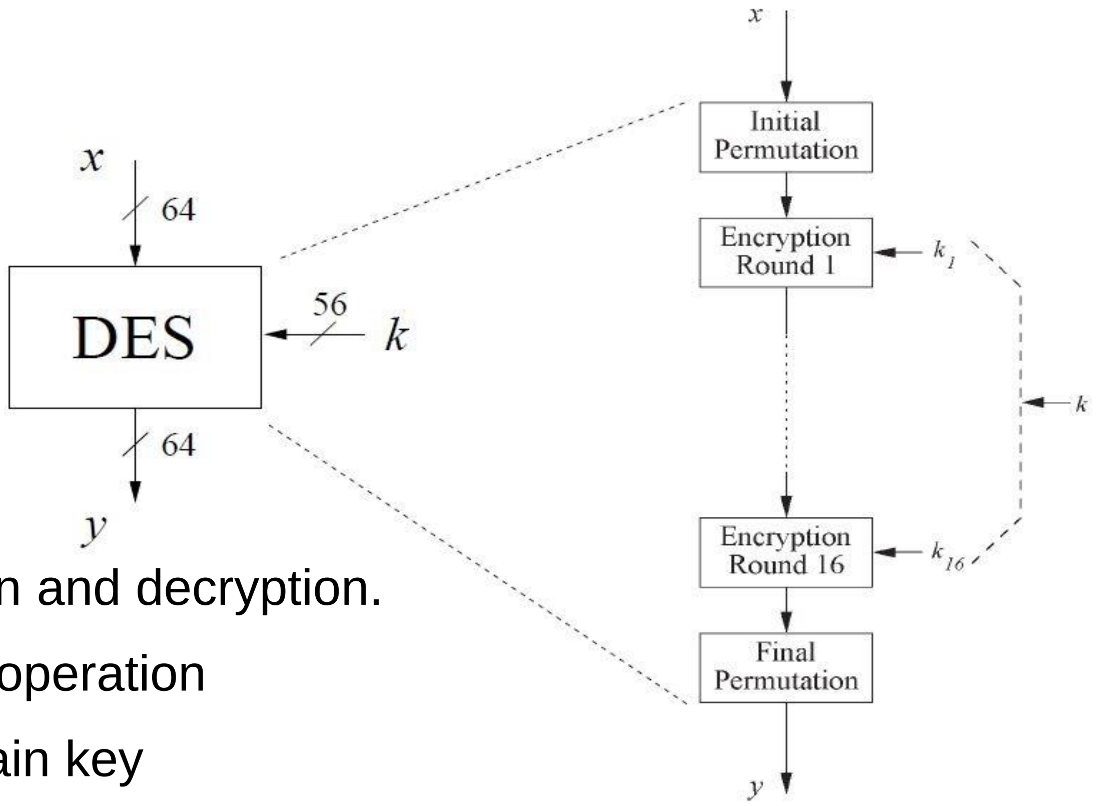
DES Algorithm

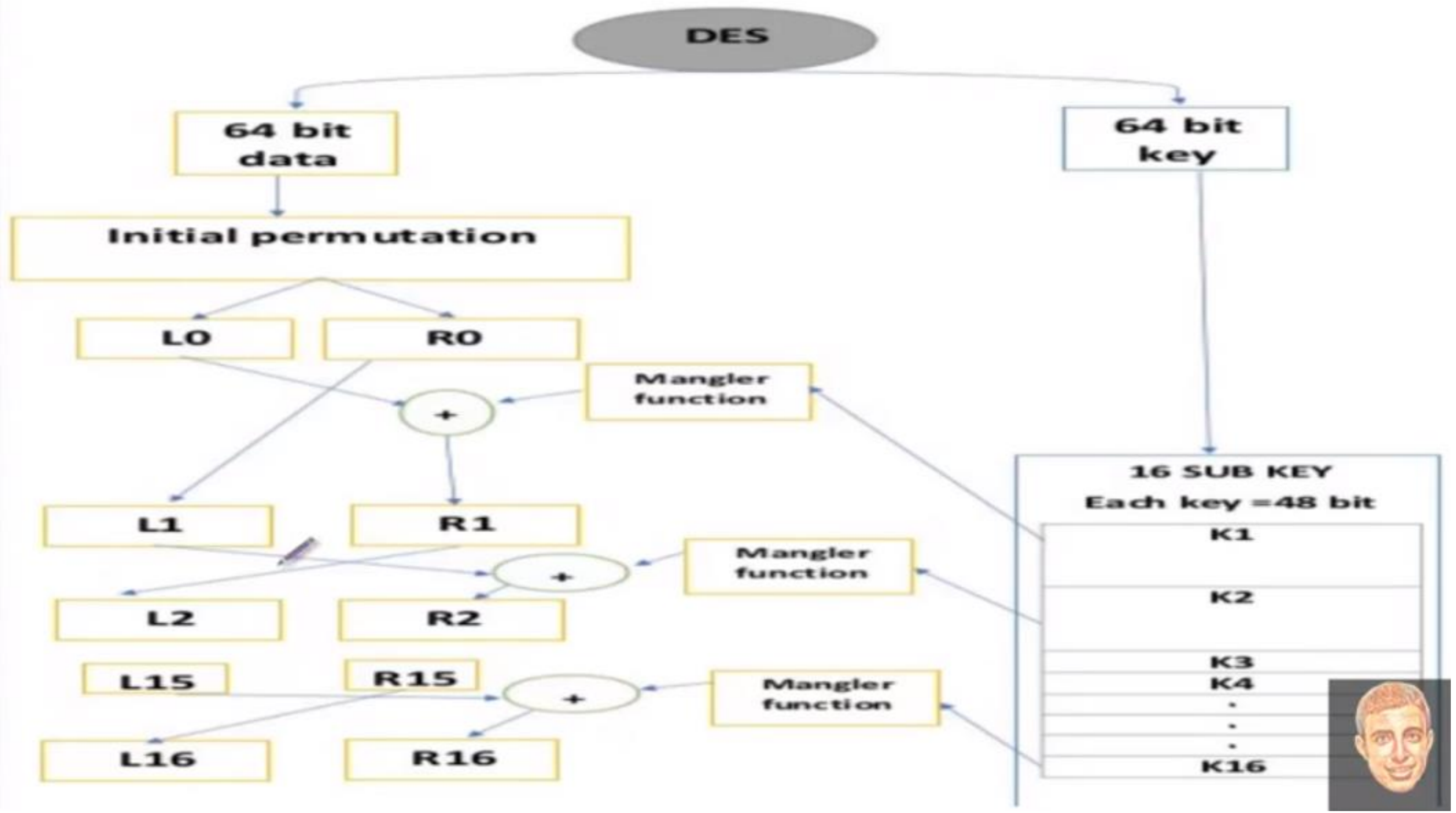
BLOCK CIPHER

Overview

- **Overview of the DES Algorithm.**
- **Encrypts blocks of size 64 bits.**
- **Uses a key of size 56 bits.**



- Symmetric cipher: uses same key for encryption and decryption.
- Uses 16 rounds which all perform the identical operation
- Different subkey in each round derived from main key


DES
64 bit data
64 bit key
Initial permutation
L0
R0
Mangler function
+
L1
R1
Mangler function
+
L2
R2
L15
R15
Mangler function
+
L16
R16
16 SUB KEY
Each key =48 bit
K1
K2
K3
K4
.
.
.
K16

Key Generation

- **Key 64 bit.**
- **Quantity = 8bit parity + 56 bit key.**
- **Every 8^{th} key is the parity bit.**
- **Uses a key of size 56 bits.**

00010011	1
00110100	2
01010111	3
01111001	4
10011011	5
10111100	6
11011111	7
11110001	8

- Symmetric cipher: uses same key for encryption and decryption.
- Uses 16 rounds which all perform the identical operation
- Different subkey in each round derived from main key

Key Generation

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	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Example: From the original 64-bit key

K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

we get the 56-bit permutation

K+ = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

Key Generation

K+ = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

C_0 = 1111000 0110011 0010101 0101111

D_0 = 0101010 1011001 1001111 0001111

Key Generation

C_0 = 1111000011001100101010101111
D_0 = 0101010101100110011110001111

C_1 = 1110000110011001010101011111
D_1 = 1010101011001100111100011110

C_2 = 1100001100110010101010111111
D_2 = 0101010110011001111000111101

C_3 = 0000110011001010101011111111
D_3 = 0101011001100111100011110101

C_4 = 0011001100101010101111111100
D_4 = 0101100110011110001111010101

C_5 = 1100110010101010111111110000
D_5 = 0110011001111000111101010101

C_6 = 0011001010101011111111000011
D_6 = 1001100111100011110101010101

C_7 = 1100101010101111111100001100
D_7 = 0110011110001111010101010110

C_8 = 0010101010111111110000110011
D_8 = 1001111000111101010101011001

C_9 = 0101010101111111100001100110
D_9 = 0011110001111010101010110011

C_{10} = 0101010111111110000110011001
D_{10} = 1111000111101010101011001100

C_{11} = 0101011111111000011001100101
D_{11} = 1100011110101010101100110011

C_{12} = 0101111111100001100110010101
D_{12} = 0001111010101010110011001111

C_{13} = 0111111110000110011001010101
D_{13} = 0111101010101011001100111100

C_{14} = 1111111000011001100101010101
D_{14} = 1110101010101100110011110001

C_{15} = 1111100001100110010101010111
D_{15} = 1010101010110011001111000111

C_{16} = 1111000011001100101010101
D_{16} = 01010101011001100111100001

Iteration Number	Number 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

Key Generation

$C_0 = 1111000011001100101010101111$
$D_0 = 0101010101100110011110001111$

$C_1 = 1110000110011001010101011111$
$D_1 = 1010101011001100111100011110$

$C_2 = 1100001100110010101010111111$
$D_2 = 0101010110011001111000111101$

$C_3 = 0000110011001010101011111111$
$D_3 = 0101011001100111100011110101$

$C_4 = 0011001100101010101111111100$
$D_4 = 0101100110011110001111010101$

$C_5 = 1100110010101010111111110000$
$D_5 = 0110011001111000111101010101$

$C_6 = 0011001010101011111111000011$
$D_6 = 1001100111100011110101010101$

$C_7 = 1100101010101111111100001100$
$D_7 = 0110011110001111010101010110$

$C_8 = 0010101010111111110000110011$
$D_8 = 1001111000111101010101011001$

$C_9 = 0101010101111111100001100110$
$D_9 = 0011110001111010101010110011$

$C_{10} = 0101010111111110000110011001$
$D_{10} = 1111000111101010101011001100$

$C_{11} = 0101011111111000011001100101$
$D_{11} = 1100011110101010101100110011$

$C_{12} = 0101111111100001100110010101$
$D_{12} = 0001111010101010110011001111$

$C_{13} = 0111111110000110011001010101$
$D_{13} = 0111101010101011001100111100$

$C_{14} = 1111111000011001100101010101$
$D_{14} = 1110101010101100110011110001$

$C_{15} = 1111100001100110010101010111$
$D_{15} = 1010101010110011001111000111$

$C_{16} = 1111000011001100101010101111$
$D_{16} = 0101010101100110011110001111$

Number of Left Shifts

1
1
2
2
2
2
2
2
1
2
2
2
2
2
2
1

Key Generation

PC-2 48 bit

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

$$C_1 = 1110000110011001010101011111$$
$$D_1 = 1010101011001100111100011110$$

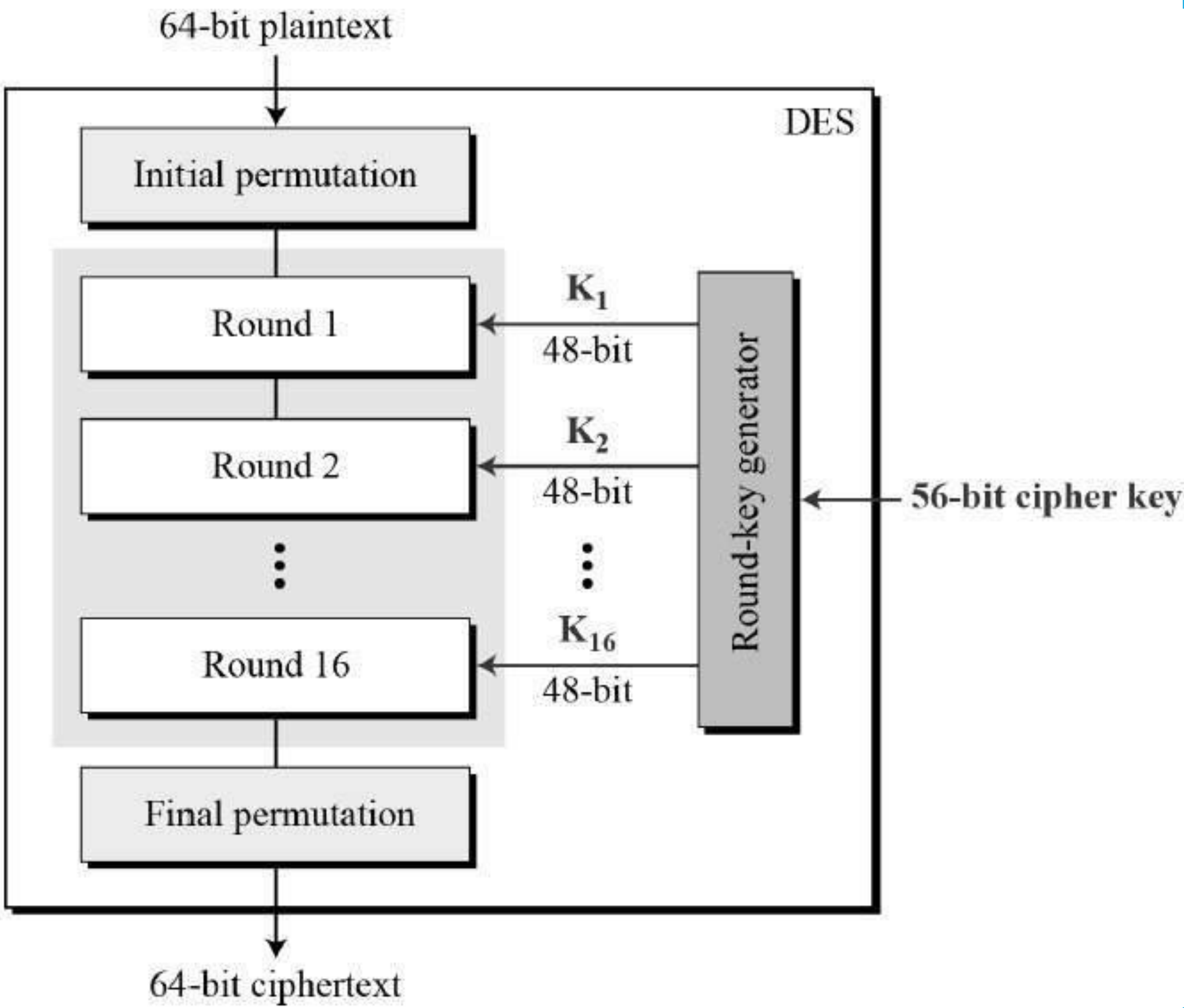
Key Generation

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

For the other keys we have

K_2 = 011110 011010 111011 011001 110110 111100 100111 100101
K_3 = 010101 011111 110010 001010 010000 101100 111110 011001
K_4 = 011100 101010 110111 010110 110110 110011 010100 011101
K_5 = 011111 001110 110000 000111 111010 110101 001110 101000
K_6 = 011000 111010 010100 111110 010100 000111 101100 101111
K_7 = 111011 001000 010010 110111 111101 100001 100010 111100
K_8 = 111101 111000 101000 111010 110000 010011 101111 111011
K_9 = 111000 001101 101111 101011 111011 011110 011110 000001
K_{10} = 101100 011111 001101 000111 101110 100100 011001 001111
K_{11} = 001000 010101 111111 010011 110111 101101 001110 000110
K_{12} = 011101 010111 000111 110101 100101 000110 011111 101001
K_{13} = 100101 111100 010111 010001 111110 101011 101001 000001
K_{14} = 010111 110100 001110 110111 111100 101110 011100 111010
K_{15} = 101111 111001 000110 001101 001111 010011 111100 001010
K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101

DES Encryption



DES Encryption

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

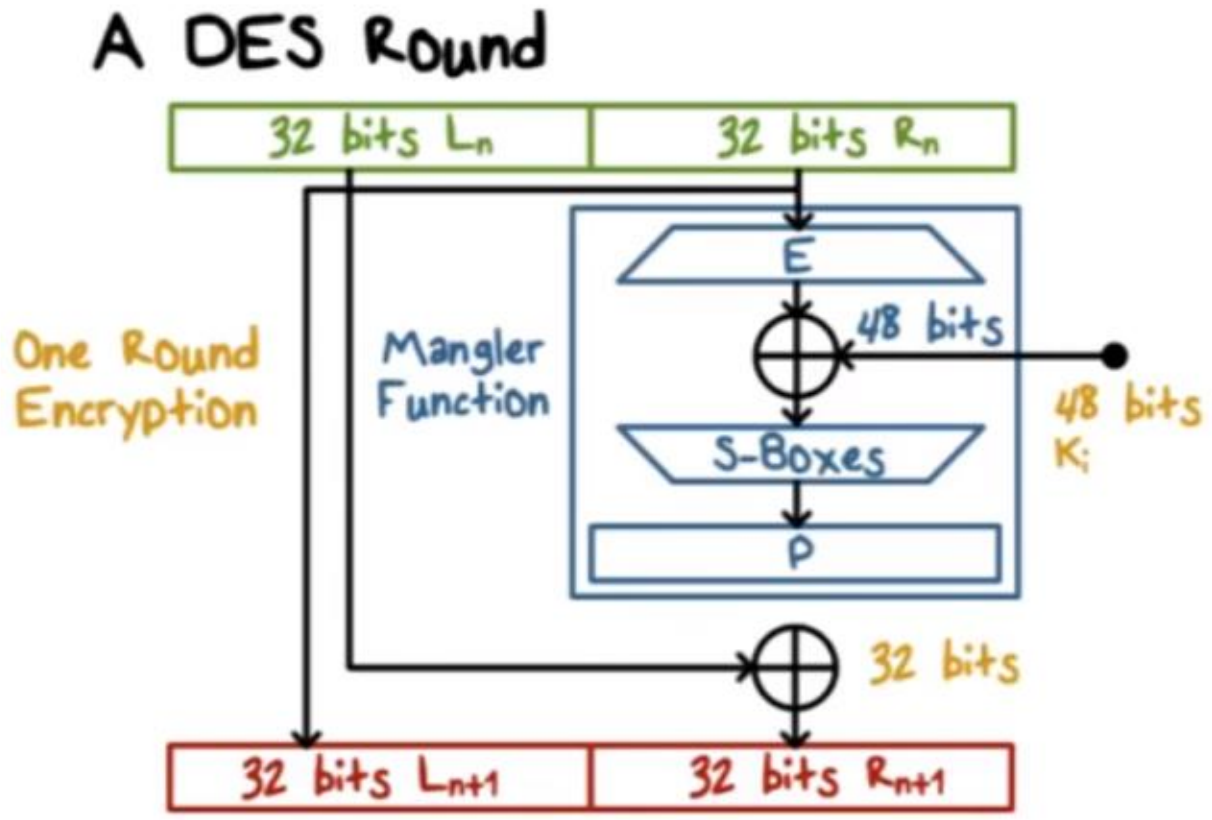
M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
IP = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010

DES Encryption

M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
IP = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010

L_0 = 1100 1100 0000 0000 1100 1100 1111 11
R_0 = 1111 0000 1010 1010 1111 0000 1010 10

DES Encryption

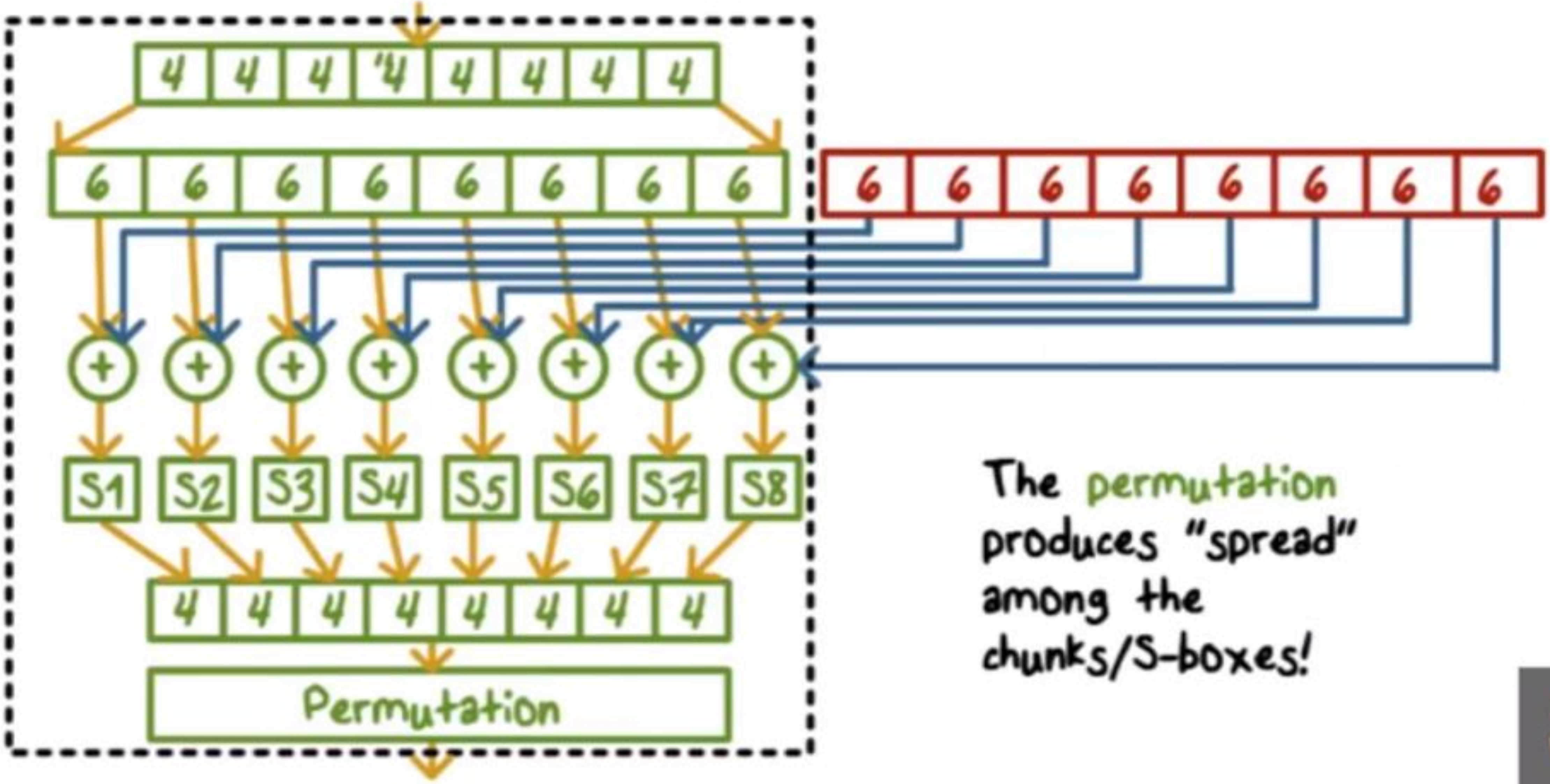


- Can be expressed as:

  $L_{n+1} = R_n$

  $R_{n+1} = L_n \text{ XOR } M(R_n, K_n)$

Mangler Function



The permutation produces "spread" among the chunks/S-boxes!

DES Encryption

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

$\boldsymbol{R_0}$ = 1111 0000 1010 1010 1111 0000 1010 1010

$\mathbf{E}(\boldsymbol{R_0})$ = 011110 100001 010101 010101 011110 100001 010101 010101

DES Encryption

R_0 = 1111 0000 1010 1010 1111 0000 1010 1010

$E(R_0)$ = 011110 100001 010101 010101 011110 100001 010101 010101

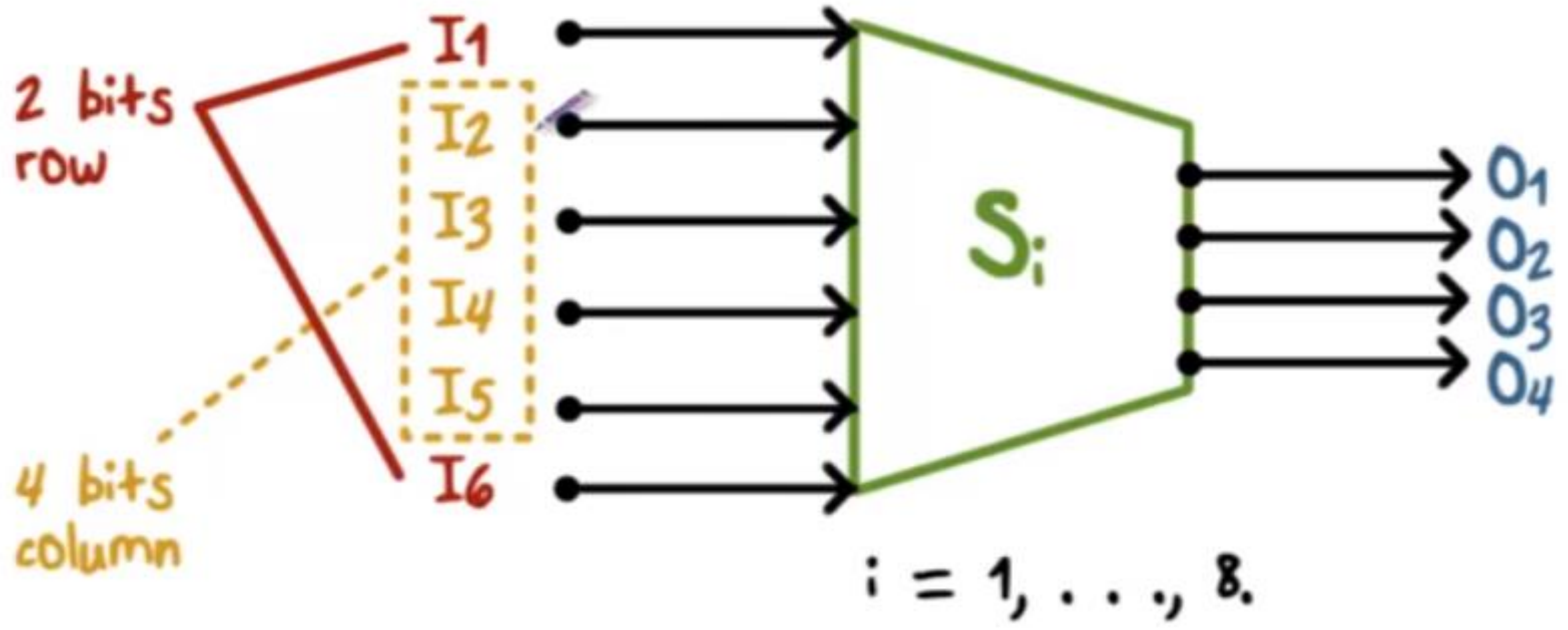
K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

$K_1 + E(R_0)$ = 011000 010001 011110 111010 100001 100110 010100 100111.

$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ = 0101 1100 1000 0010 1011 0101 1001 0111

S-Box (Substitute and Shrink)

- 48 bits => 32 bits. (8*6 => 8*4)
- 2 bits used to select amongst 4 substitutions for the rest of the 4-bit quantity



DES Encryption

Example:

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

DES Encryption

$K_1 + \mathbf{E}(R_0) =$ 011000 010001 011110 111010 100001 100110 010100 100111.

$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8) =$ 0101 1100 1000 0010 1011 0101 1001 0111

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

$f =$ 0010 0011 0100 1010 1010 1001 1011 1011

DES Encryption

$L_0 =$ 1100 1100 0000 0000 1100 1100 1111 11

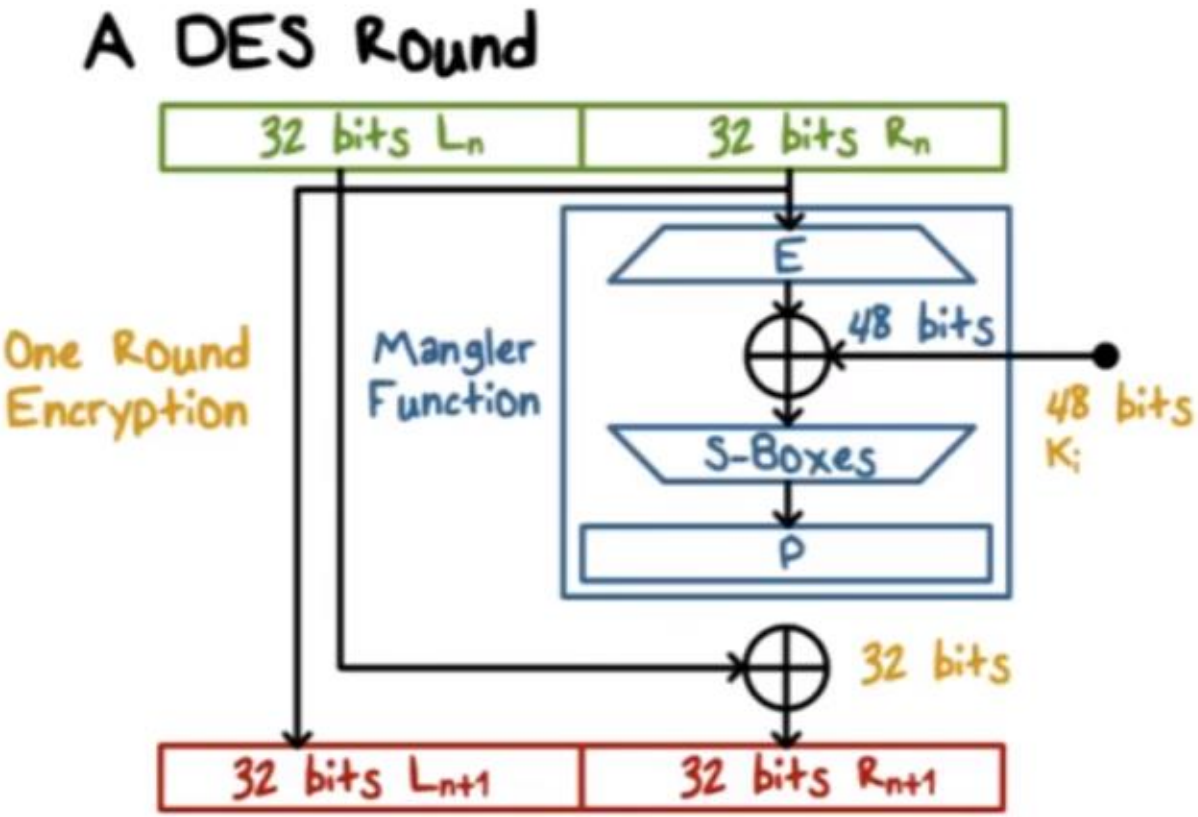
$R_0 =$ 1111 0000 1010 1010 1111 0000 1010 10

$f =$ 0010 0011 0100 1010 1010 1001 1011 1011

$R_1 = L_0 + f(R_0 , K_1)$

= 1100 1100 0000 0000 1100 1100 1111 1111
+ 0010 0011 0100 1010 1010 1001 1011 1011
= 1110 1111 0100 1010 0110 0101 0100 0100

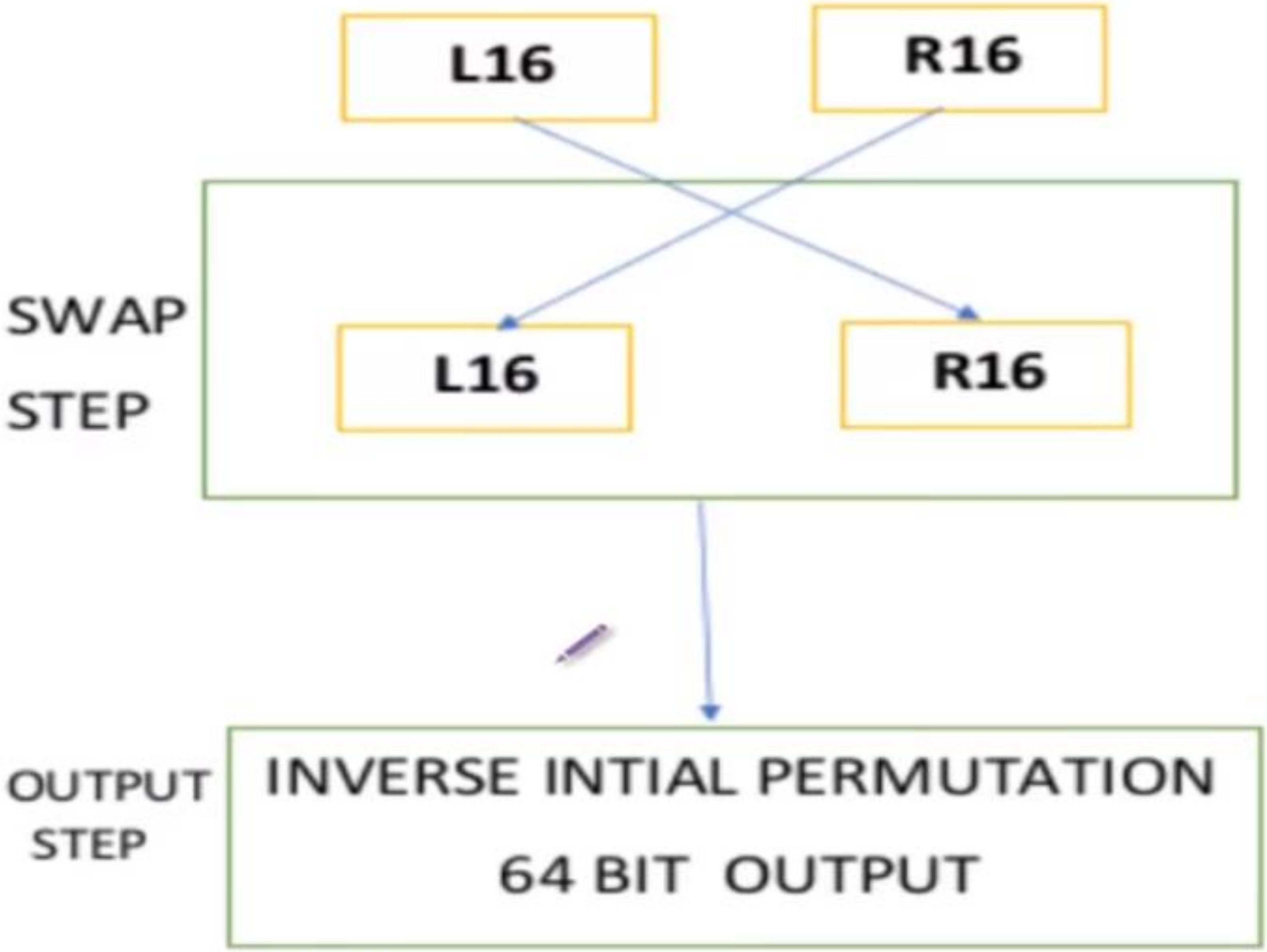
DES Encryption



- Can be expressed as:

  $L_{n+1}=R_n$

  $R_{n+1}=L_n \text{ XOR } M(R_n,K_n)$


L16
R16
SWAP
STEP
L16
R16
OUTPUT
STEP
INVERSE INTIAL PERMUTATION
64 BIT OUTPUT

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

L_{16} = 0100 0011 0100 0010 0011 0010 0011 0100
R_{16} = 0000 1010 0100 1100 1101 1001 1001 0101

$R_{16}L_{16}$ = 00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100

IP^{-1} = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101

M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

M = 0123456789ABCDEF

K = 133457799BBCDFF1

C=85E813540F0AB405

IP^{-1} = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101

which in hexadecimal format is

85E813540F0AB405.

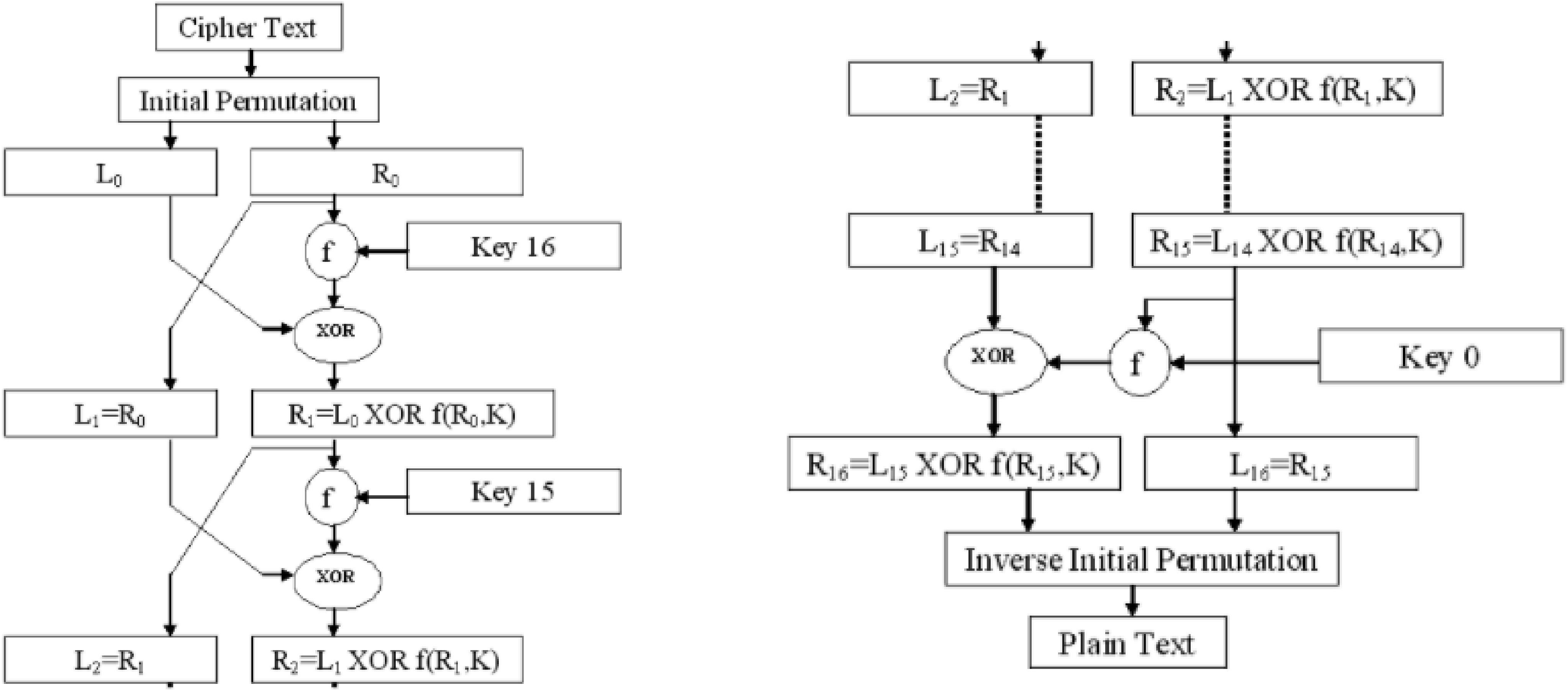
This is the encrypted form of **M** = 0123456789ABCDEF: namely, **C** = 85E813540F0AB405.

DES Decryption

M = 0123456789ABCDEF

K = 133457799BBCDFF1

C=85E813540F0AB405



Question-1

3.1. As stated in Sect. 3.5.2, one important property which makes DES secure is that the S-boxes are nonlinear. In this problem we verify this property by computing the output of S_1 for several pairs of inputs.

Show that $S_1(x_1) \oplus S_1(x_2) \neq S_1(x_1 \oplus x_2)$, where "$\oplus$" denotes bitwise XOR, for:

1. $x_1 = 000000, x_2 = 000001$
2. $x_1 = 111111, x_2 = 100000$

	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

$S[0] : (x_0, \underbrace{x_1, x_2, x_3, x_4}_{\text{column}}, x_5) \rightarrow (y_0, y_1, y_2, y_3)$

$(1, 1, 0, 0, 1, 1)$: row 3, column 9, $S[0](1, 1, 0, 0, 1, 1) = 11 = (1, 0, 1, 1)$

Lets try for x of i where i from 1 to 5

x1,x2,x3,x4,x5

Question-1 Solution

	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

$S[0] : (x_0, \underbrace{x_1, x_2, x_3, x_4}_{\text{column}}, x_5) \rightarrow (y_0, y_1, y_2, y_3)$

$(1, 1, 0, 0, 1, 1)$: row 3, column 9, $S[0](1, 1, 0, 0, 1, 1) = 11 = (1, 0, 1, 1)$

1. $x_1 = 000000, x_2 = 000001$ 00 = 0, 0000 = 0, s(x1) =14 =1110, 01 = 1, 0000 =0, s(x2) = 0 = 0000

2. $x_1 = 111111, x_2 = 100000$ 11 = 3, 1111 = 15, s(x1) =13 =1101, 10 = 2, 0000 =0, s(x2) = 4 = 0100

1.Right 1110⨁0000 =1110, Left 000000⨁000001 = 000001, s(x3) = 0000, **1110≠0000**

2.Right 1101⨁0100 = 1001, Left 111111⨁100000 = 011111, s(x3) = 1000, **1001≠1000** **(NOT LINEAR)**

Question-2

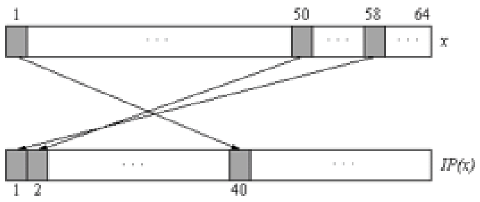
3.2. We want to verify that $IP(\cdot)$ and $IP^{-1}(\cdot)$ are truly inverse operations. We consider a vector $x = (x_1, x_2, \ldots, x_{64})$ of 64 bit. Show that $IP^{-1}(IP(x)) = x$ for the first five bits of x, i.e. for x_i, $i = 1, 2, 3, 4, 5$.

Lets try for x of i where i from 1 to 5

x1,x2,x3,x4,x5

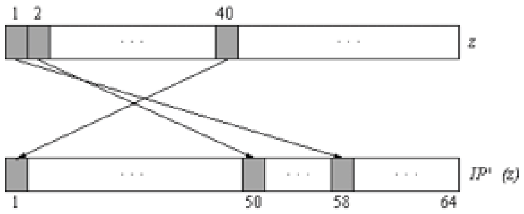
Initial Permutation

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



Final Permutation

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



Question-2 Solution

Lets try for x of i (where i from 1 to 5)

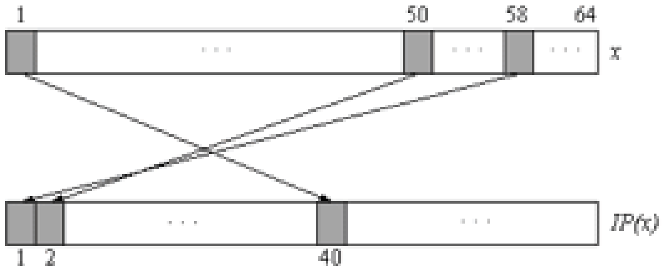
x1 , x2, x3 , x4, x5

x40, x8, x48, x16, x56

x1 , x2, x3 , x4, x5

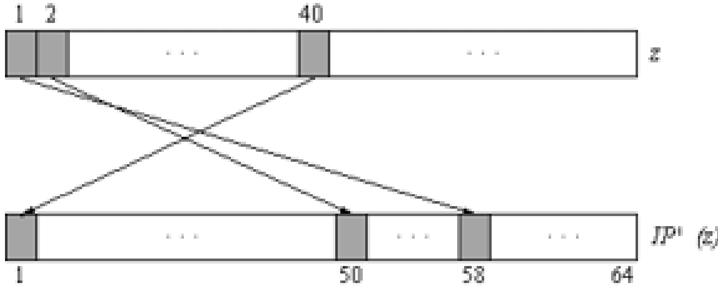
Initial Permutation

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



Final Permutation

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



M = 0123456789ABCDEF

K = 133457799BBCDFF1

C=85E813540F0AB405

Assignment

Use DES to encrypt and decrypt a message with the following requirements for the DES:

- Message will be entered in hexadecimal format you will have to convert it to binary.
- Key will be entered in hexadecimal format you will have to convert it to binary.
- You have to show every step results in the CLI (sub keys and each permutations results)
- S box will be the same in our Assignment only but in real life scenario it suppose to be different.
- You should decrypt the message and get the original one in hexadecimal format.
- Will be submitted on blackboard by max 20th of Nov 2021, and will be discussed on the next lab.

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