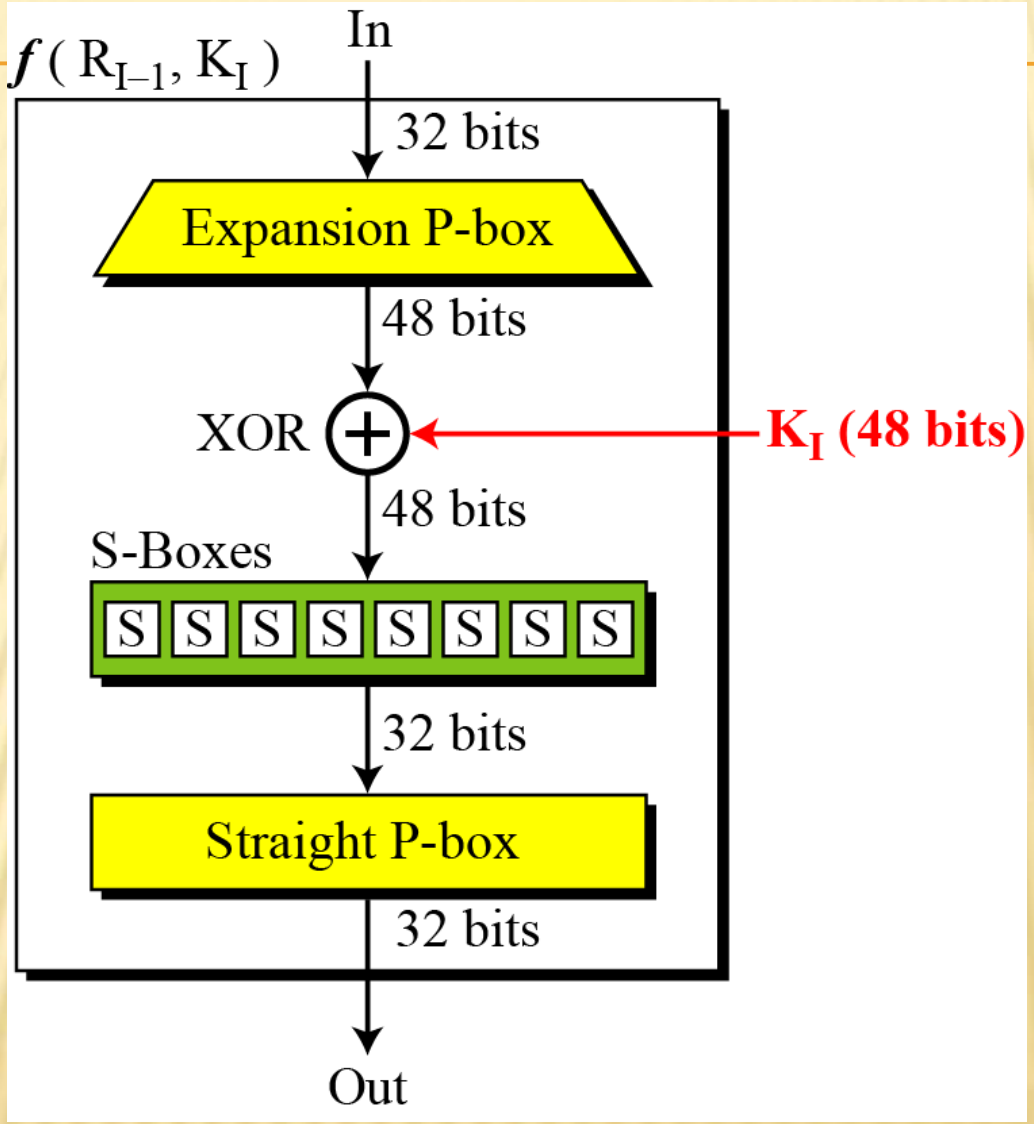
S box 1 – S1

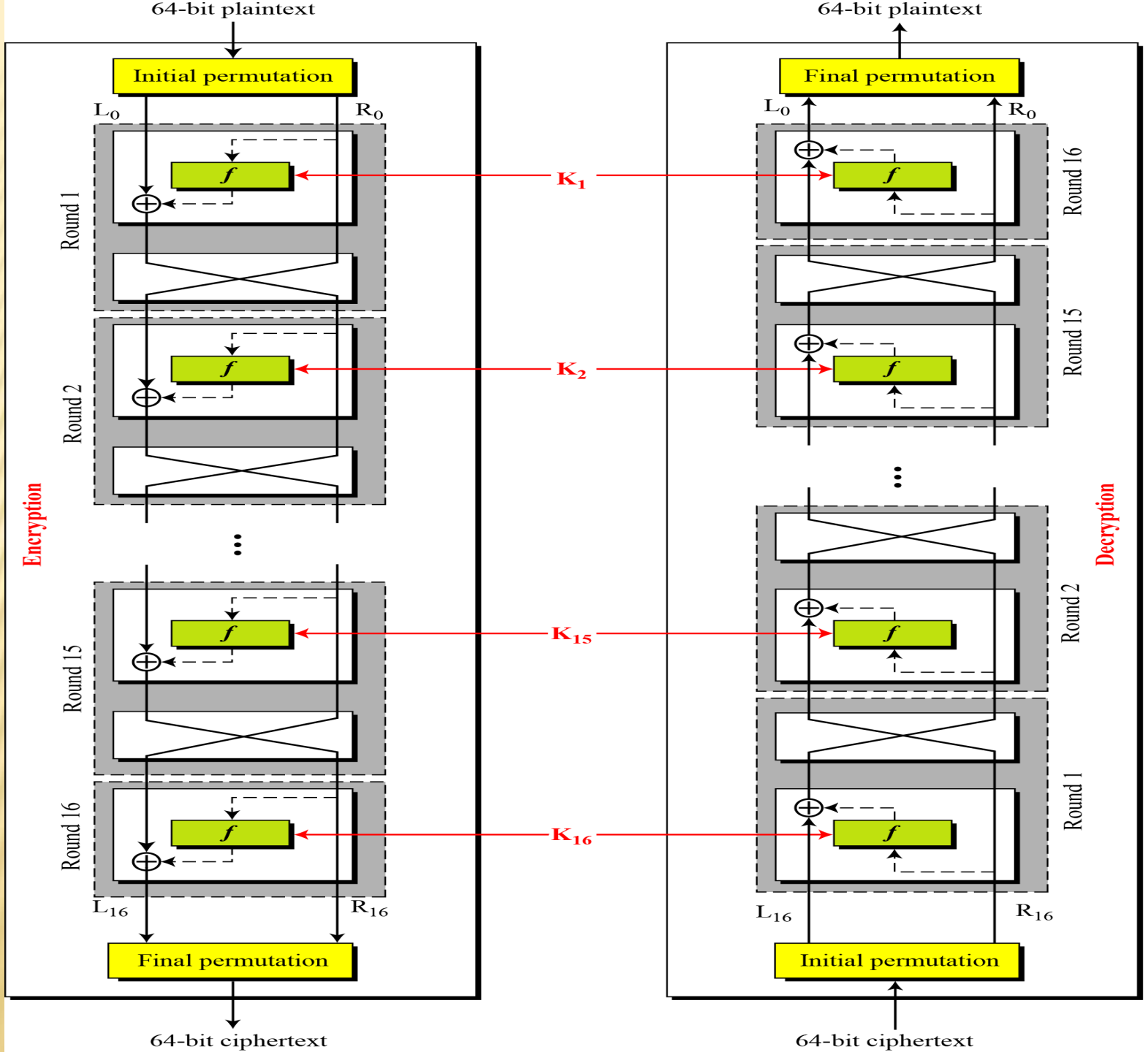
- Example

- Input to S1 – 011001
- Row – 01 ie row 1
- Column – 1100 (column 12)
- The value in row 1 and column 12
- The value is 9, so the output is 1001.

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

Fig: Straight Permutation Table


f (R I–1, K I)
In
32 bits
Expansion P-box
48 bits
XOR
K I (48 bits)
48 bits
S-Boxes
S S S S S S S S
32 bits
Straight P-box
32 bits
Out


64-bit plaintext
Initial permutation
L_0
R_0
Round 1
Round 2
Round 15
Round 16
f
K_1
K_2
K_{15}
K_{16}
Encryption
L_{16}
R_{16}
Final permutation
64-bit ciphertext
64-bit plaintext
Final permutation
Round 16
Round 15
Round 2
Round 1
Decryption
Initial permutation
64-bit ciphertext

KEY GENERATION

- The round key generator creates sixteen 48 bit keys.
- The input to the algorithm is a 64 bit key.

- **Parity Drop**
- The preprocess before key generation is called parity drop.
- Every eighth bit is dropped which makes it 56 key.

- The 56 bit key are undergoes a permutation.

57	49	41	33	25	17	09	01
58	50	42	34	26	18	10	02
59	51	43	35	27	19	11	03
60	52	44	36	63	55	47	39
31	23	15	07	62	54	46	38
30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04

- **Circular left shift**

- The resulting 56 bit key is treated as two 28 bit halves
- Each halves undergoes circular left shift of 1 or 2 bit positions .

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

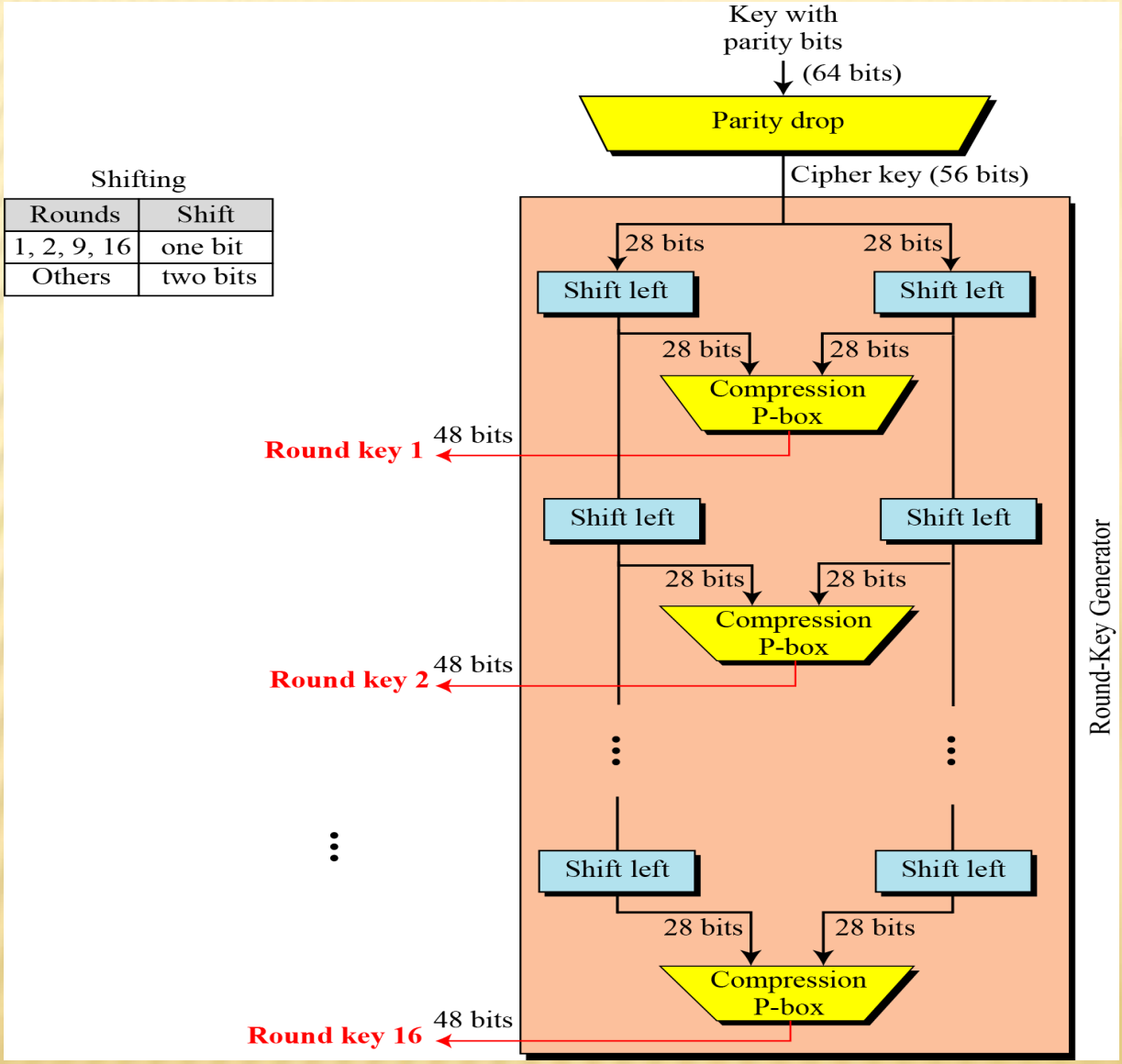
- **Compression permutation**
- This P-box changes the 56 bits into 48 bits.
- This 48 bit output serves as the key for a round.
- This key is given as input to function F.

14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Fig : Key Compression Table

Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits



DES CHARACTERISTICS

- **Avalanche effect** :

- A small alteration of plaintext results in a large change in the cipher text.
- DES exhibits strong avalanche effect.

- **Completeness Effect :**

- Each bit of the cipher text needs to depend on many bits on the plaintext.

- The diffusion and confusion produced by P-Boxes and S-Boxes provides DES with completeness effect

- **Weakness in the cipher key**

- **Weak key**

- A weak key consist of all 0’s all 1’s or half 0’s and half 1’s after parity drop
- The round key created form any of these weak keys are the same.
- If we encrypt a block with a weak key and subsequently encrypt the result with the same weak key ,we get original block.
- Four out of the 2^{56} keys are weak keys.

- **Semi-weak keys**

- A semi-weak key creates only two different round keys and each of them is repeated eight times.

Round key 1	9153E54319BD	6EAC1ABCE642
Round key 2	6EAC1ABCE642	9153E54319BD
Round key 3	6EAC1ABCE642	9153E54319BD
Round key 4	6EAC1ABCE642	9153E54319BD
Round key 5	6EAC1ABCE642	9153E54319BD
Round key 6	6EAC1ABCE642	9153E54319BD
Round key 7	6EAC1ABCE642	9153E54319BD
Round key 8	6EAC1ABCE642	9153E54319BD
Round key 9	9153E54319BD	6EAC1ABCE642
Round key 10	9153E54319BD	6EAC1ABCE642
Round key 11	9153E54319BD	6EAC1ABCE642
Round key 12	9153E54319BD	6EAC1ABCE642
Round key 13	9153E54319BD	6EAC1ABCE642
Round key 14	9153E54319BD	6EAC1ABCE642
Round key 15	9153E54319BD	6EAC1ABCE642
Round key 16	6EAC1ABCE642	9153E54319BD

- **Possible weak key**
- A key that creates only four distinct round keys.

- **Key Compliment**

- In the key domain half of the keys are compliments of each other.
- A key complement is made by inverting the bits in the key.

Security of DES

- Brute force attack is possible.
- DES is resistant to differential cryptanalysis.
  - Designers of DES designed S-Box and chose 16 as the number of rounds to make DES resistant to this type of attack
  - It's a chosen plaintext attack
- DES is vulnerable to linear cryptanalysis.
  - It's a known plaintext attack

MULTIPLE ENCRYPTION WITH DES

- DES is vulnerable to Brute-force attack.
- One alternative for that is multiple encryption with DES and multiple keys.

- The alternatives are
  - Double DES
  - Triple DES with two keys
  - Triple DES with three keys

DOUBLE DES

- Simplest form of multiple encryption.
- It has two stages and two keys.

- Encryption is given as

  $C = E_{k2} [E_{k1} [P]]$
- Decryption is given as

  $P = D_{k1} [D_{k2} [C]]$

- The key length here is 112 bits.

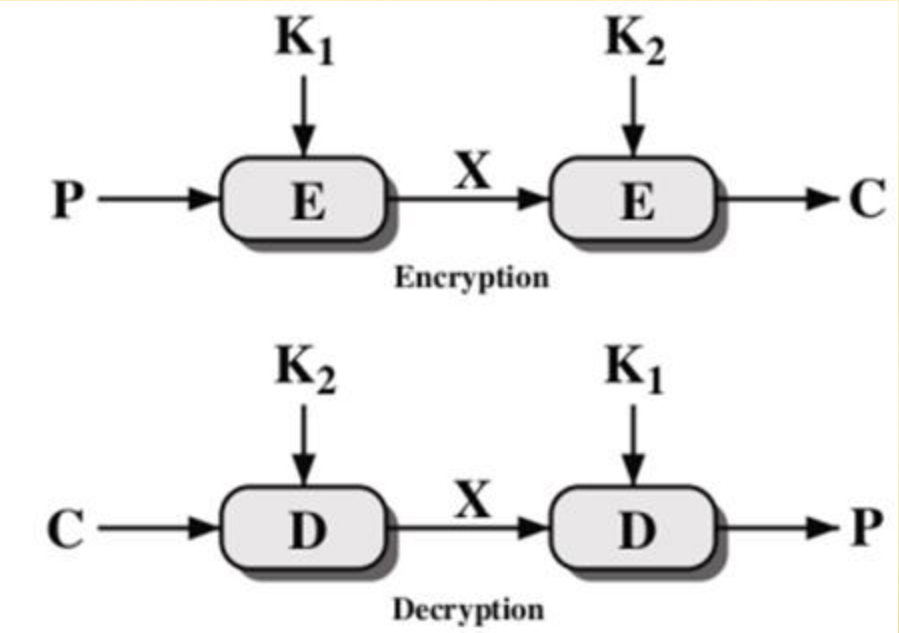


Fig : Double DES

- Brute- force attack is difficult since there are 2^{112} possibilities.
- **Meet in the middle** attack is possible.

- $X = E_{k1} [P] = D_{k2} [C]$

- Suppose the adversary has intercepted plaintext and cipher text pair (P , C).

- Encrypt P using all 2^{56} possible values of K1.
- Store the results in a table.
- Next decrypt C using all 2^{56} possible values of K2.
- Store the result in a table .
- Both the tables are sorted according to the value of X.
- Now he compares the values of X until he finds the match.

- If there is only one match, then attacker obtain (K1, K2).
- If there is more than one match found , attacker takes another pair of PT and CT and use each candidate key pairs to see if he can get the cipher text from the plaintext.
- If he again finds more than one candidate pair , repeat the previous step until he finds a unique match.

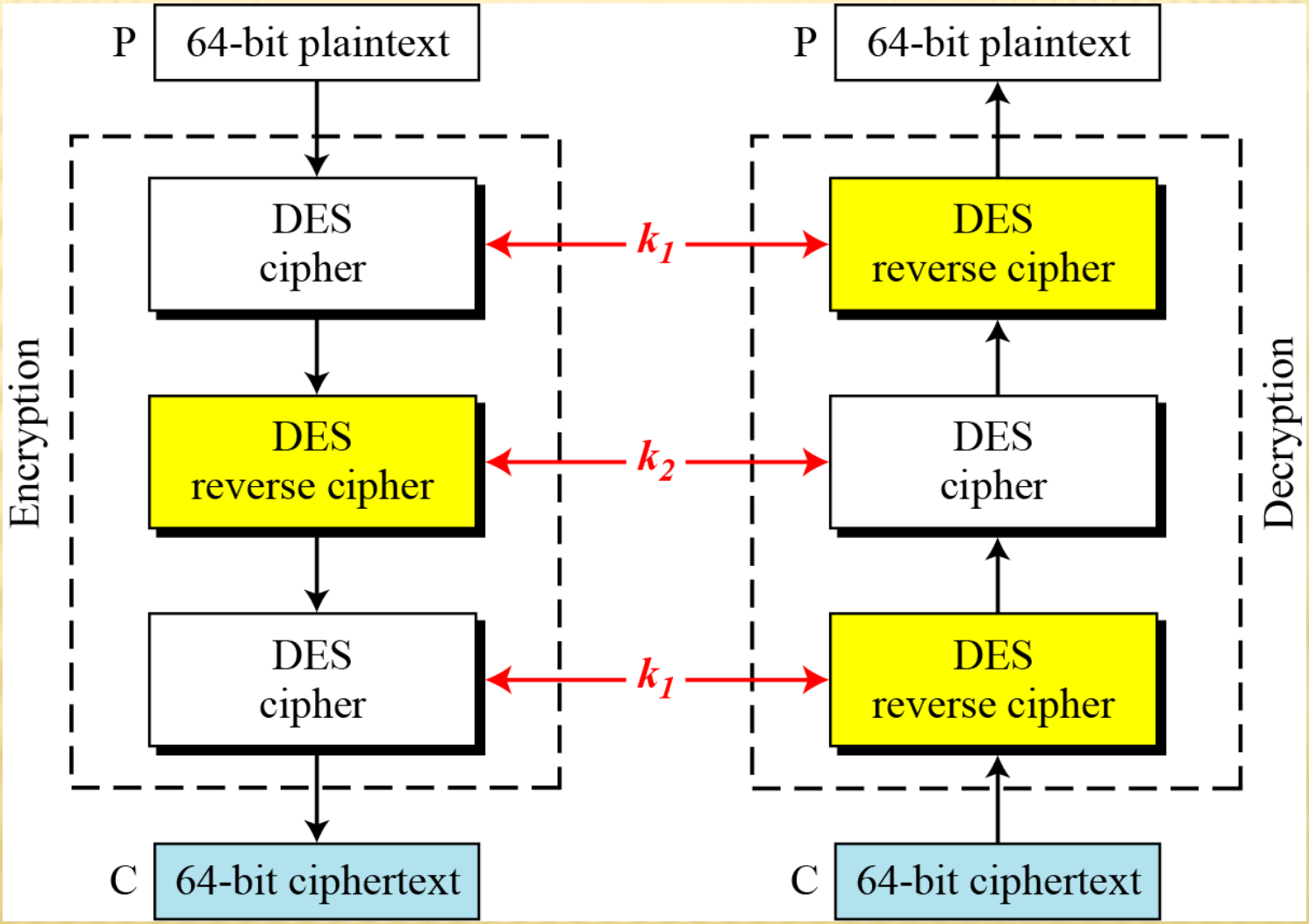
TRIPLE DES WITH TWO KEYS

- In this there are 3 stages of encryption with two different keys.
- The function follows an encrypt-decrypt-encrypt sequence.

- The cipher text is obtained as follows.

$$C = E_{k1} [D_{k2} [E_{k1} [P]]]$$

Fig: Triple DES with two keys



- Brute force attack is difficult on triple DES with two keys.
- There are 2^{112} possibilities.
- It’s vulnerable to known plaintext attack .

TRIPLE DES WITH THREE KEYS

- It use three of encryption stages with three different keys.
- It has a key length of 168 bits.
- Cipher text is obtained as

  $C = E_{k3} [D_{k2} [E_{k1} [P]]]$

- Brute force attack is not possible because of 2^{168} possibilities.
- Cryptanalytic attacks are also not possible.