Computer Security
Lecture 4

Block Ciphers and the Data Encryption Standard

Dr. Mohamed Loey

Lecturer, Faculty of Computers and Information

Benha University

Egypt

Table of Contents

Stream Ciphers and Block Ciphers

Data Encryption Standard

DES Algorithm

DES Key Creation

DES Encryption

The Strength Of DES

Table of Contents

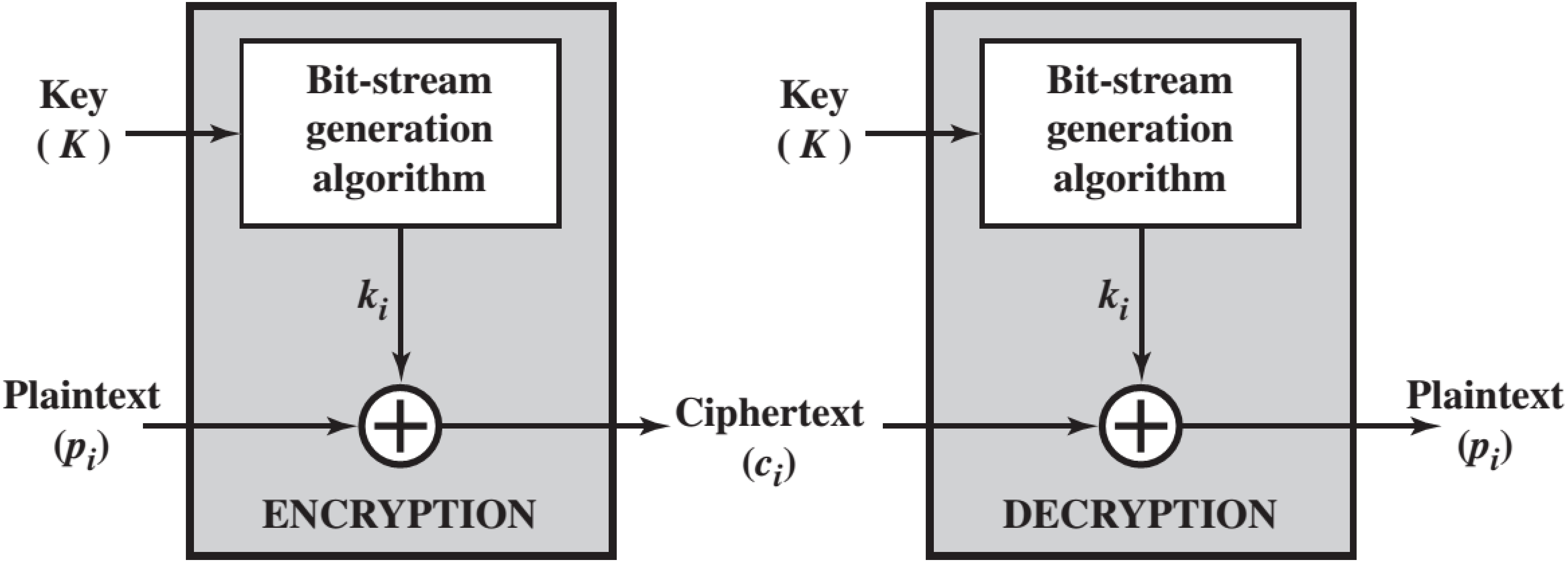
- Stream Ciphers and Block Ciphers
- Data Encryption Standard
- DES Algorithm
- DES Key Creation
- DES Encryption
- The Strength Of DES

Stream Ciphers and Block Ciphers

- **Stream cipher** is one that encrypts a digital data stream one bit or one byte at a time.

- **Block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.

Stream Cipher



Block Cipher

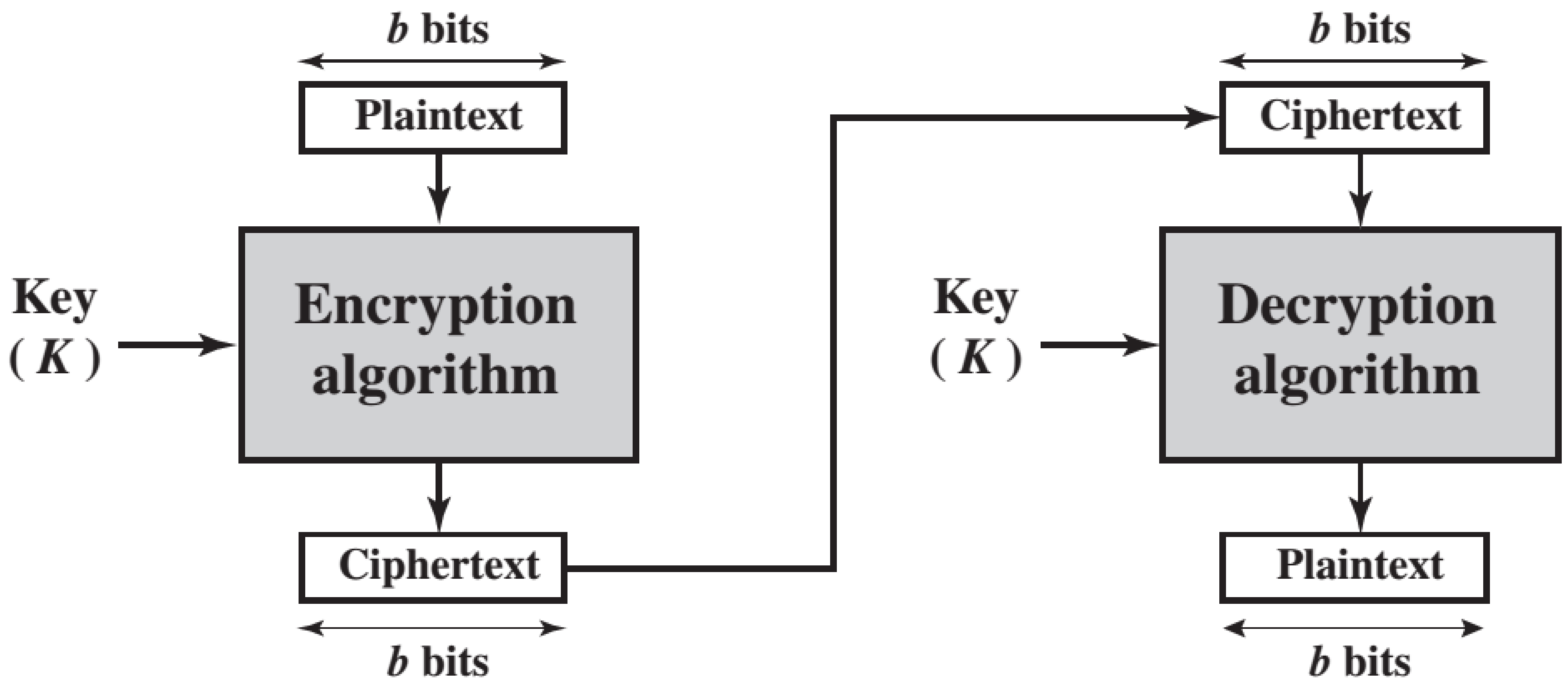


Table of Contents

Stream Ciphers and Block Ciphers

Data Encryption Standard

DES Algorithm

DES Key Creation

DES Encryption

The Strength Of DES

Data Encryption Standard

- ❑ Data Encryption Standard is a symmetric-key algorithm for the encryption of electronic data.
- ❑ Developed in the early 1970s at IBM and based on an earlier design by Horst Feistel.
- ❑ DES was issued in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46

Data Encryption Standard

- DES, data are encrypted in 64-bit blocks using a 56 bits (+8 parity bits) key.
- The algorithm transforms 64-bit input in a series of steps into a 64-bit output.
- The same steps, with the same key, are used to reverse the encryption.

Data Encryption Standard

- DES uses "keys" where are also apparently 16 hexadecimal numbers long, or apparently 64 bits long. However, every 8th key bit is ignored in the DES algorithm, so that the effective key size is 56 bits.

Table of Contents

- Stream Ciphers and Block Ciphers
- Data Encryption Standard
- DES Algorithm
- DES Key Creation
- DES Encryption
- The Strength Of DES

DES Algorithm

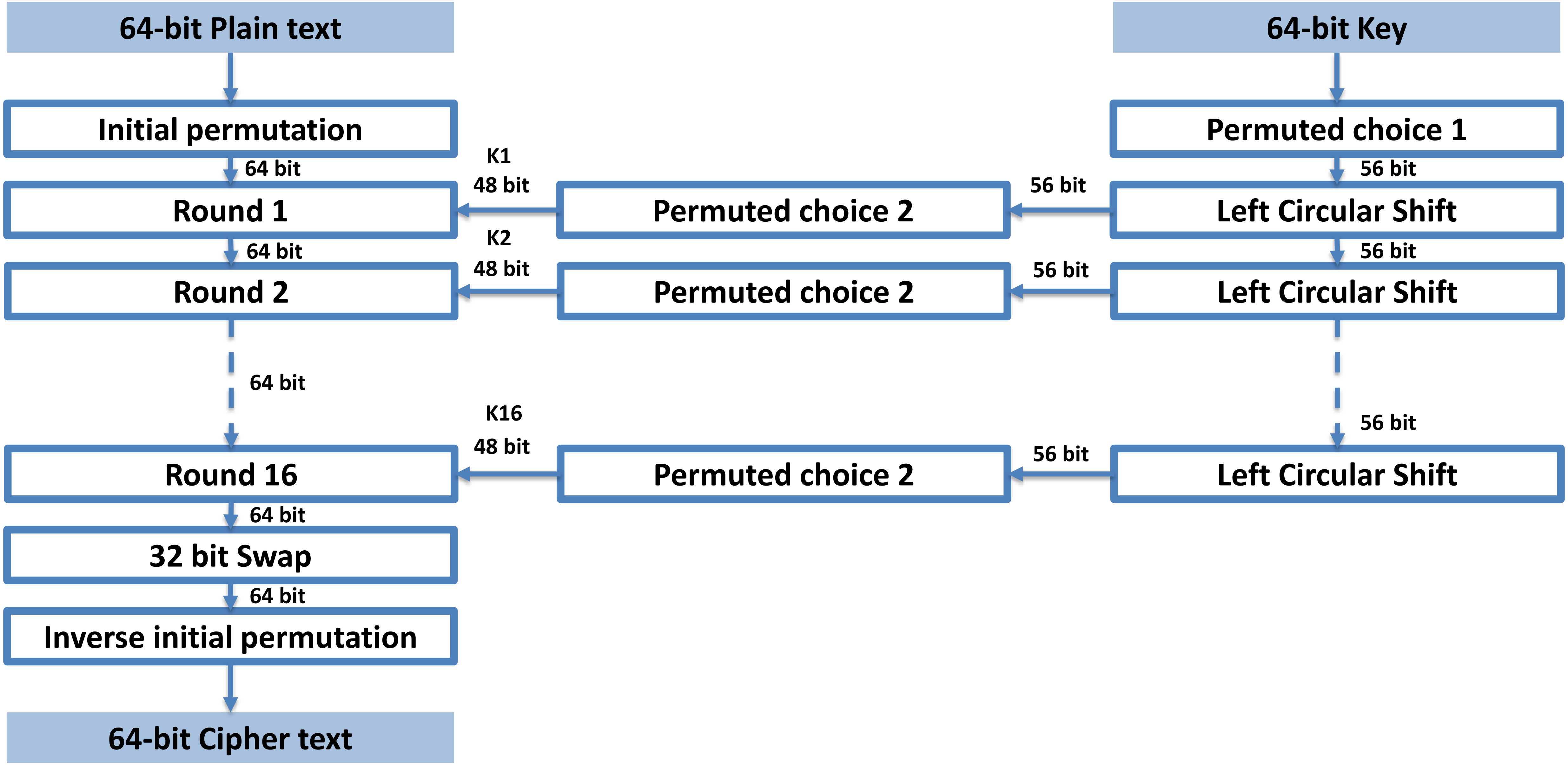


Table of Contents

Stream Ciphers and Block Ciphers

Data Encryption Standard

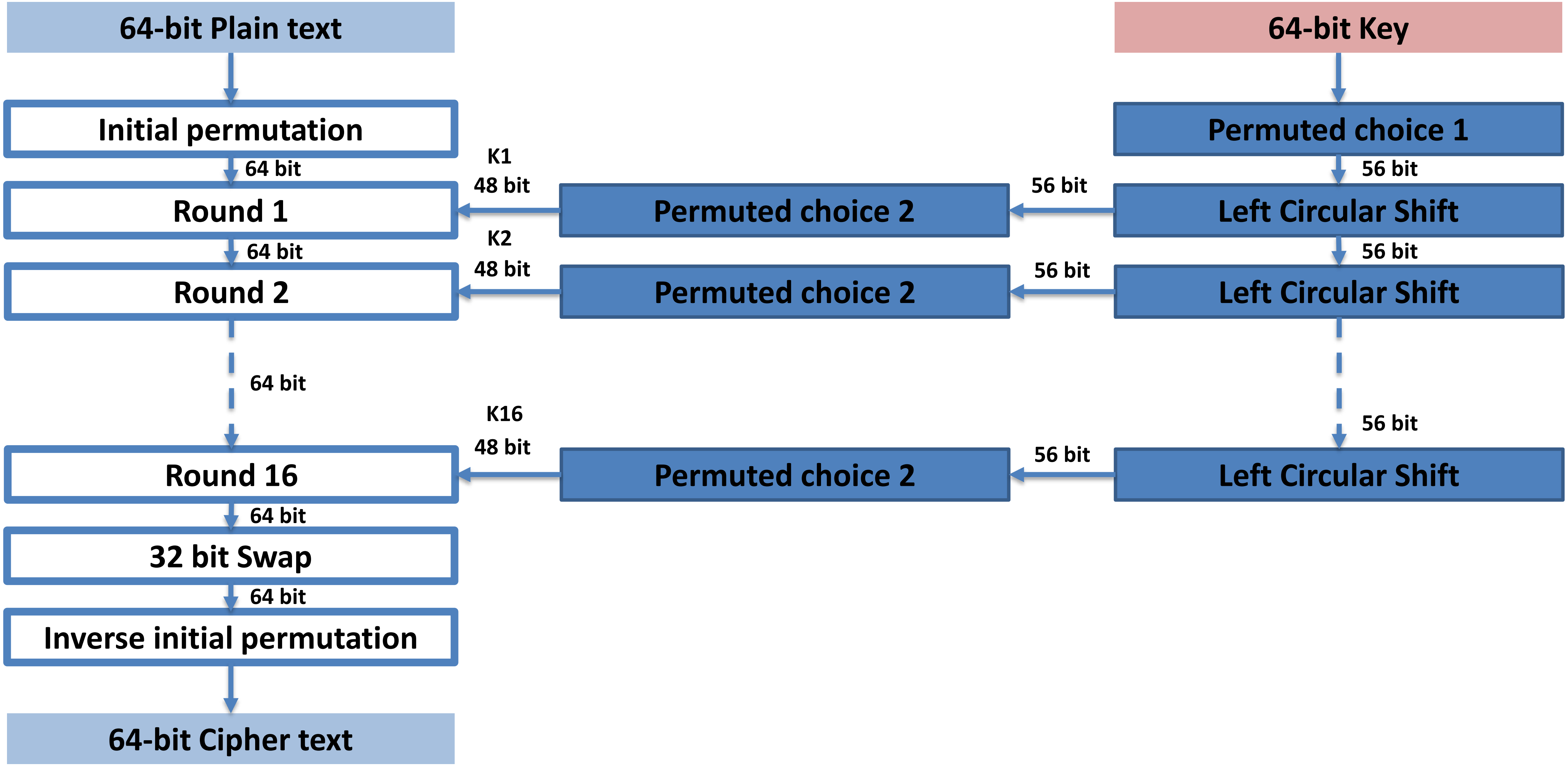
DES Algorithm

DES Key Creation

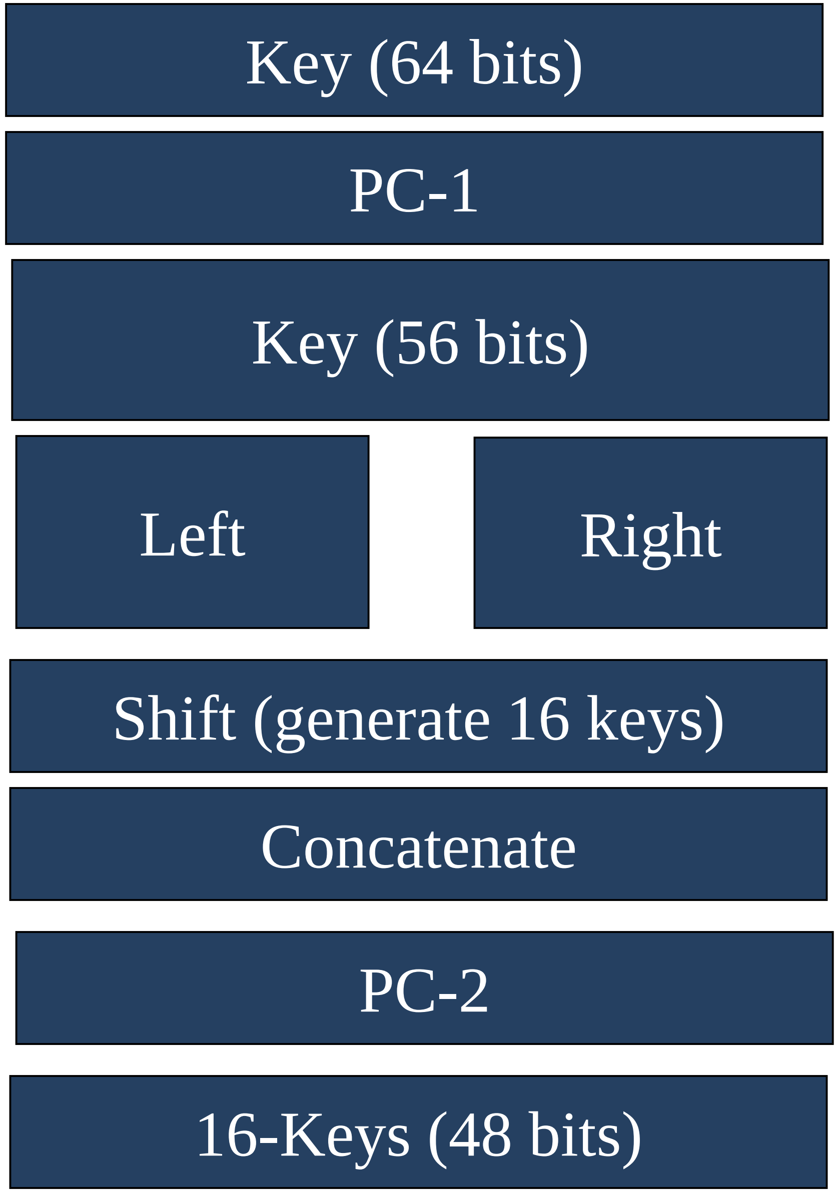
DES Encryption

The Strength Of DES

DES Algorithm



DES Key Creation



Key (64 bits)

PC-1

Key (56 bits)

Left

Right

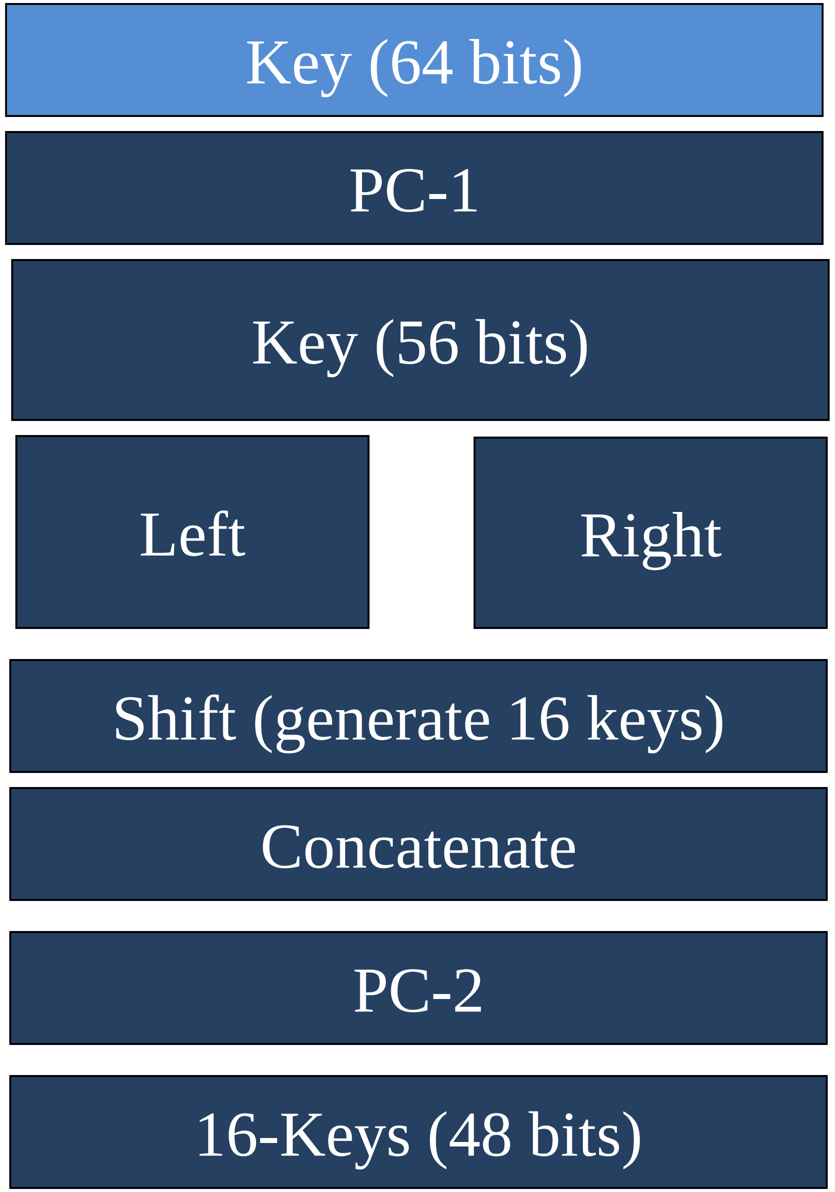
Shift (generate 16 keys)

Concatenate

PC-2

16-Keys (48 bits)

DES Key Creation



Key (64 bits)

PC-1

Key (56 bits)

Left

Right

Shift (generate 16 keys)

Concatenate

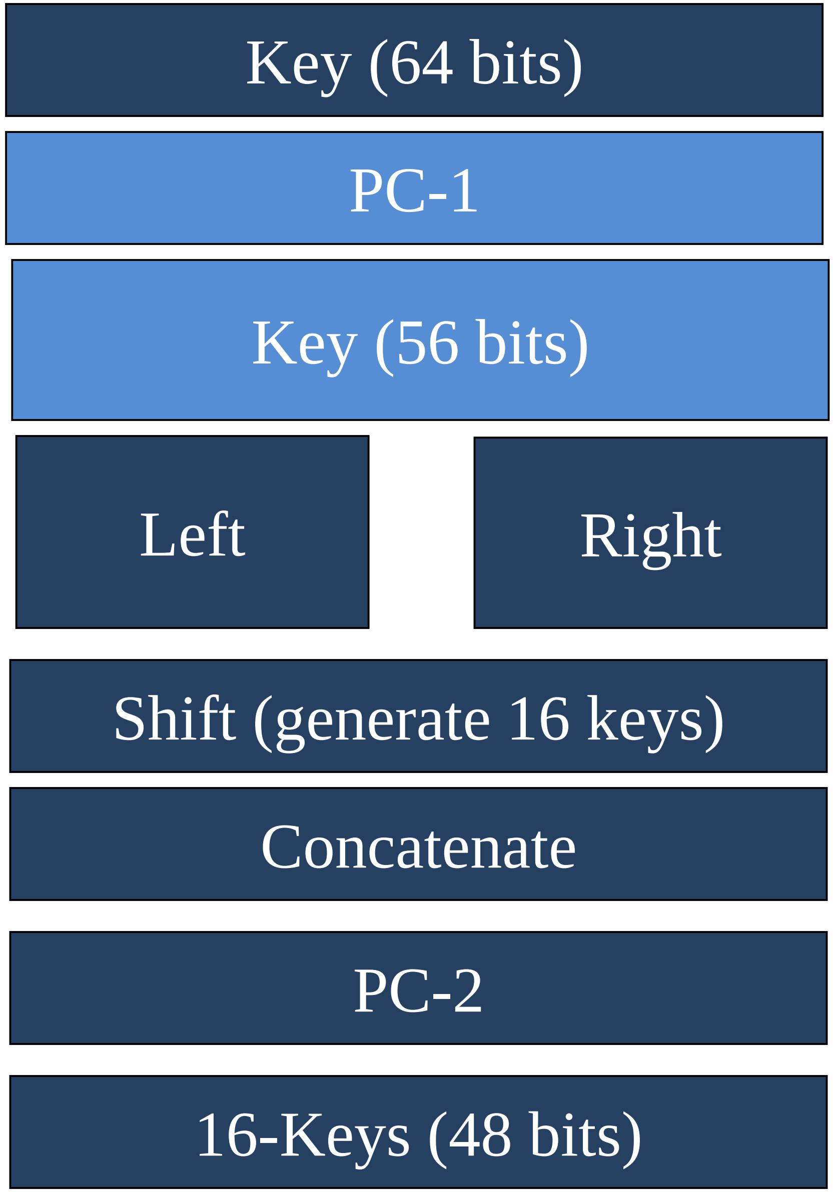
PC-2

16-Keys (48 bits)

DES Key Creation

- ❑ Let K be the hexadecimal key K = 133457799BBCDFF1
- ❑ K (binary) = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001
- ❑ K=64bit
- ❑ The DES algorithm uses the following steps:

DES Key Creation



Key (64 bits)

PC-1

Key (56 bits)

Left

Right

Shift (generate 16 keys)

Concatenate

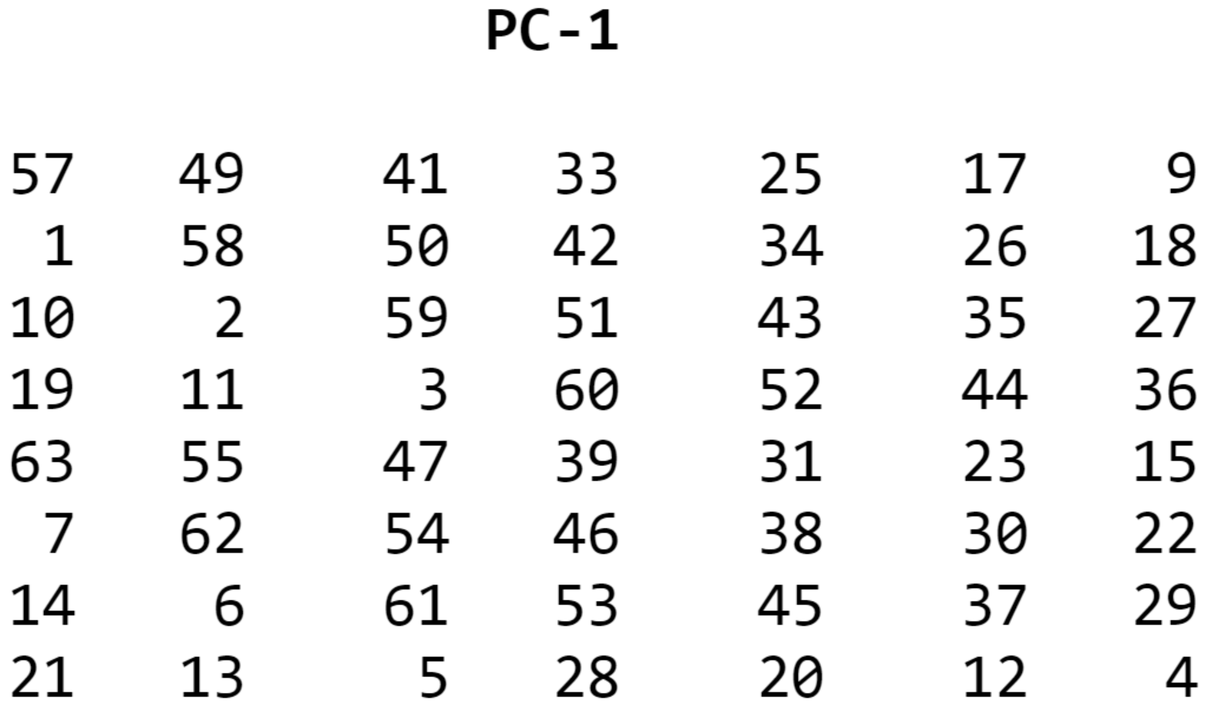
PC-2

16-Keys (48 bits)

DES Key Creation

1) **Step 1: Apply** permutation choice -1 **(PC-1)**

- The 64-bit key is permuted according to the following table, **PC-1**



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

DES Key Creation

- ❑ K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

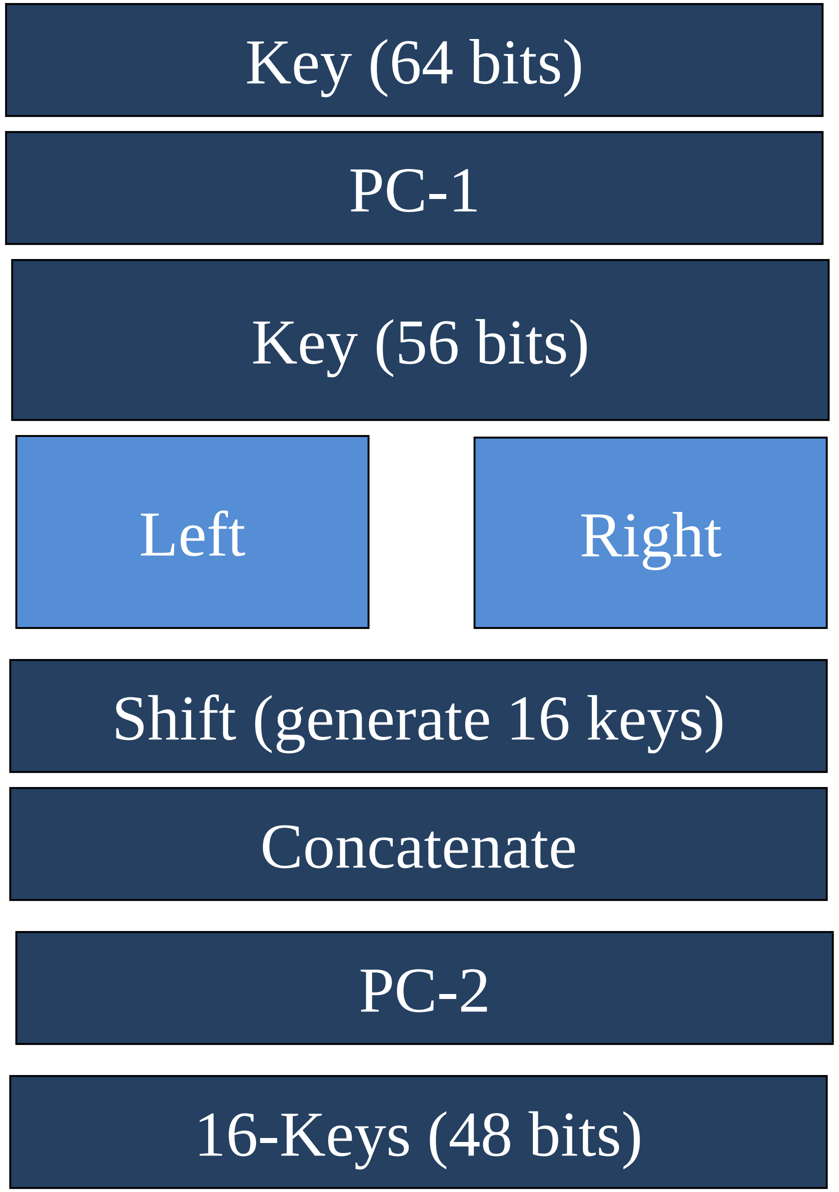
PC-1

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

- ❑ we get the 56-bit permutation

- ❑ Kp = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

DES Key Creation



Key (64 bits)

PC-1

Key (56 bits)

Left

Right

Shift (generate 16 keys)

Concatenate

PC-2

16-Keys (48 bits)

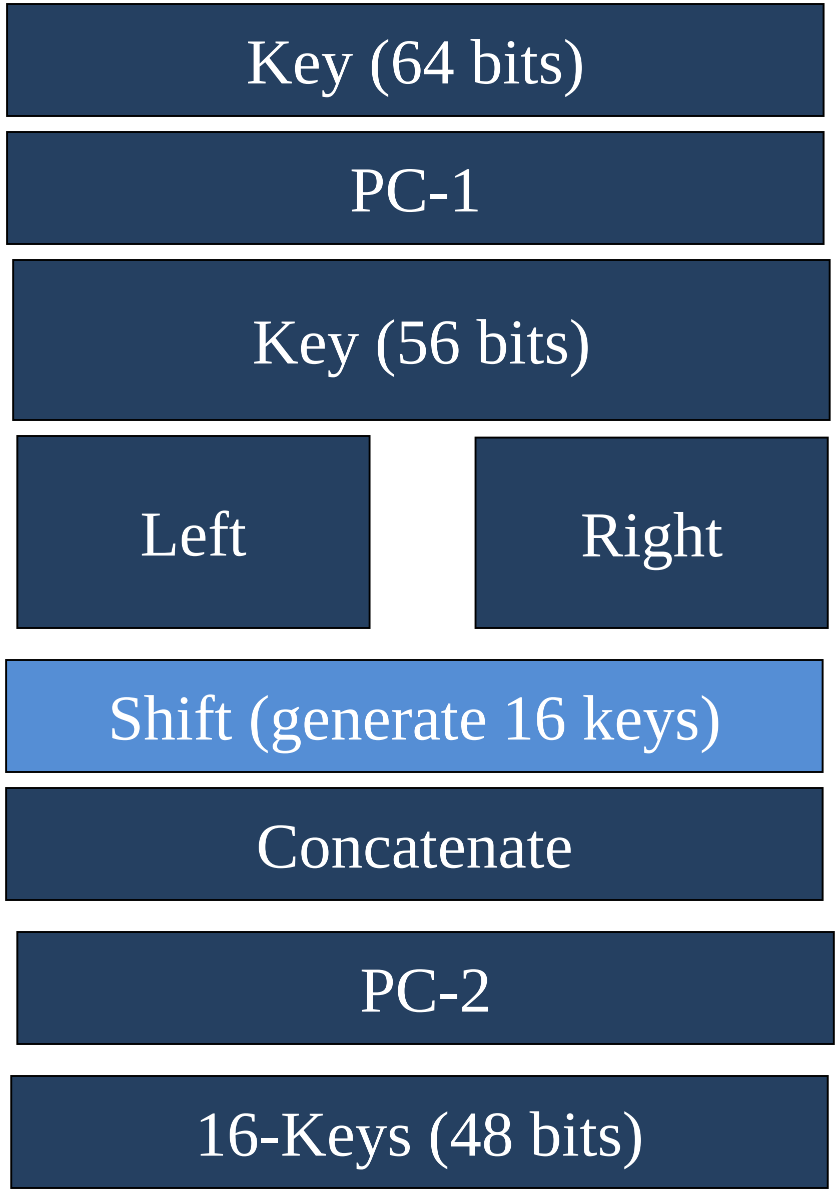
DES Key Creation

2) Split Kp key into left and right halves

- Kp = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

- K_L = 1111000 0110011 0010101 0101111
- K_R = 0101010 1011001 1001111 0001111

DES Key Creation



Key (64 bits)

PC-1

Key (56 bits)

Left

Right

Shift (generate 16 keys)

Concatenate

PC-2

16-Keys (48 bits)

DES Key Creation

2) Apply Shifts (Left) as describe on table

- ❑ K_L = 1111000 0110011 0010101 0101111
- ❑ K_R = 0101010 1011001 1001111 0001111
- ❑ K_{L1} =Shift(K_L)= 1110000110011001010101011111
- ❑ K_{R1} =Shift(K_R)= 1010101011001100111100011110
- ❑ K_{L2} =Shift(K_{L1})= 1100001100110010101010111111
- ❑ K_{R2} =Shift(K_{R1})= 0101010110011001111000111101
- ❑ K_{L3} =Shift(K_{L2})= 0000110011001010101011111111
- ❑ K_{R3} =Shift(K_{R2})= 0101011001100111100011110101

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

DES Key Creation

2) Apply Shifts (Left) as describe on table

- ❑ K_{L4} = 0011001100101010101111111100
- ❑ K_{R4} = 0101100110011110001111010101
- ❑ K_{L5} = 1100110010101010111111110000
- ❑ K_{R5} = 0110011001111000111101010101
- ❑ K_{L6} = 001100101010101111111100001
- ❑ K_{R6} = 1001100111100011110101010101
- ❑ K_{L7} = 1100101010101111111100001100
- ❑ K_{R7} = 0110011110001111010101010110

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

DES Key Creation

2) Apply Shifts (Left) as describe on table

- ❑ K_{L8} = 0010101010111111110000110011
- ❑ K_{R8} = 1001111000111101010101011001
- ❑ K_{L9} = 0101010101111111100001100110
- ❑ K_{R9} = 0011110001111010101010110011
- ❑ K_{L10} = 0101010111111110000110011001
- ❑ K_{R10} = 1111000111101010101011001100
- ❑ K_{L11} = 0101011111111000011001100101
- ❑ K_{R11} = 1100011110101010101100110011

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

DES Key Creation

2) Apply Shifts (Left) as describe on table

- ❑ K_{L12} = 0101111111100001100110010101
- ❑ K_{R12} = 0001111010101010110011001111
- ❑ K_{L13} = 0111111110000110011001010101
- ❑ K_{R13} = 0111101010101011001100111100
- ❑ K_{L14} = 1111111000011001100101010101
- ❑ K_{R14} = 1110101010101100110011110001
- ❑ K_{L15} = 1111100001100110010101010111
- ❑ K_{R15} = 1010101010110011001111000111

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

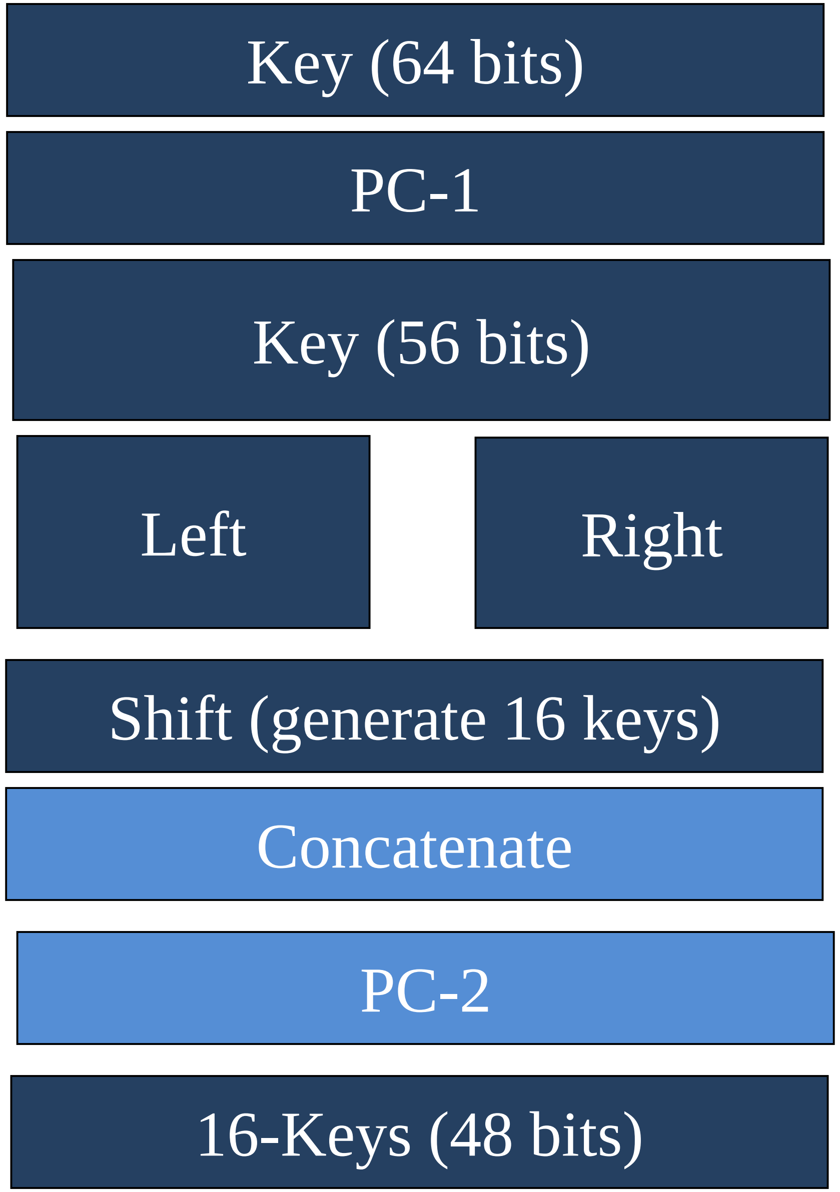
DES Key Creation

2) Apply Shifts (Left) as describe on table

- ❑ K_{L16} = 1111000011001100101010101111
- ❑ K_{R16} = 0101010101100110011110001111

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

DES Key Creation



Key (64 bits)

PC-1

Key (56 bits)

Left

Right

Shift (generate 16 keys)

Concatenate

PC-2

16-Keys (48 bits)

DES Key Creation

3) Concatenate K_L and K_R

- *K1=1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110*

4) Apply PC-2 to all 16 keys

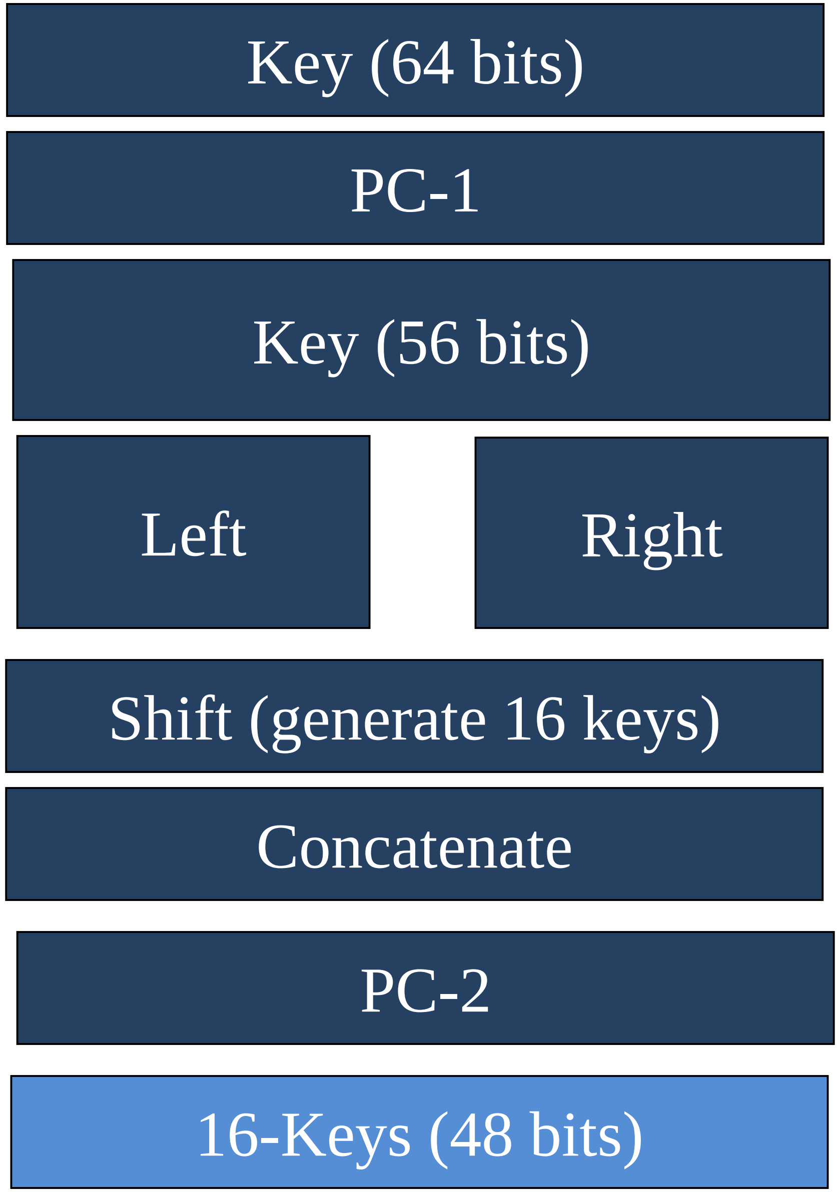
- *we get the 48-bit permutation for each key*

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

- *K1=000110 110000 001011 101111 111111 000111 000001 110010*

DES Key Creation



Key (64 bits)

PC-1

Key (56 bits)

Left

Right

Shift (generate 16 keys)

Concatenate

PC-2

16-Keys (48 bits)

DES Key Creation

□ K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

□ 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

DES Key Creation

- 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 Algorithm

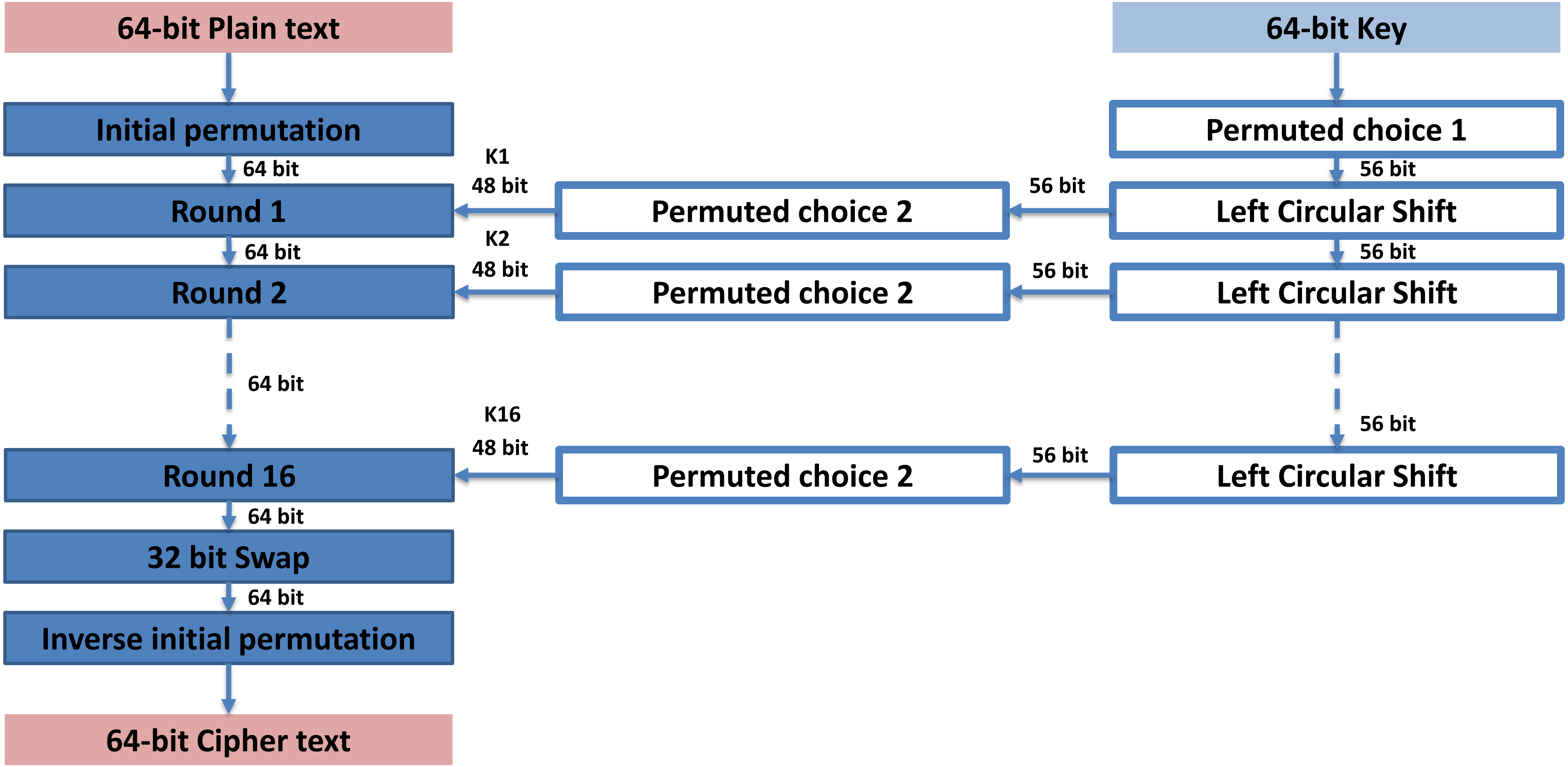


Table of Contents

Stream Ciphers and Block Ciphers

Data Encryption Standard

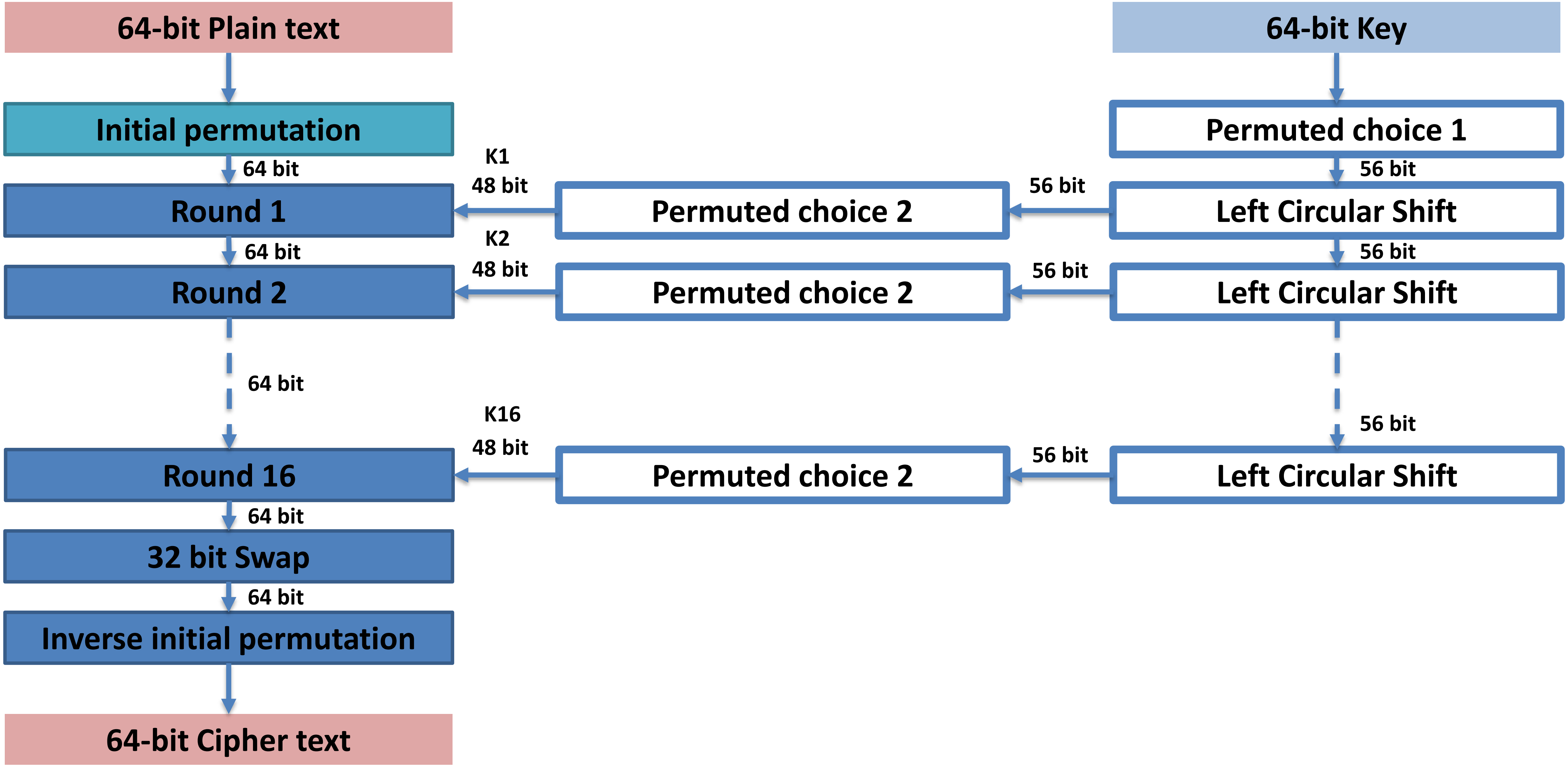
DES Algorithm

DES Key Creation

DES Encryption

The Strength Of DES

DES Algorithm



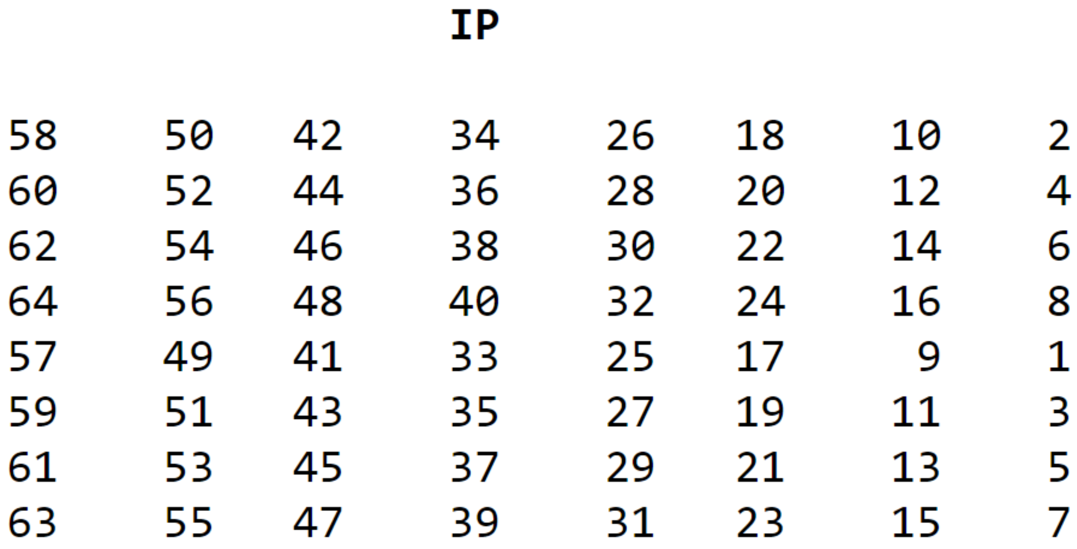
DES Encryption

- Assume: M = 0123456789ABCDEF

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

DES Encryption

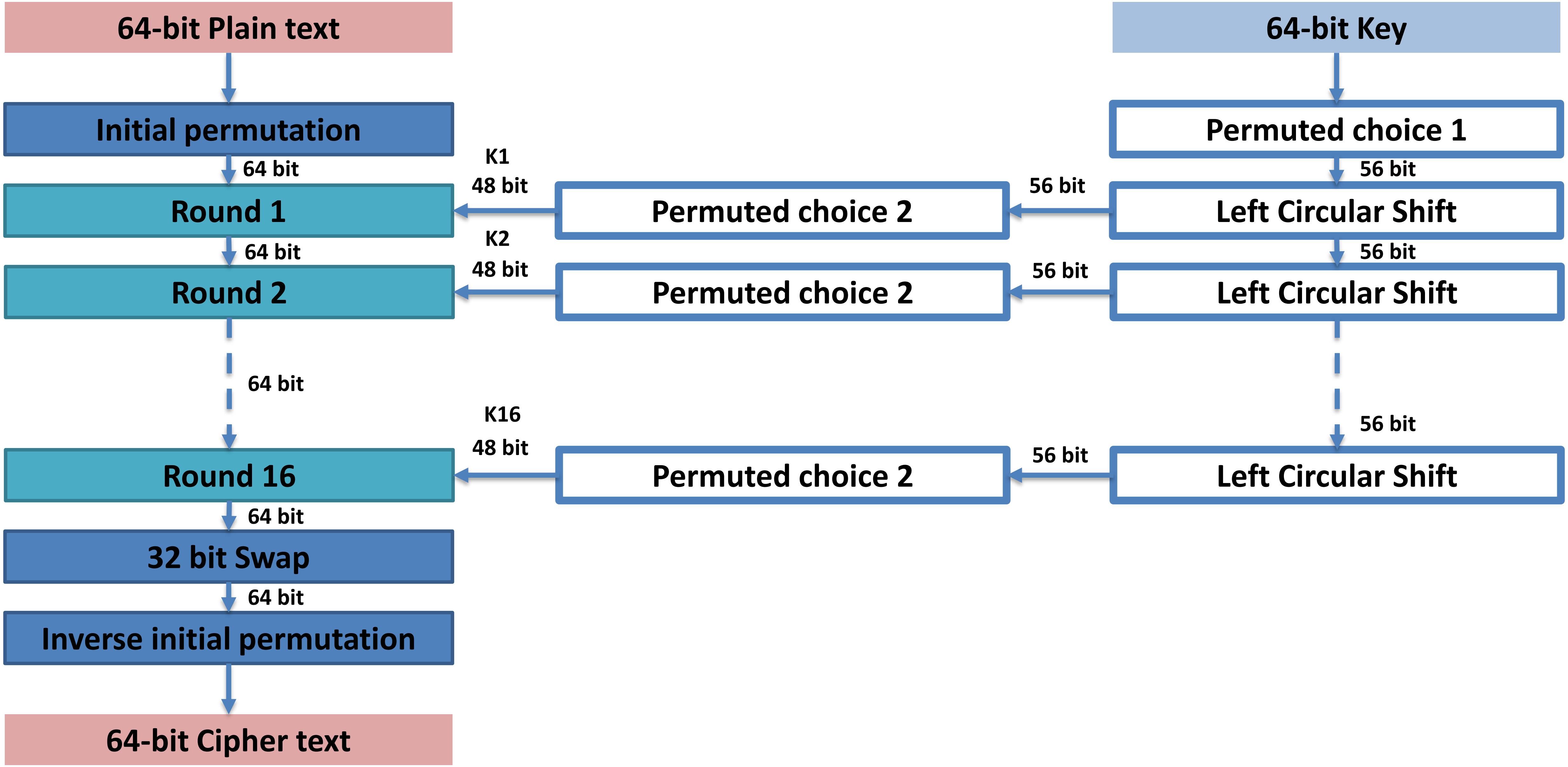
- ❑ M=0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
- ❑ Applying the initial permutation to the block of text P (64bit).
- ❑ IP=1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010



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

DES Algorithm

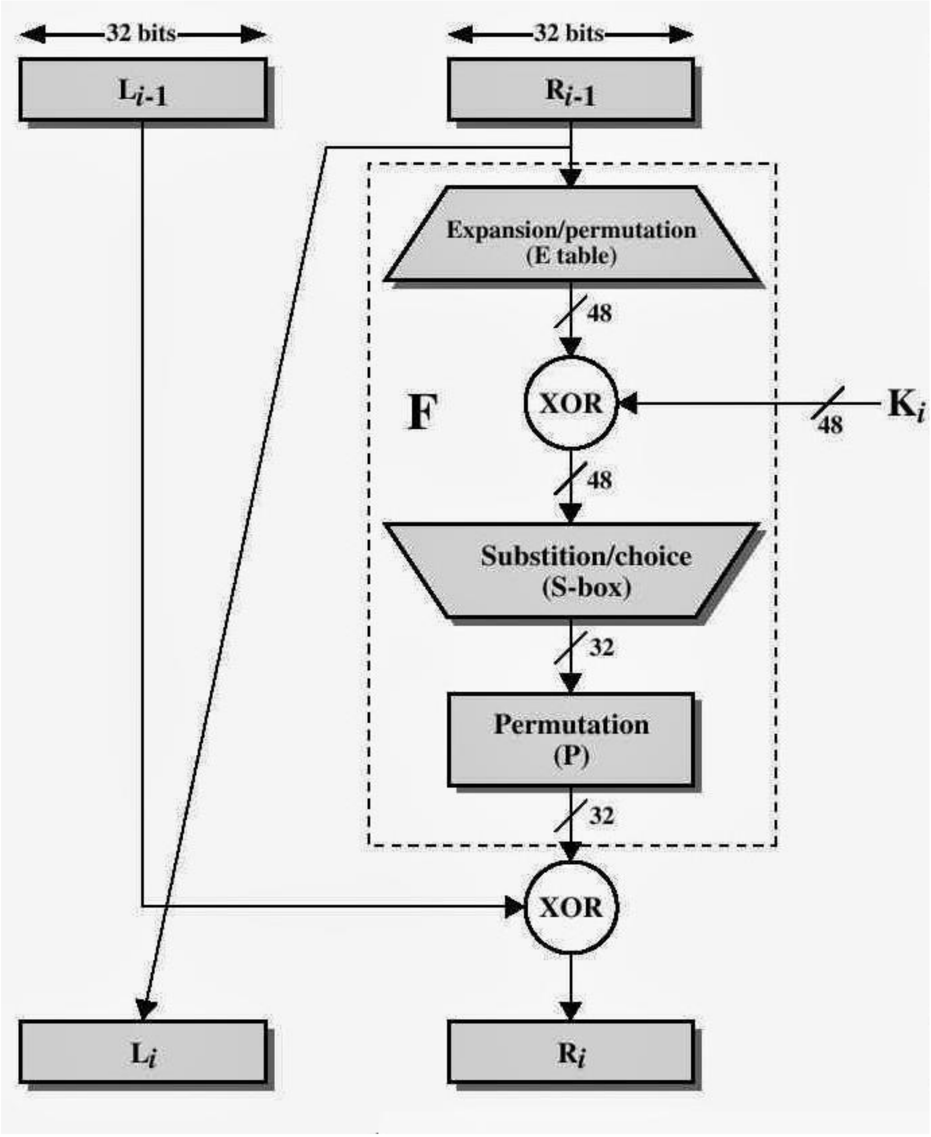


DES Encryption

- Divide the permuted block **IP** into a left half L_0 of 32 bits, and a right half R_0 of 32 bits.
- L_0 = 1100 1100 0000 0000 1100 1100 1111 1111
- R_0 = 1111 0000 1010 1010 1111 0000 1010 1010

- $L_n = R_{n-1}$
- $R_n = L_{n-1} \oplus f(R_{n-1},K_n)$

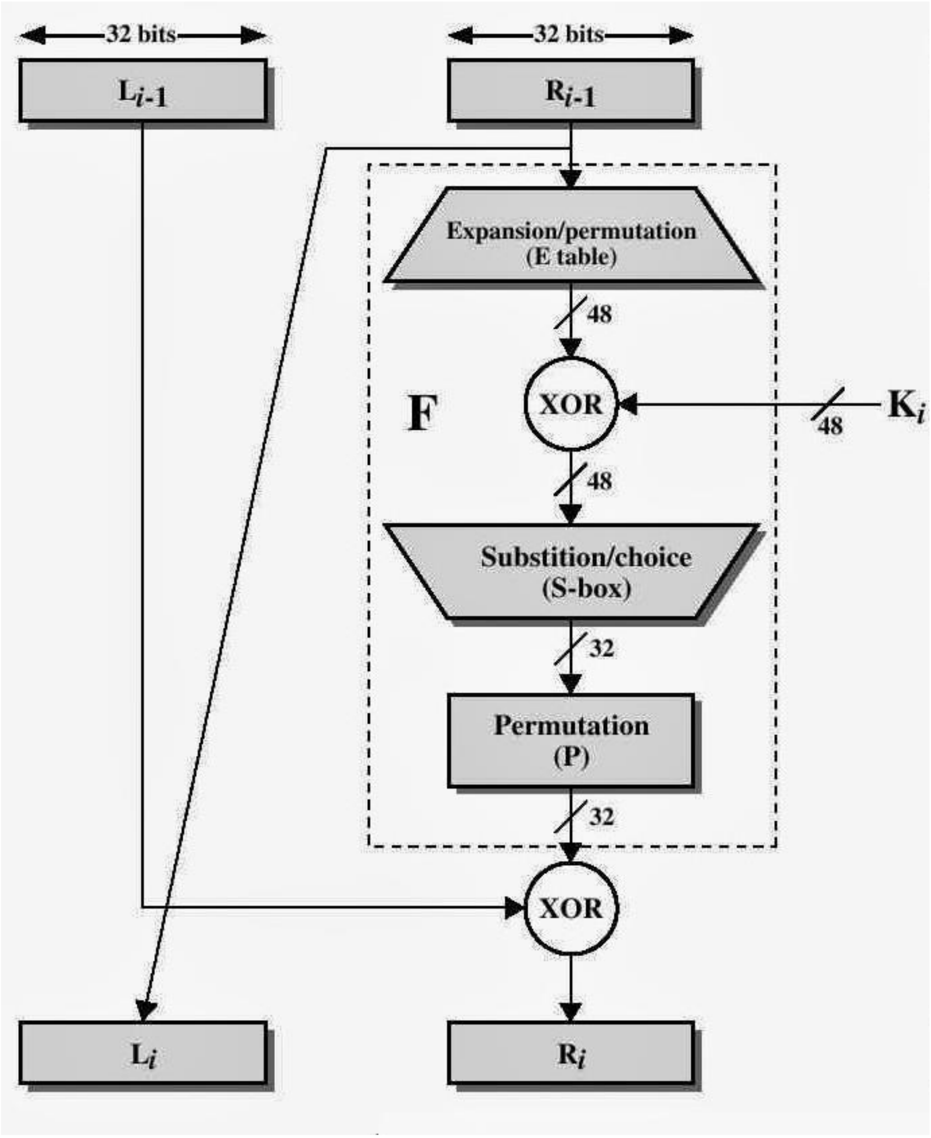
DES Encryption



DES Encryption

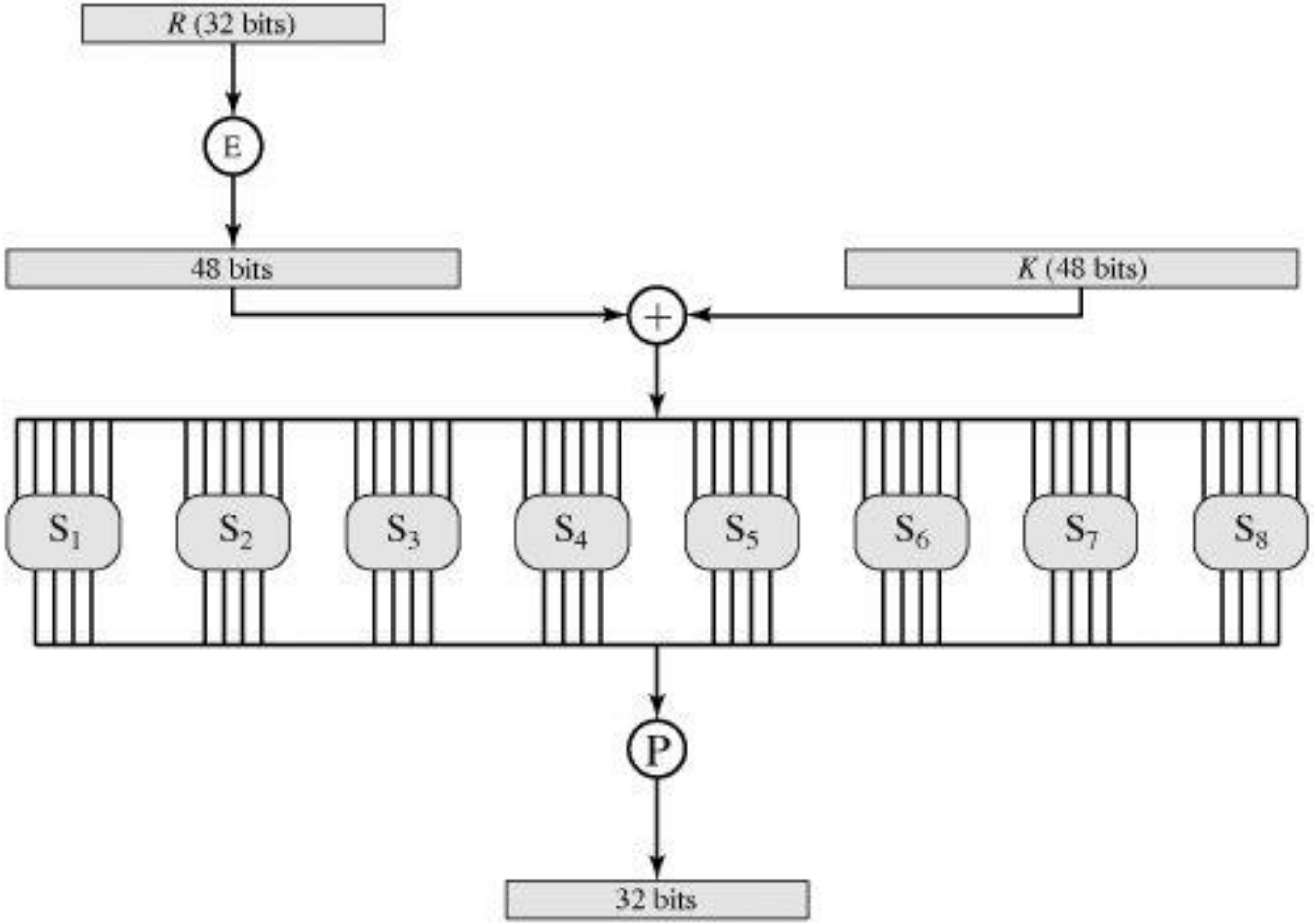
- **Example:** For $n = 1$, we have
- $L_0 = 1100\ 1100\ 0000\ 0000\ 1100\ 1100\ 1111\ 1111$
- $R_0 = 1111\ 0000\ 1010\ 1010\ 1111\ 0000\ 1010\ 1010$
- $K_1 = 000110\ \ 110000\ \ 001011\ \ 101111\ \ 111111\ \ 000111\ \ 000001\ \ 110010$
- $L_1 = R_0 = 1111\ 0000\ 1010\ 1010\ 1111\ 0000\ 1010\ 1010$
- $R_1 = L_0 \oplus f(R_0, K_1)$ K=48bit but R_0=32bit problem

DES Encryption



DES Encryption

- *Calculate $f(R_0, K_1)$*



DES Encryption

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

E BIT-SELECTION TABLE

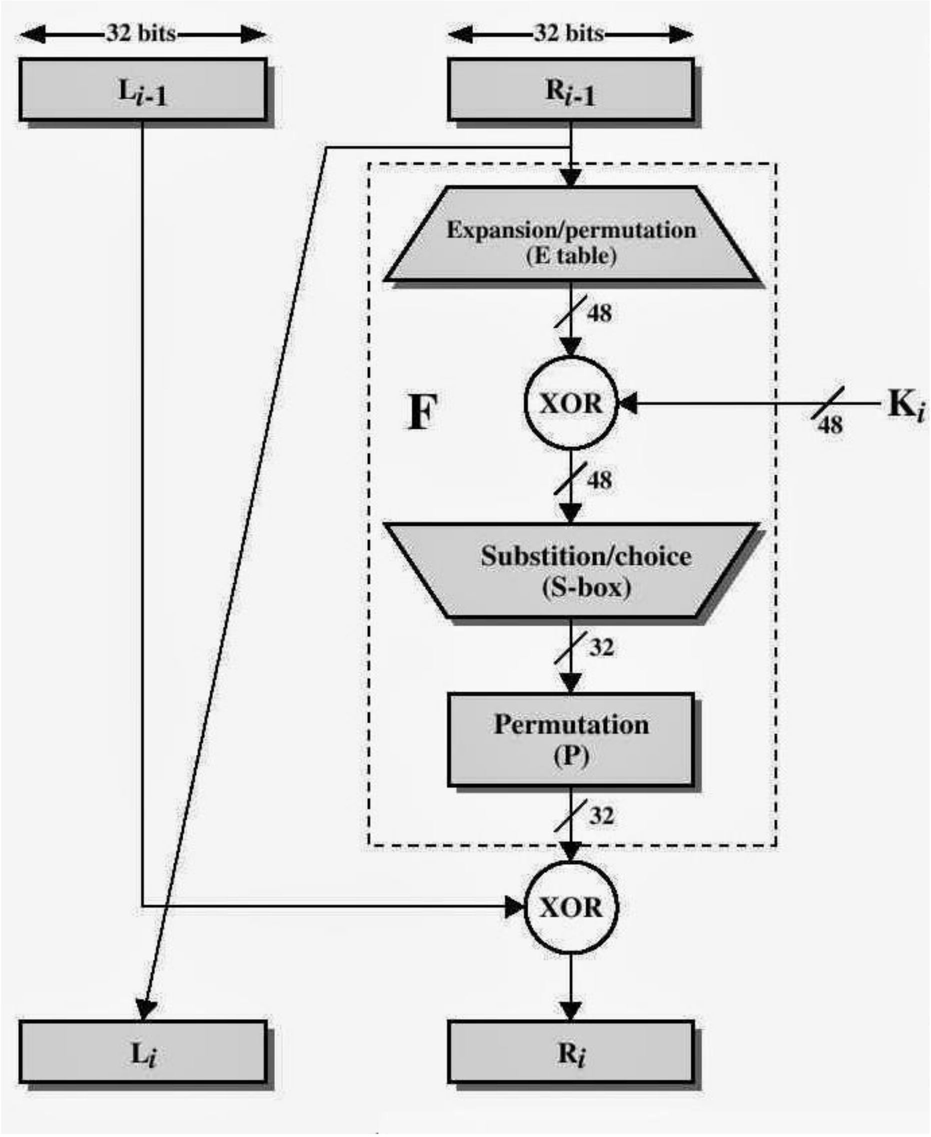
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

- R_0 = 1111 0000 1010 1010 1111 0000 1010 1010
- $E(R_0)$ = 011110 100001 010101 010101 011110 100001 010101 010101

DES Encryption

- $R_1 = L_0 \oplus f(R_0, K_1)$
- K_1 = 000110 110000 001011 101111 111111 000111 000001 110010
- $E(R_0)$ = 011110 100001 010101 010101 011110 100001 010101 010101
- $K_1 \oplus E(R_0)$ = 011000 010001 011110 111010 100001 100110 010100 100111

DES Encryption

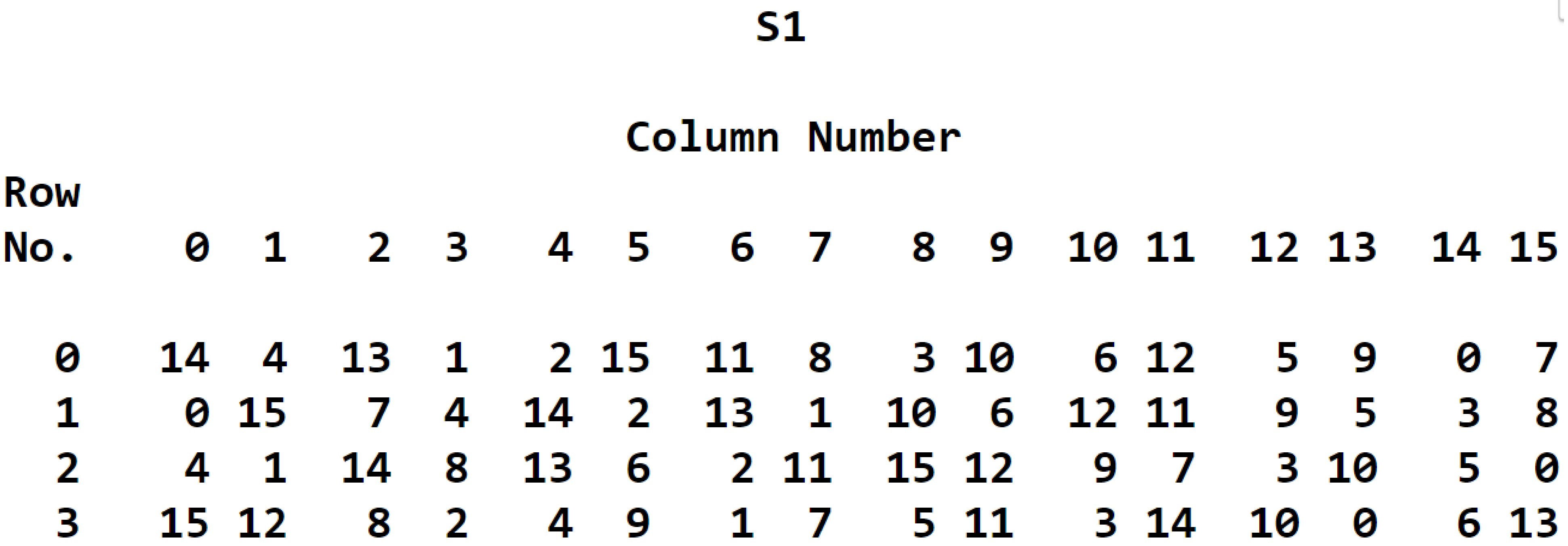


DES Encryption

- ❑ We have not yet finished calculating the function *f*
- ❑ We now have 48 bits, or eight groups of six bits. We now do something strange with each group of six bits: we use them as addresses in tables called "**S boxes**".
- ❑ $f(K_n , \mathrm{E}(R_{n-1})) = B_1B_2B_3B_4B_5B_6B_7B_8$
- ❑ where each B_i is a group of six bits. We now calculate
- ❑ $S_1(B_1)\ \ S_2(B_2)\ \ S_3(B_3)\ \ S_4(B_4)\ \ S_5(B_5)\ \ S_6(B_6)\ \ S_7(B_7)\ \ S_8(B_8)$

DES Encryption

- ❑ The net result is that the eight groups of 6 bits are transformed into eight groups of 4 bits (the 4-bit outputs from the S boxes) for 32 bits total.



S1

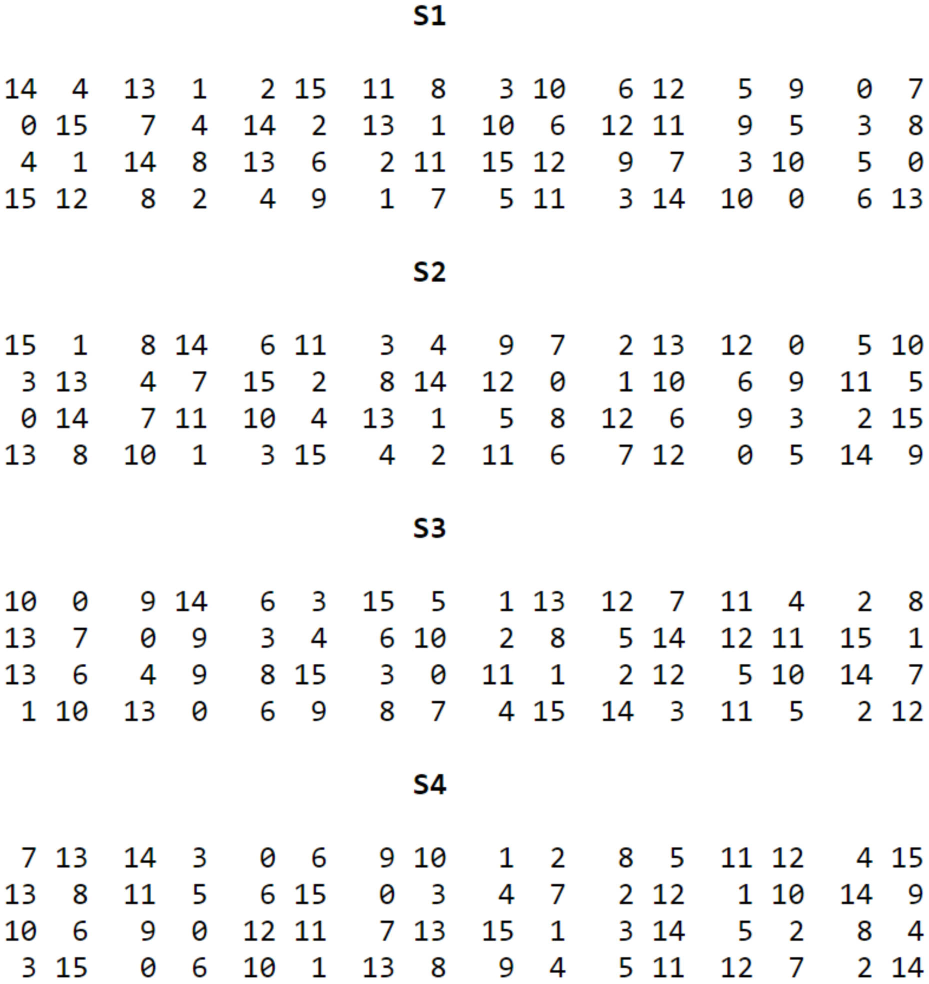
Column Number

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

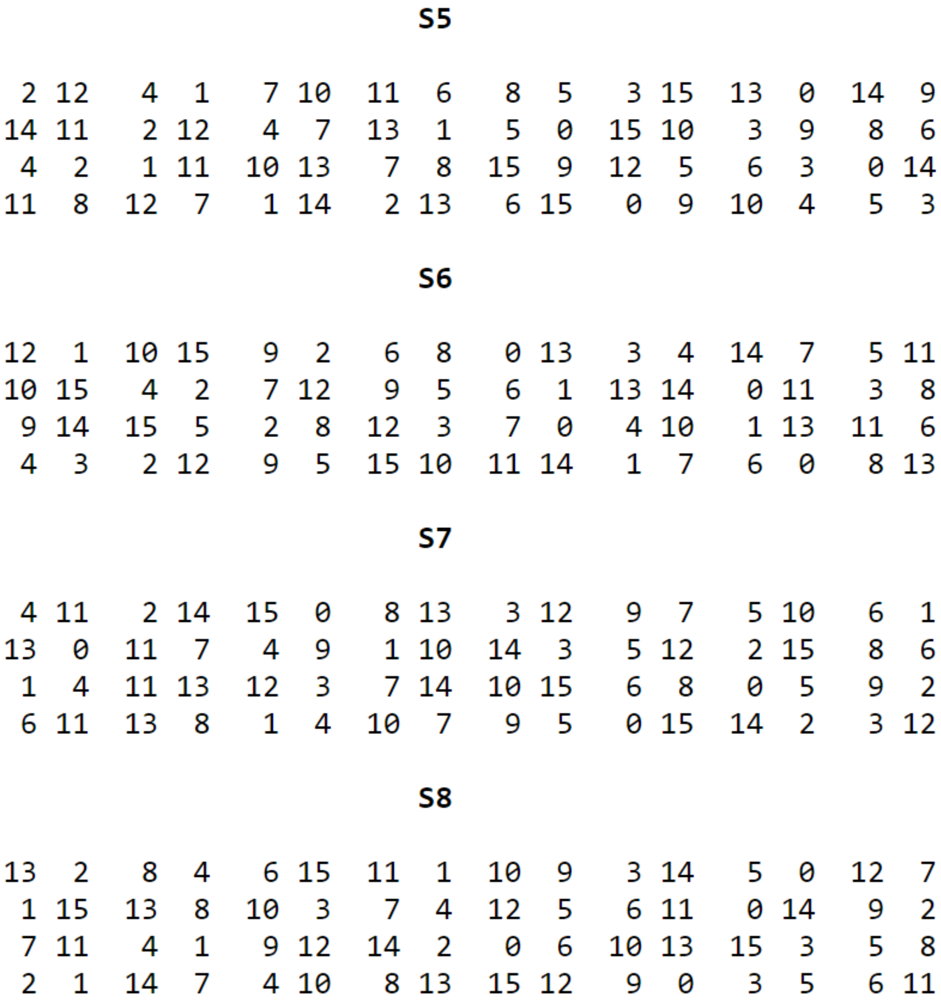
DES Encryption

- For example, for input block B = 011011 the first bit is "0" and the last bit "1" giving 01 as the row. This is row 1. The middle four bits are "1101". This is the binary equivalent of decimal 13, so the column is column number 13. In row 1, column 13 appears 5. This determines the output; 5 is binary 0101, so that the output is 0101. Hence S_1(011011) = 0101.

DES Encryption



```
                          S1

14  4   13  1    2 15   11  8    3 10    6 12    5  9    0  7
 0 15    7  4   14  2   13  1   10  6   12 11    9  5    3  8
 4  1   14  8   13  6    2 11   15 12    9  7    3 10    5  0
15 12    8  2    4  9    1  7    5 11    3 14   10  0    6 13

                          S2

15  1    8 14    6 11    3  4    9  7    2 13   12  0    5 10
 3 13    4  7   15  2    8 14   12  0    1 10    6  9   11  5
 0 14    7 11   10  4   13  1    5  8   12  6    9  3    2 15
13  8   10  1    3 15    4  2   11  6    7 12    0  5   14  9

                          S3

10  0    9 14    6  3   15  5    1 13   12  7   11  4    2  8
13  7    0  9    3  4    6 10    2  8    5 14   12 11   15  1
13  6    4  9    8 15    3  0   11  1    2 12    5 10   14  7
 1 10   13  0    6  9    8  7    4 15   14  3   11  5    2 12

                          S4

 7 13   14  3    0  6    9 10    1  2    8  5   11 12    4 15
13  8   11  5    6 15    0  3    4  7    2 12    1 10   14  9
10  6    9  0   12 11    7 13   15  1    3 14    5  2    8  4
 3 15    0  6   10  1   13  8    9  4    5 11   12  7    2 14
```



```
                          S5

 2 12    4  1    7 10   11  6    8  5    3 15   13  0   14  9
14 11    2 12    4  7   13  1    5  0   15 10    3  9    8  6
 4  2    1 11   10 13    7  8   15  9   12  5    6  3    0 14
11  8   12  7    1 14    2 13    6 15    0  9   10  4    5  3

                          S6

12  1   10 15    9  2    6  8    0 13    3  4   14  7    5 11
10 15    4  2    7 12    9  5    6  1   13 14    0 11    3  8
 9 14   15  5    2  8   12  3    7  0    4 10    1 13   11  6
 4  3    2 12    9  5   15 10   11 14    1  7    6  0    8 13

                          S7

 4 11    2 14   15  0    8 13    3 12    9  7    5 10    6  1
13  0   11  7    4  9    1 10   14  3    5 12    2 15    8  6
 1  4   11 13   12  3    7 14   10 15    6  8    0  5    9  2
 6 11   13  8    1  4   10  7    9  5    0 15   14  2    3 12

                          S8

13  2    8  4    6 15   11  1   10  9    3 14    5  0   12  7
 1 15   13  8   10  3    7  4   12  5    6 11    0 14    9  2
 7 11    4  1    9 12   14  2    0  6   10 13   15  3    5  8
 2  1   14  7    4 10    8 13   15 12    9  0    3  5    6 11
```

DES Encryption

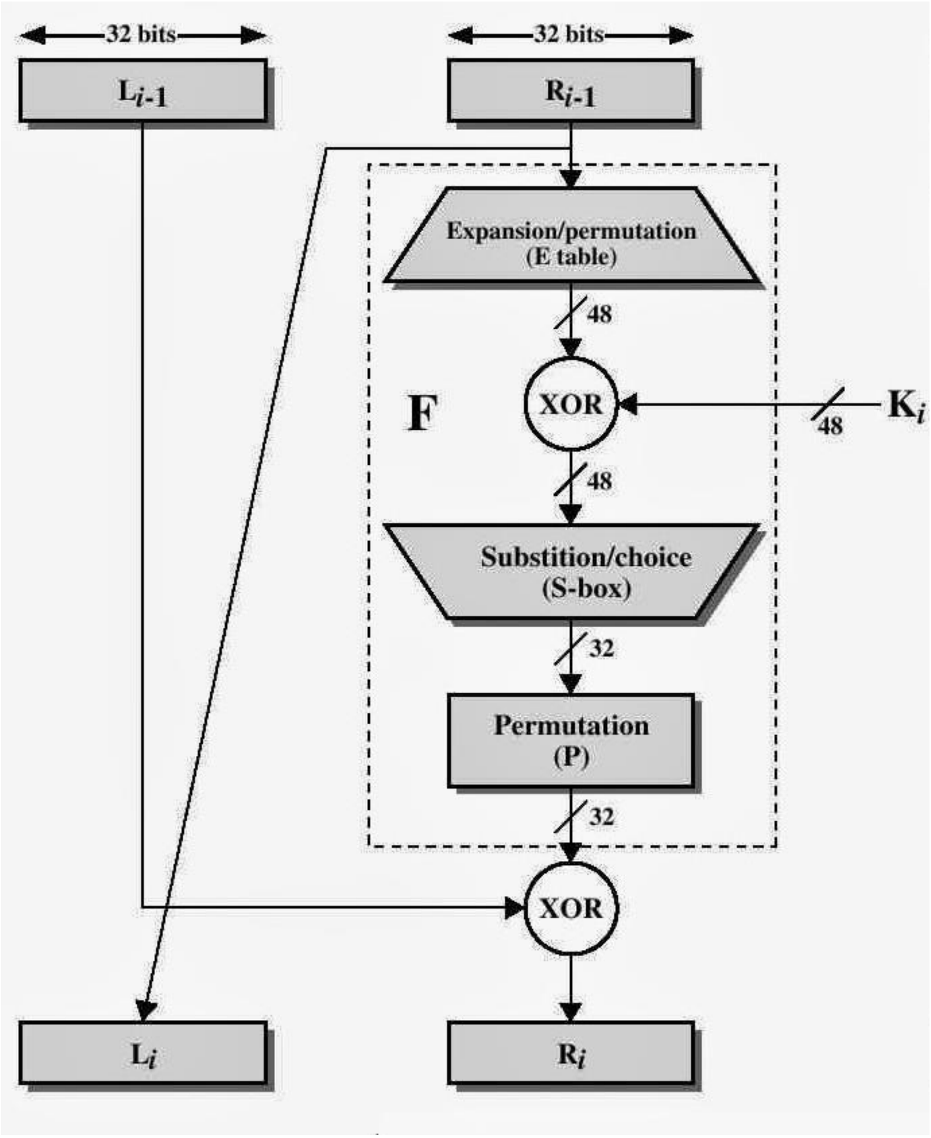
- ❑ $K_1 \oplus \mathbf{E}(R_0)$ = 011000 010001 011110 111010 100001 100110 010100 100111.

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

- ❑ The final stage in the calculation of *f* is to do a permutation **P** of the S-box output to obtain the final value of *f*:

- ❑ $f = \mathrm{P}(S_1(B_1)S_2(B_2)\ldots S_8(B_8))$

DES Encryption



DES Encryption

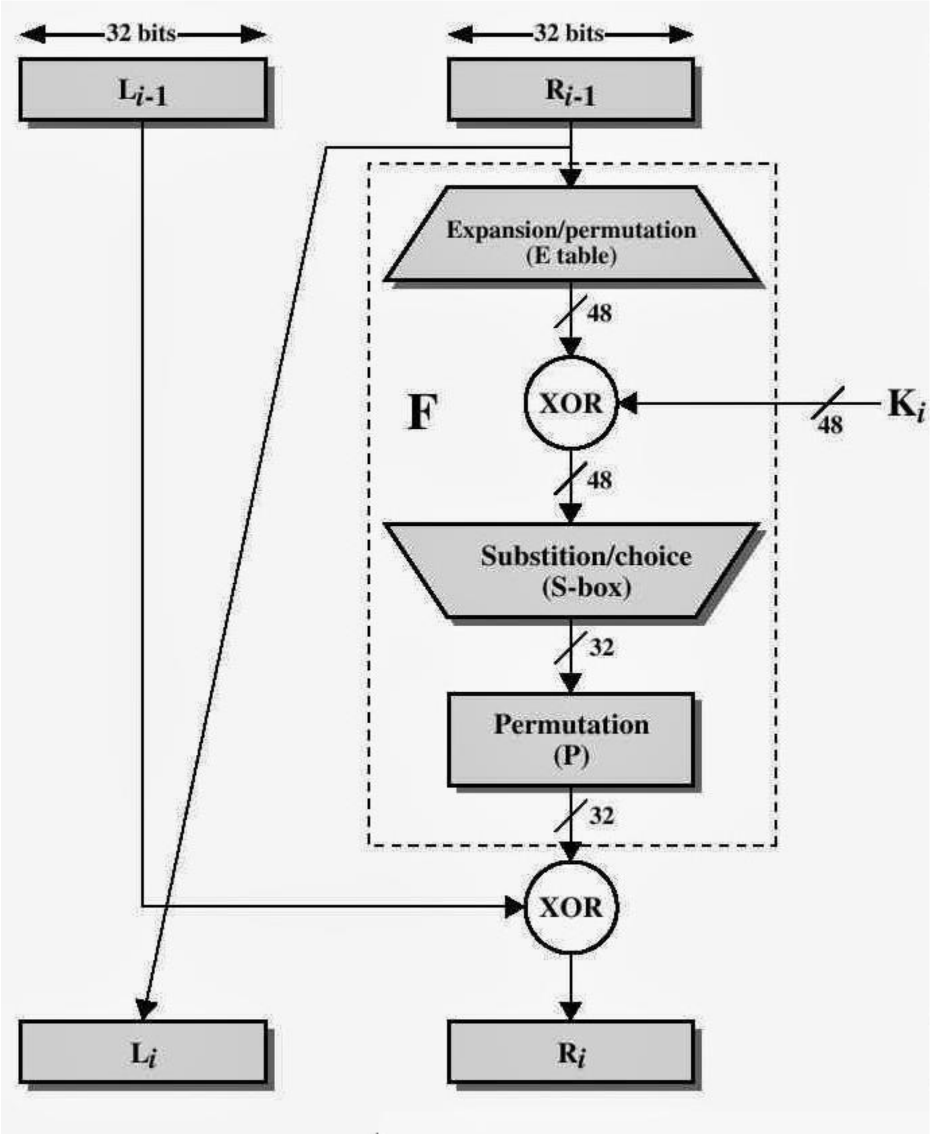
- $S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ = 0101 1100 1000 0010 1011 0101 1001 0111

- f = 0010 0011 0100 1010 1010 1001 1011 1011

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

DES Encryption



DES Encryption

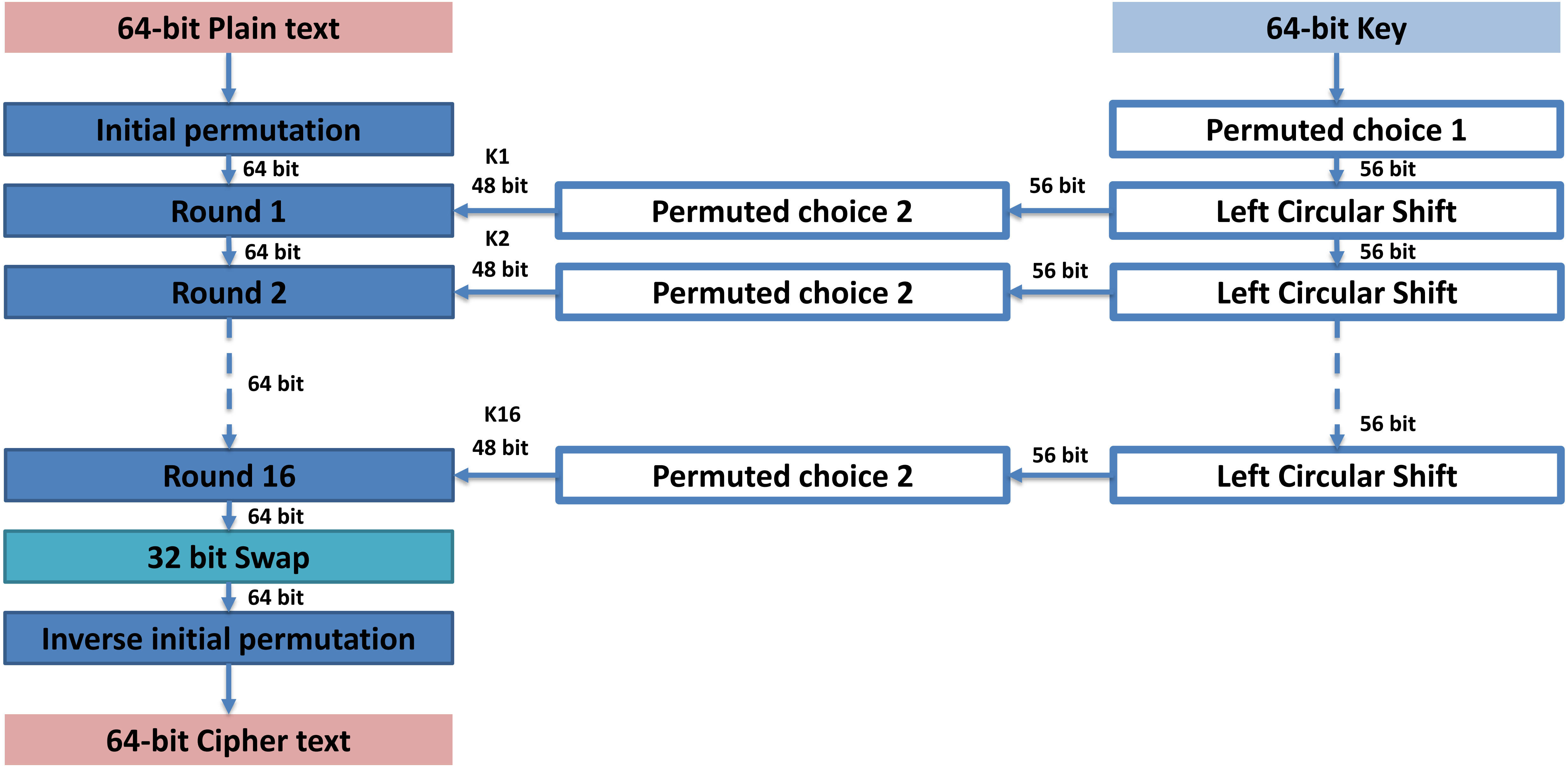
- ❑ $R_1 = L_0 \oplus f(R_0, K_1) =$

1100 1100 0000 0000 1100 1100 1111 1111

$\oplus$ 0010 0011 0100 1010 1010 1001 1011 1011

= 1110 1111 0100 1010 0110 0101 0100 0100

DES Algorithm



64-bit Plain text

Initial permutation

64 bit

Round 1

64 bit

Round 2

64 bit

Round 16

64 bit

32 bit Swap

64 bit

Inverse initial permutation

64-bit Cipher text

K1

48 bit

K2

48 bit

K16

48 bit

Permuted choice 2

Permuted choice 2

Permuted choice 2

56 bit

56 bit

56 bit

64-bit Key

Permuted choice 1

56 bit

Left Circular Shift

56 bit

Left Circular Shift

56 bit

Left Circular Shift

DES Encryption

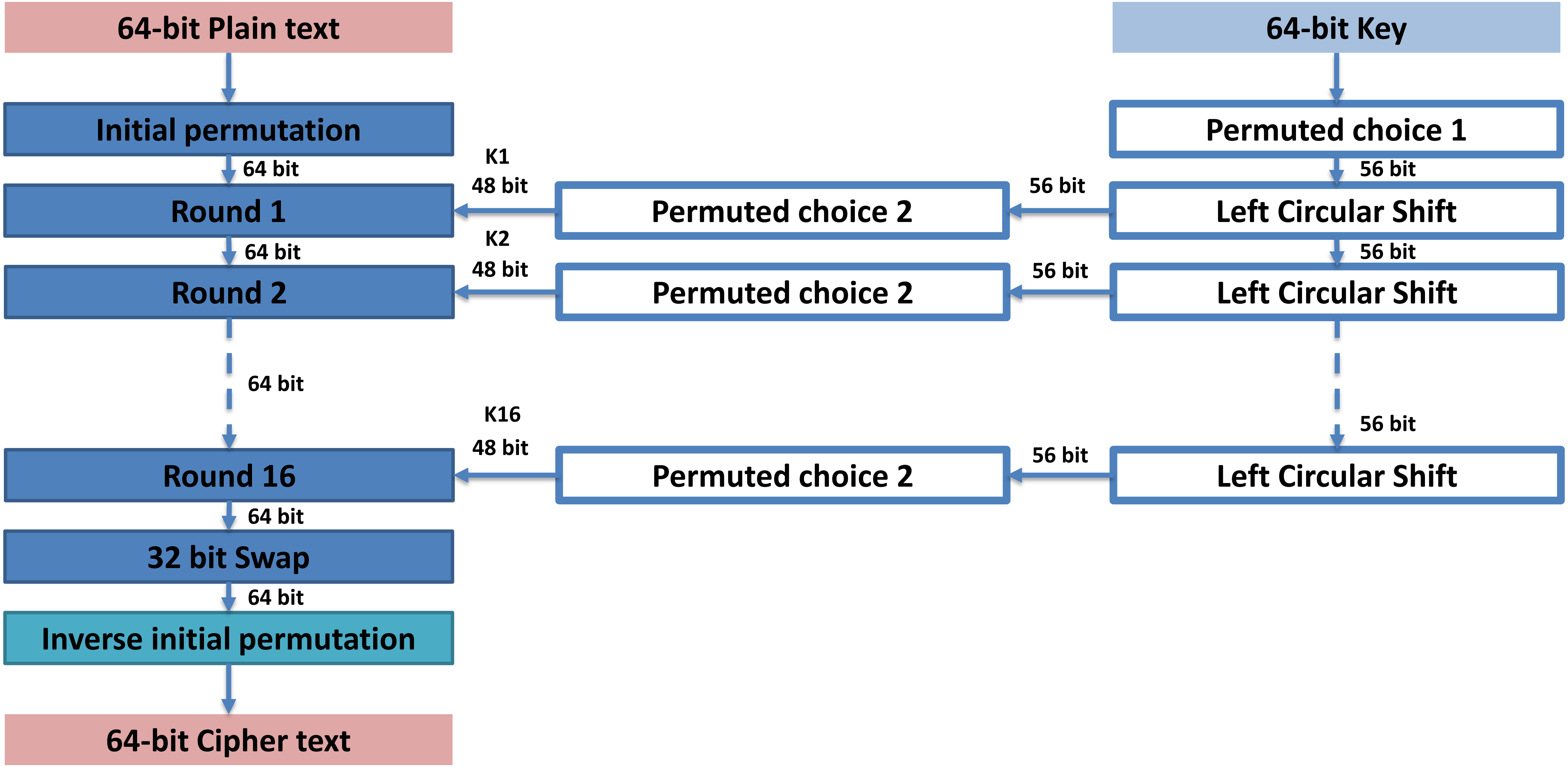
- ❑ In the next round, we will have $L_2 = R_1$, which is the block we just calculated, and then we must calculate $R_2 = L_1 \oplus f(R_1, K_2)$, and so on for 16 rounds.

- ❑ At the end of the sixteenth round we have the blocks L_{16} and R_{16}. We then ***reverse*** the order of the two blocks into the 64-bit block

DES Encryption

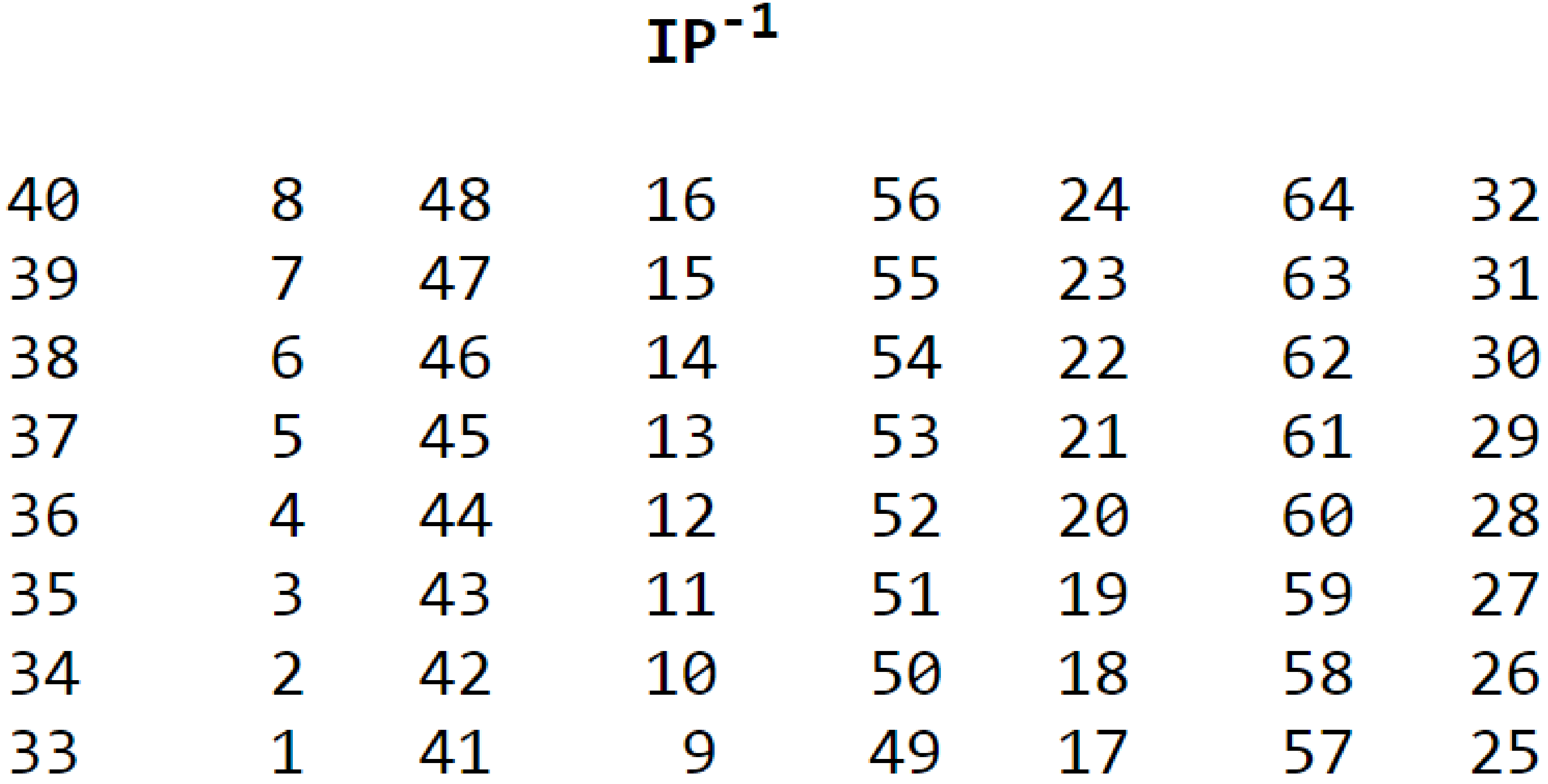
- ❑ If we process all 16 blocks using the method defined previously, we get, on the 16th round
- ❑ L_{16} = 0100 0011 0100 0010 0011 0010 0011 0100
- ❑ R_{16} = 0000 1010 0100 1100 1101 1001 1001 0101
- ❑ We reverse the order of these two blocks and apply the final permutation to
- ❑ $R_{16}L_{16}$ = 00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100

DES Algorithm



DES Encryption

- Then apply a final permutation IP^{-1} as defined by the 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

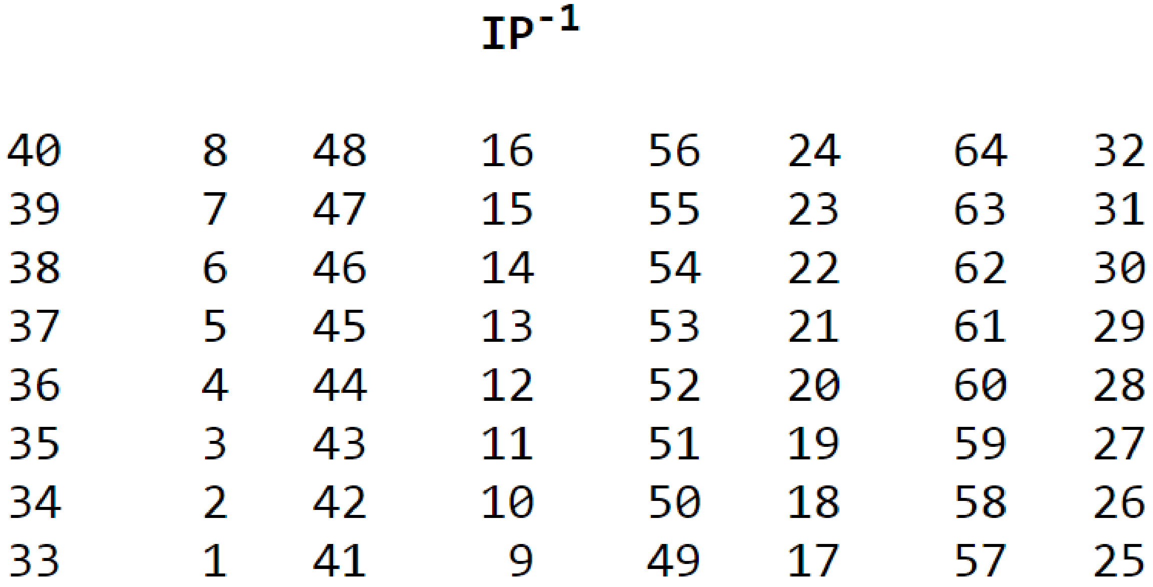
DES Encryption

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

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

- ❑ which in hexadecimal format is

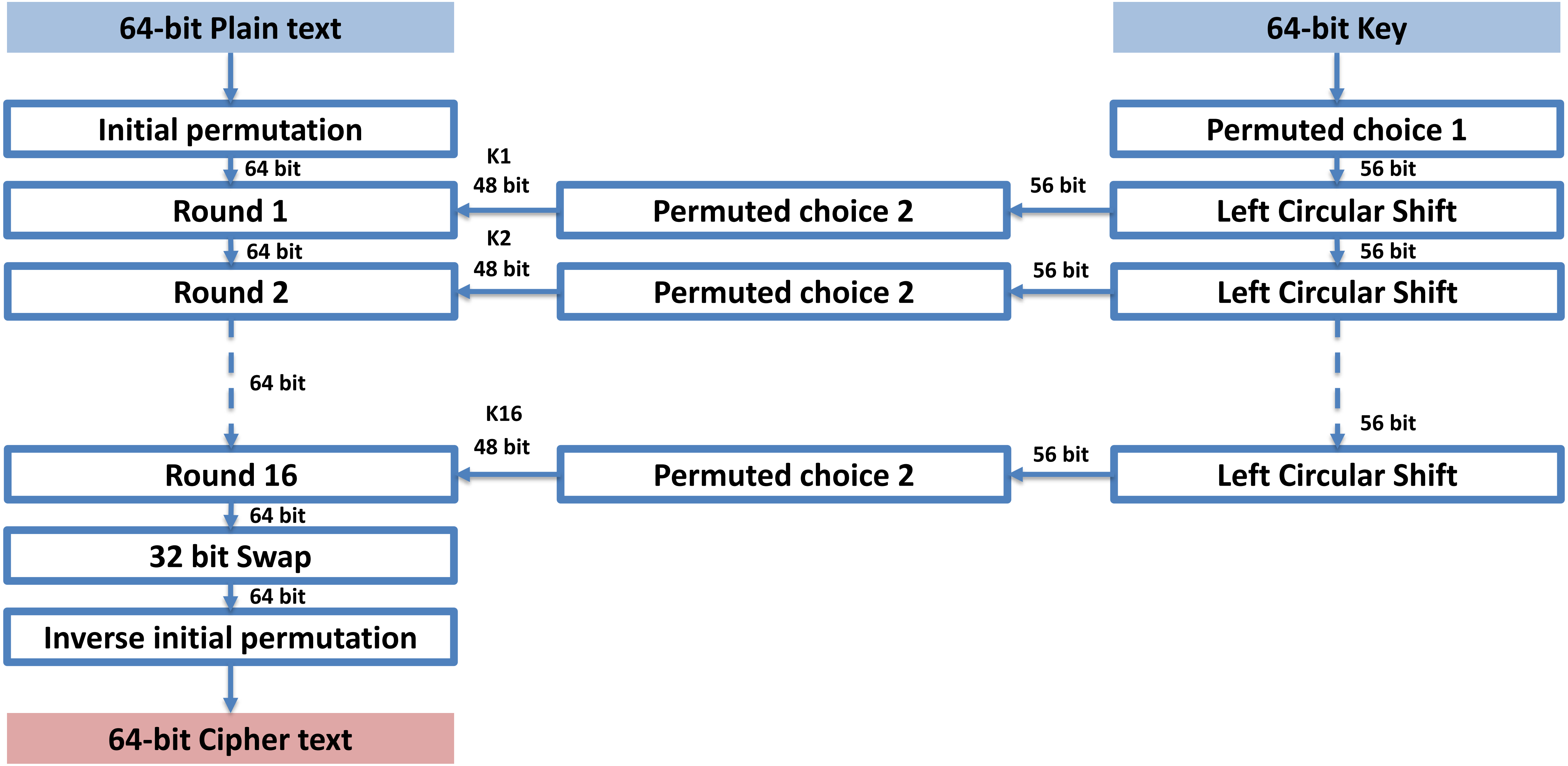
85E813540F0AB405



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

DES Algorithm



64-bit Plain text

Initial permutation

64 bit

Round 1

64 bit

Round 2

64 bit

Round 16

64 bit

32 bit Swap

64 bit

Inverse initial permutation

64-bit Cipher text

K1

48 bit

K2

48 bit

K16

48 bit

Permuted choice 2

Permuted choice 2

Permuted choice 2

56 bit

56 bit

56 bit

64-bit Key

Permuted choice 1

56 bit

Left Circular Shift

56 bit

Left Circular Shift

56 bit

Left Circular Shift

DES Encryption

- ❑ This is the encrypted form of M = 0123456789ABCDEF with K = 133457799BBCDFF1

- ❑ C = 85E813540F0AB405

Table of Contents

- Stream Ciphers and Block Ciphers
- Data Encryption Standard
- DES Algorithm
- DES Key Creation
- DES Encryption
- The Strength Of DES

The Strength Of DES

- With a key length of 56 bits, there are 2^{56} possible keys
- Brute force search looks hard
- Fast forward to 1998. Under the direction of John Gilmore of the EFF, a team spent $220,000 and built a machine that can go through the entire 56-bit DES key space in an average of 4.5 days.
- On July 17, 1998, they announced they had cracked a 56-bit key in 56 hours. The computer, called Deep Crack, uses 27 boards each containing 64 chips, and is capable of testing 90 billion keys a second.

Contact Me

facebook.com/mloey

mohamedloey@gmail.com

twitter.com/mloey

linkedin.com/in/mloey

mloey@fci.bu.edu.eg

mloey.github.io

THANKS FOR YOUR TIME