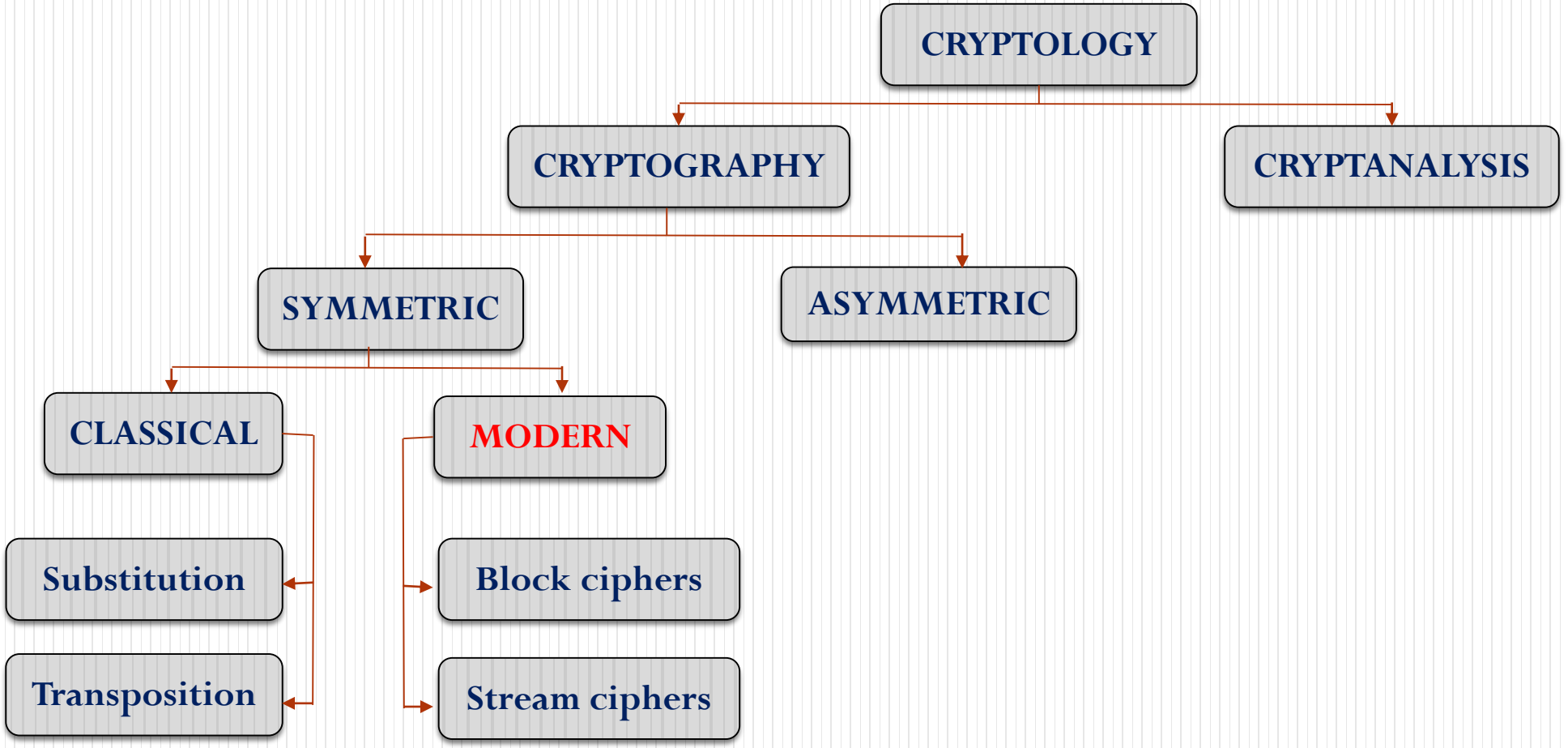
SYMMETRIC CIPHERS

Contents

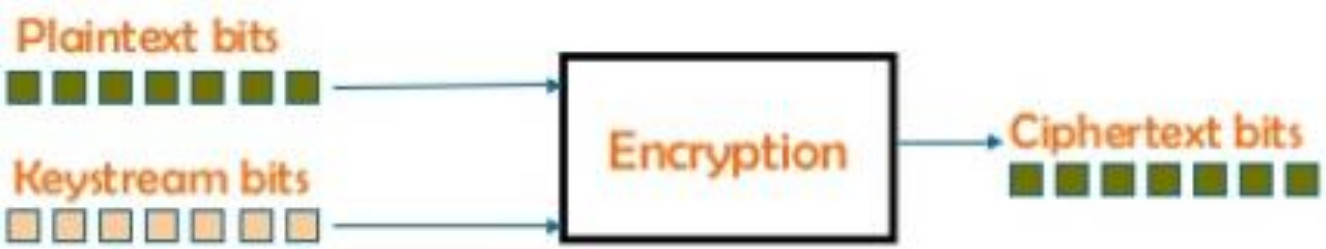
1) Block Cipher Principles
2) Feistel Cipher
3) Data Encryption Standard (DEC)


CRYPTOLOGY
CRYPTOGRAPHY
CRYPTANALYSIS
SYMMETRIC
ASYMMETRIC
CLASSICAL
MODERN
Substitution
Transposition
Block ciphers
Stream ciphers

1. Block Cipher Principles

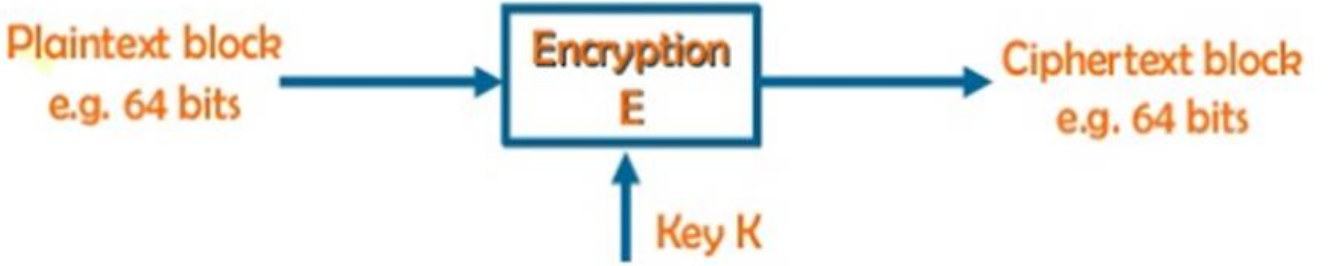
Stream Ciphers and Block Ciphers

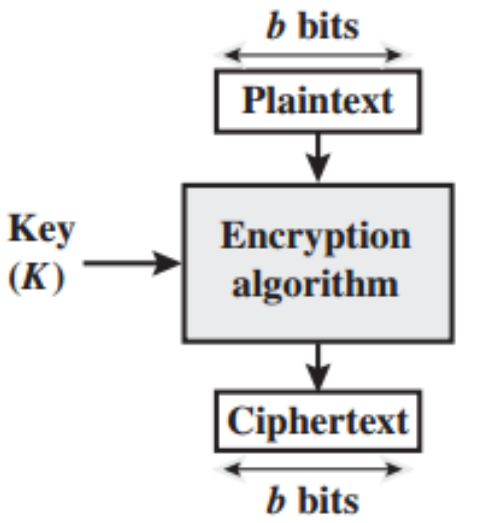
- A stream cipher is a type pf symmetric encryption in which input data is encrypted one bit (byte) at a time.



Stream Ciphers and Block Ciphers

- **Block Ciphers** is one in which the plaintext is divided in blocks and one block is encrypted at a time producing a ciphertext of equal length





- Block size?

2. Feistel Cipher

The Feistel Cipher

- Horst Feistel devised the feistel cipher
- Most symmetric block ciphers are based on a feistel cipher structure.
- Feistel proposed the use of a cipher that alternates **substitutions** and **permutations**, where these terms are defined as follows:
- **Substitution**: Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements
- **Permutation**: A sequence of plaintext elements is replaced by a permutation of that sequence

Feistel Cipher Structure

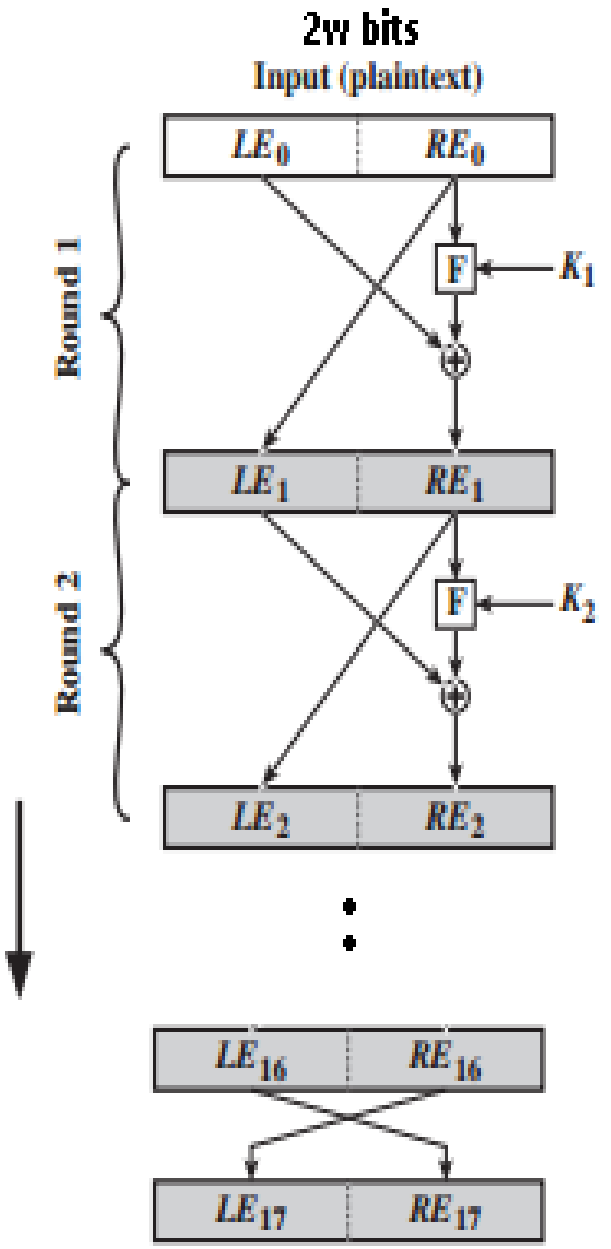
- The Plaintext block: 2w bits
- The plaintext block is divided into two halves,

L_0 and R_0.

- L, R pass through n rounds of processing and then combine to produce the ciphertext block.
- Each round *i (1->16)*:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$



Feistel Cipher Principles

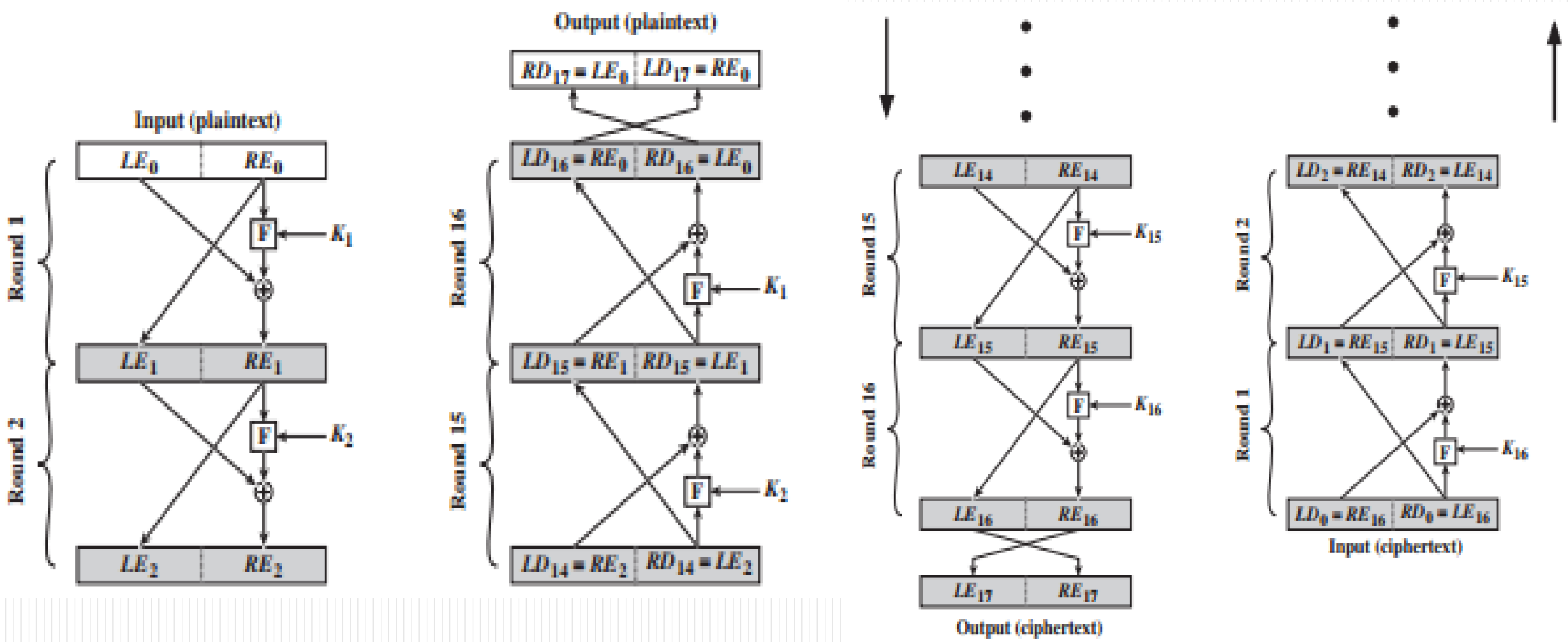
The exact realization of a Feistel network depends on the choice of the following parameters and design features:

- **Block size**: Larger block sizes mean greater security but reduced encryption/decryption speed for a given algorithm.
- **Key size**: Larger key size means greater security but may decrease encryption/decryption speed.
- **Number of rounds**: increase number improves security, but slows cipher

Feistel Cipher Principles (cont.)

- **Subkey generation algorithm:** Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.
- **Round function F:** greater complexity generally means greater resistance to cryptanalysis.

Feistel decryption algorithm



Feistel decryption algorithm

- Consider the encryption process

$$LE_{16} = RE_{15}$$
$$RE_{16} = LE_{15} \oplus \mathrm{F}(RE_{15}, K_{16})$$

- On the decryption side

$$\begin{aligned} LD_1 &= RD_0 = LE_{16} = RE_{15} \\ RD_1 &= LD_0 \oplus \mathrm{F}(RD_0, K_{16}) \\ &= RE_{16} \oplus \mathrm{F}(RE_{15}, K_{16}) \\ &= [LE_{15} \oplus \mathrm{F}(RE_{15}, K_{16})] \oplus \mathrm{F}(RE_{15}, K_{16}) \end{aligned}$$

Feistel decryption algorithm

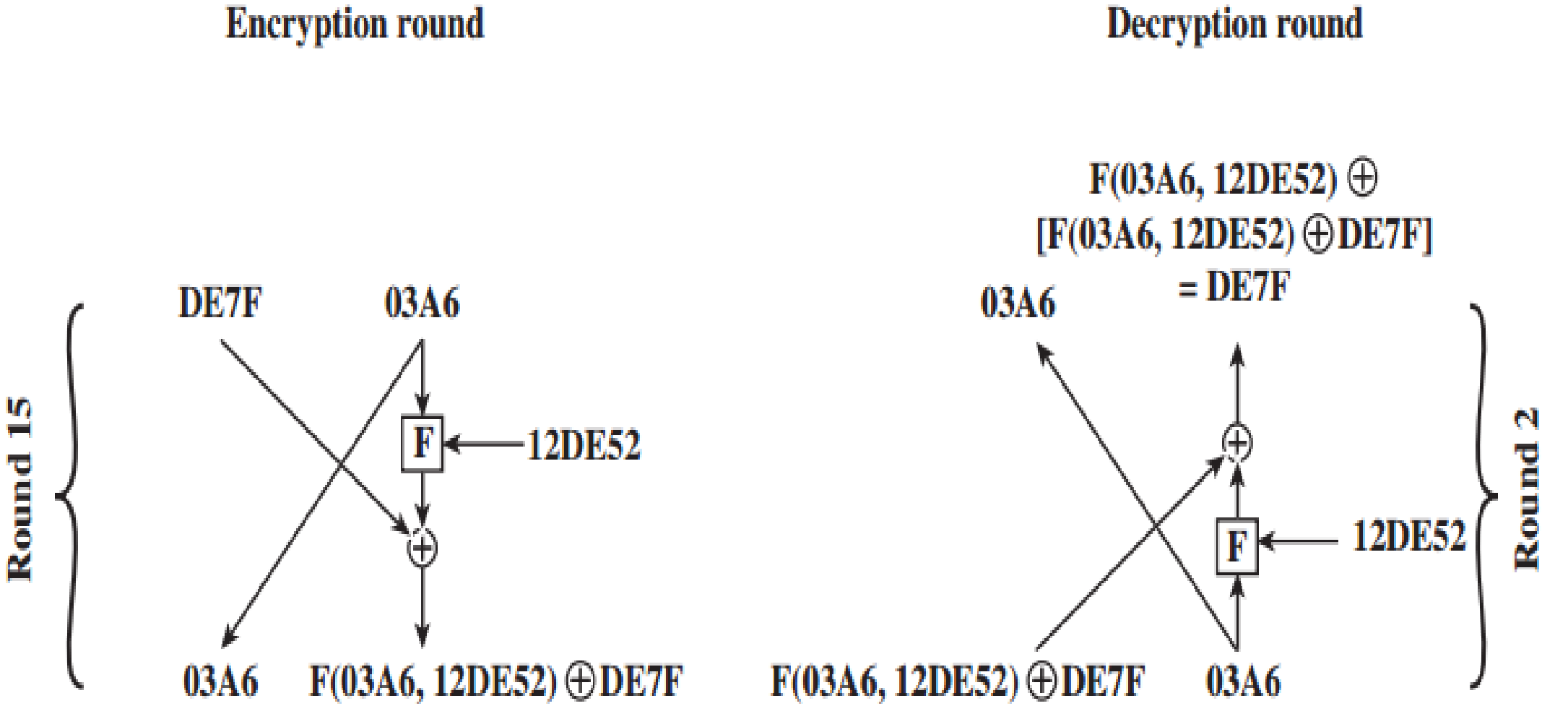
- The XOR has the following properties:

$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$
$$D \oplus D = 0$$
$$E \oplus 0 = E$$

- Thus, we have:

$$LD_1 = RE_{15} \text{ and } RD_1 = LE_{15}$$

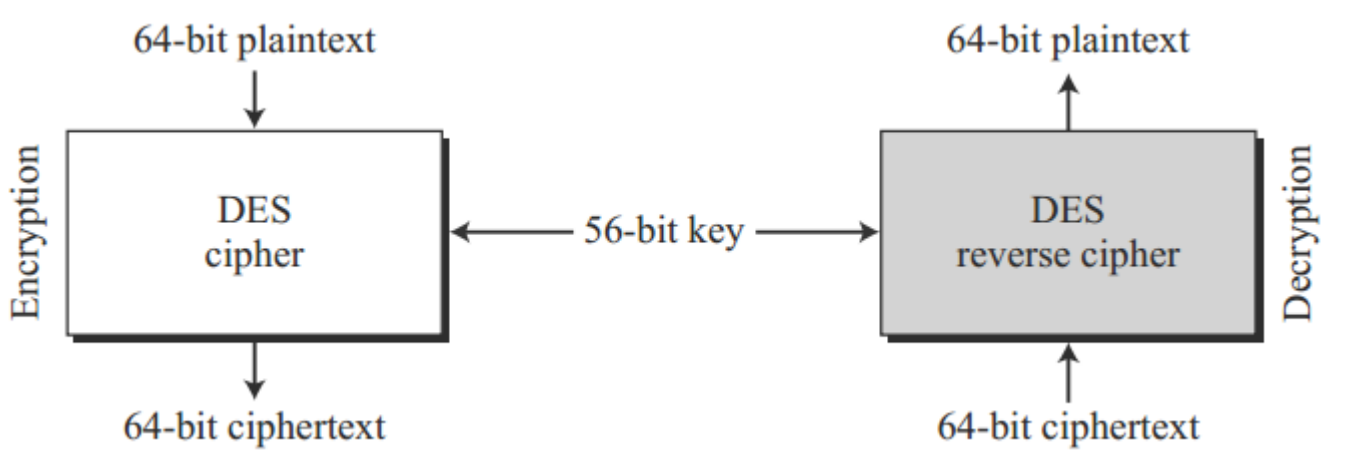
Feistel Example

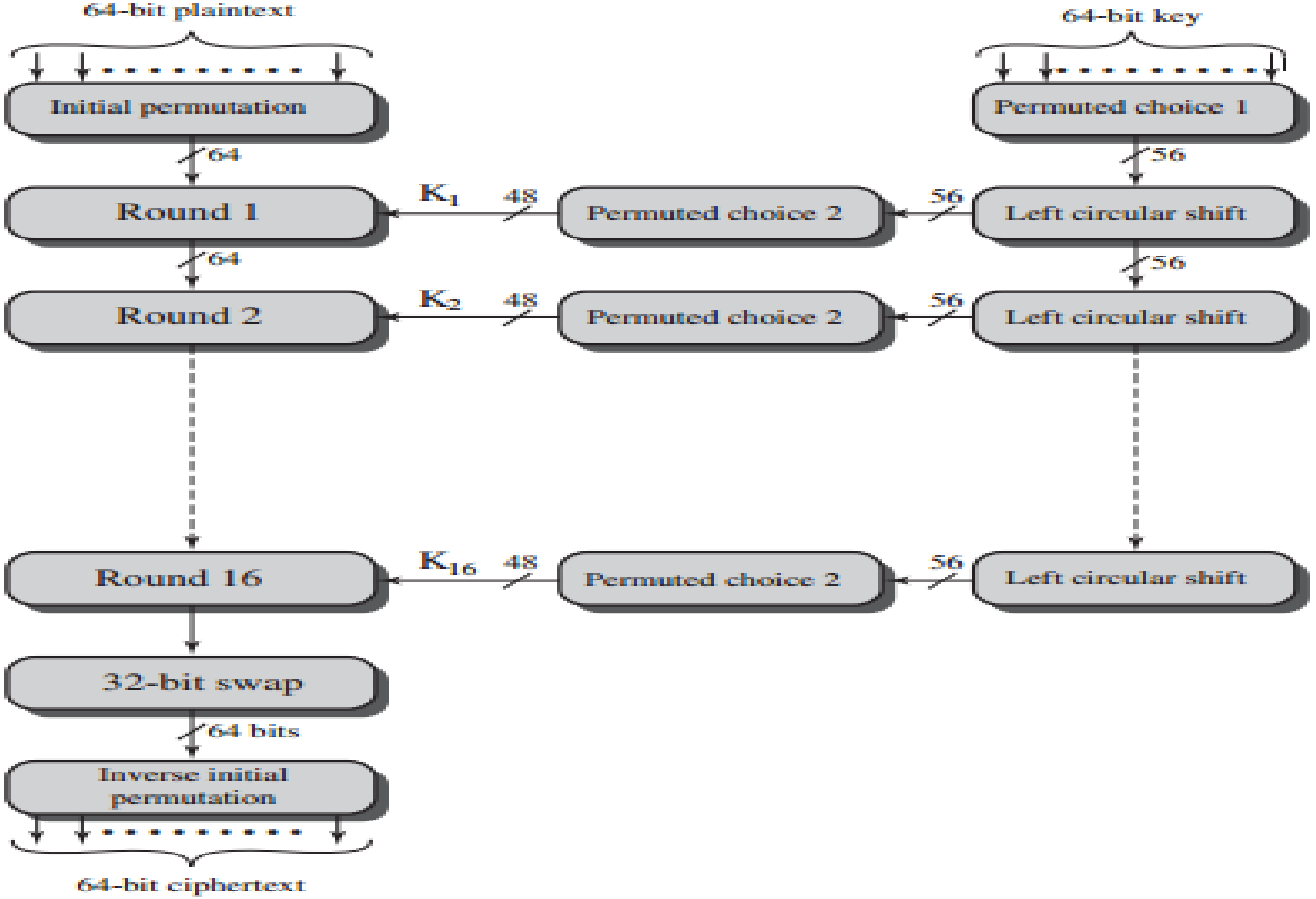


3. Data Encryption Standard - DEC

Data Encryption Standard - DEC

- The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).
- A block cipher
- Data Encrypted in 64-bits blocks using a 56-bit key; ciphertext is of 64-bit long.
- Number of rounds: 16




64-bit plaintext
64-bit key
Initial permutation
Permuted choice 1
64
56
Round 1
K_1
48
Permuted choice 2
56
Left circular shift
64
56
Round 2
K_2
48
Permuted choice 2
56
Left circular shift
Round 16
K_{16}
48
Permuted choice 2
56
Left circular shift
32-bit swap
64 bits
Inverse initial permutation
64-bit ciphertext

Description

- The 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input.
- Sixteen rounds of the same function, which involves both permutation and substitution functions. The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the preoutput.
- Finally, the preoutput is passed through the inverse of the initial permutation function, to produce the 64-bit ciphertext

Initial Permutation (IP)

- The ouput bit-1 for example is the input bit 58

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

- Find the output of the initial permutation box when the input is given in hexadecimal as: **0x0002 0000 0000 0001**

Inverse Initial Permutation (IP^{-1})

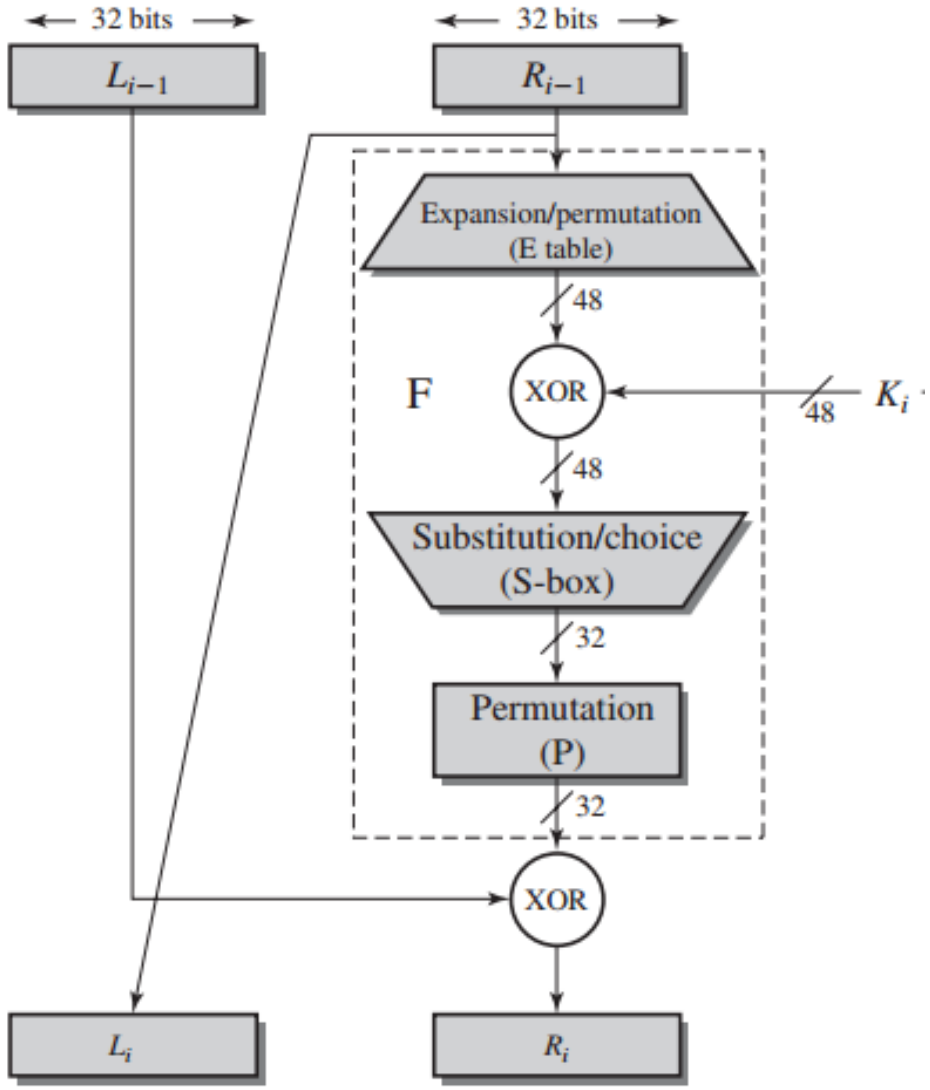
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Find the output of the inverse initial permutation box when the input is given in hexadecimal as: **0x0000 0080 0000 0002**

Details Of single round

- The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L (left) and R (right)

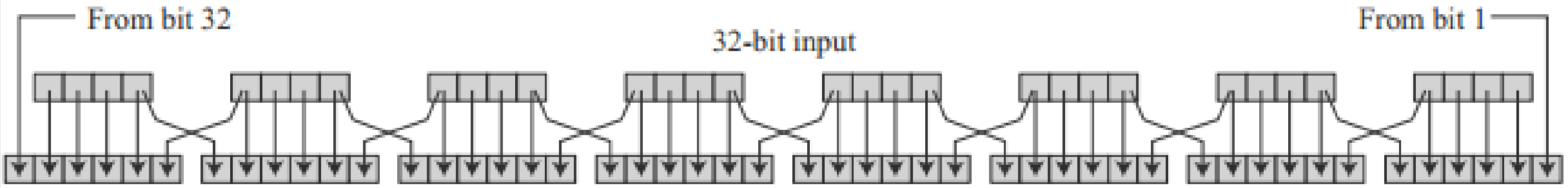
$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$



E table

- Expansion Permutation: 32bit → 48 bit

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



Details Of single round (cont.)

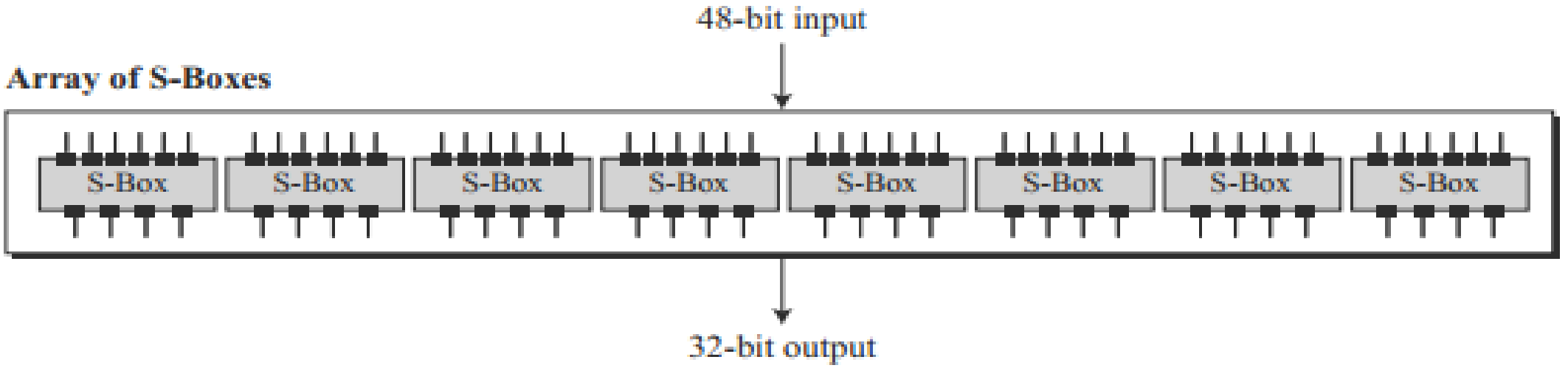
- After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key

XOR	b	
a	0	1
0	0	1
1	1	0

S-Boxes

- Input: 6-bit
- Output: 4-bit

The 48-bit data from the second operation is divided into eight 6-bit chunks, and each chunk is fed into a box. The result of each box is a 4-bit chunk



S-Boxes (cont.)

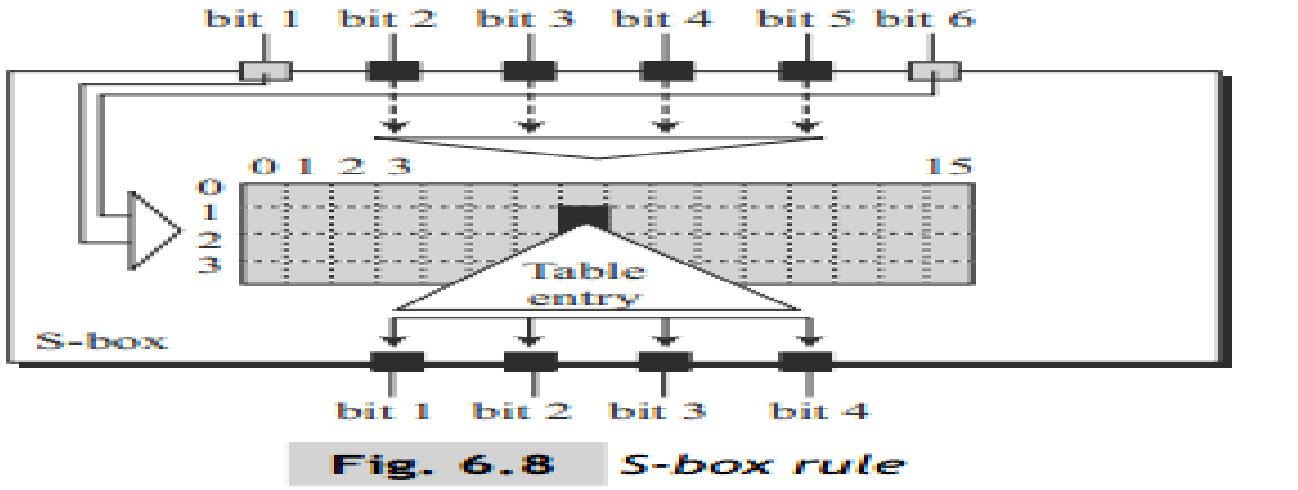


Fig. 6.8 S-box rule

- The combination of bits 1 and 6 of the input defines one of four rows
- The combination of bits 2 through 5 defines one of the sixteen columns
- The values of the outputs are given as decimal numbers to save space. These need to be changed to binary

S-Boxes (cont.)

S-box 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	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 2

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

S-box 3

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

S-box 4

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

S-box 5

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

S-box 6

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

S-box 7

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

S-box 8

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

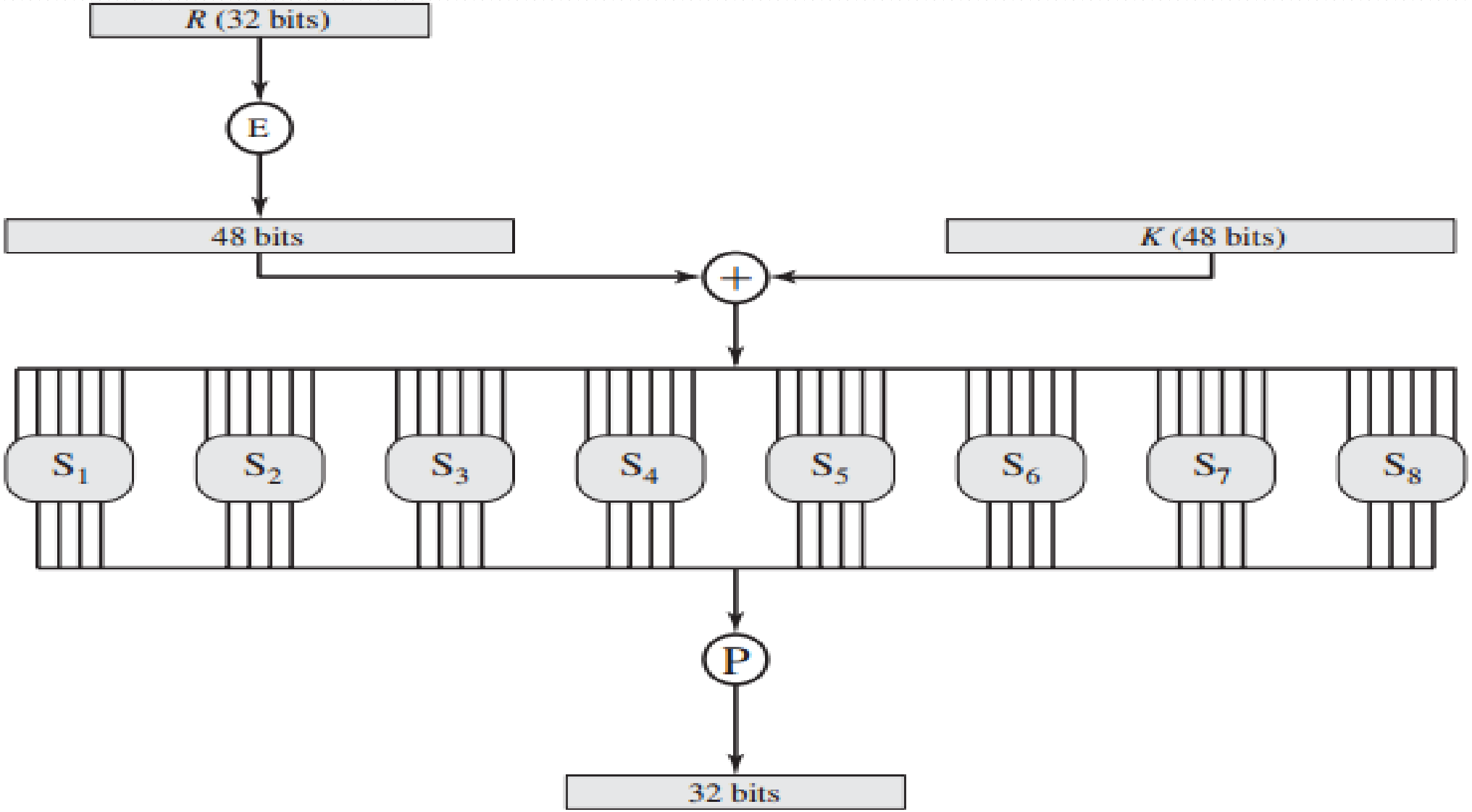
S-Boxes (cont.)

- The input to S-box 1 is 100011. What is the output?
- The input to S-box 8 is 000000. What is the output?

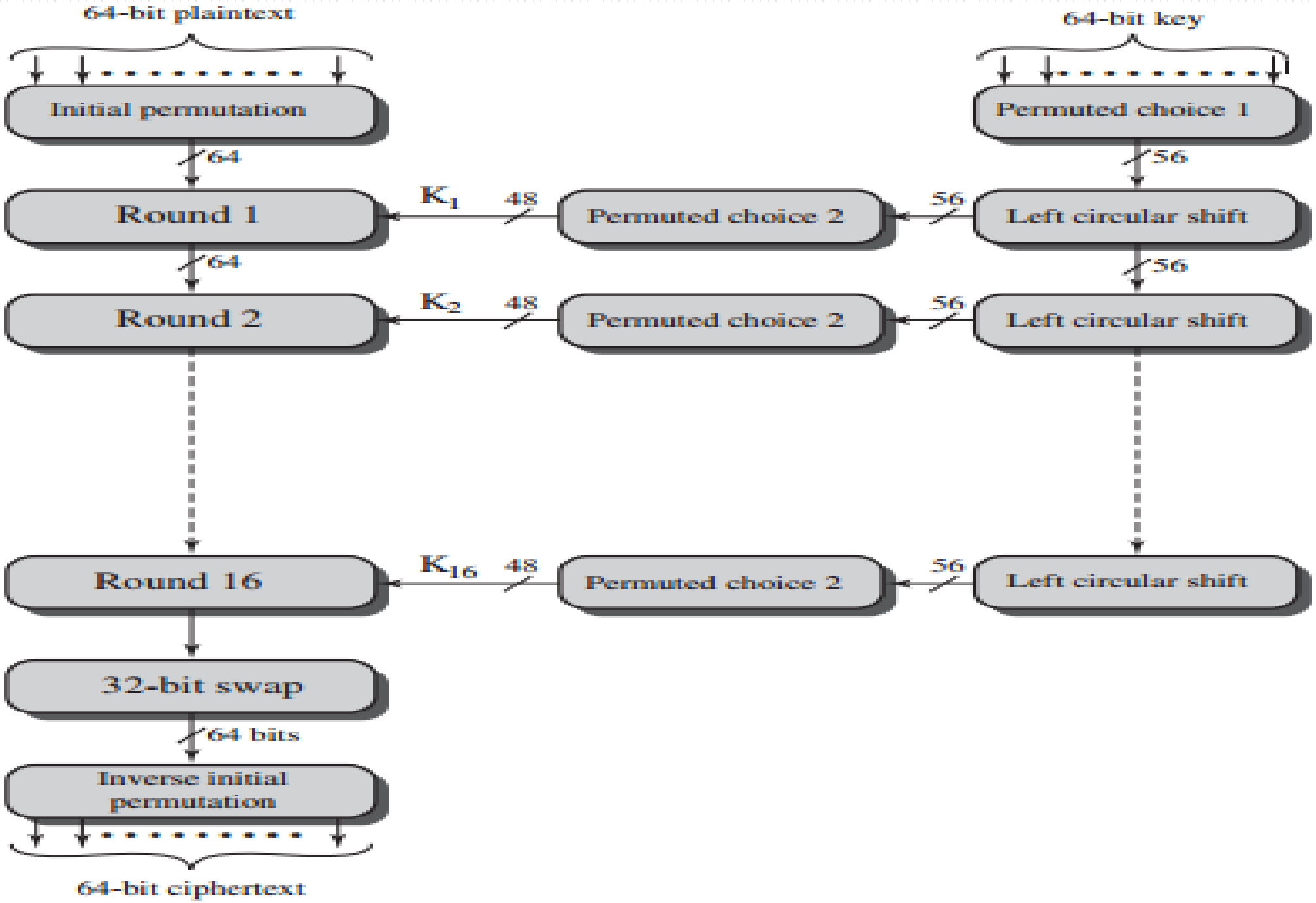
Permutation Function (P)

- The 32-bit output is permutated as defined in the permutation table also to produce the output

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25



Calculation of F(R, K)


64-bit plaintext
64-bit key
Initial permutation
Permuted choice 1
64
56
Round 1
K_1
48
Permuted choice 2
56
Left circular shift
64
56
Round 2
K_2
48
Permuted choice 2
56
Left circular shift
Round 16
K_{16}
48
Permuted choice 2
56
Left circular shift
32-bit swap
64 bits
Inverse initial permutation
64-bit ciphertext

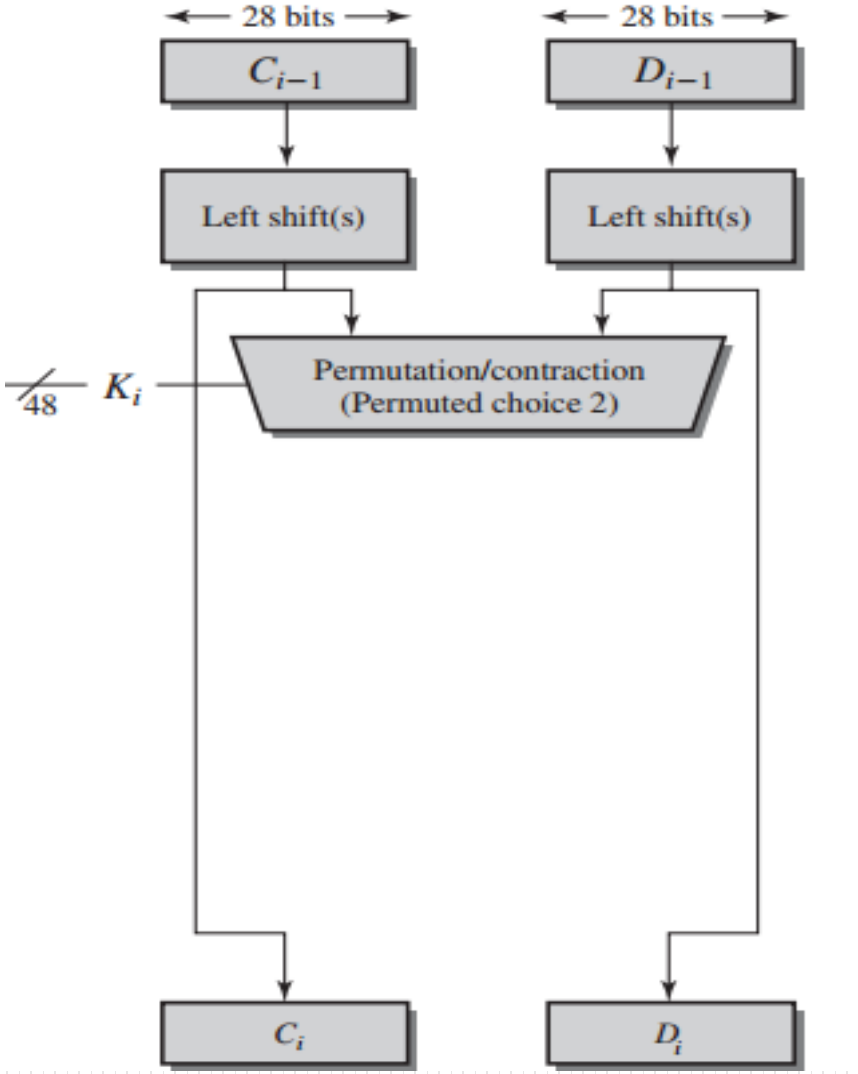
Key Generation (cont.)

- Permuted Choice One (PC-1):
  - Input: 64-bit
  - Ouput: 56-bit (Discard the parity-check bits)

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

Key Generation (cont.)

- The resulting 56-bit key is then treated as two 28-bit quantities, labeled C_0 and D_0
- At each round, C_{i-1} and D_{i-1} are separately subjected to a circular left shift or (rotation) of 1 or 2 bits governed by the DES key calculation table



Schedule of Left Shifts

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

Permuted Choice Two (PC-2)

- PC-2 changes the 58 bits to 48 bits, which are used as a key for a round.

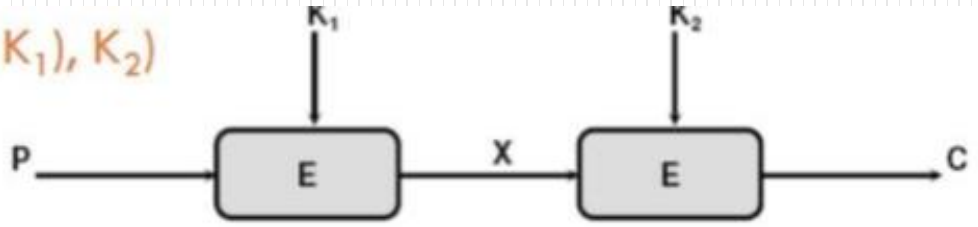
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

The Strength Of Des

- Brute-force attack: With a key length of 56 bits, there are 2^{56} possible keys, which is approximately $7.2 * 10^6$ keys.
  - 1997 on alarge network of computers in afew months
  - 1998 on dedicated H/W in a few days
  - 1999 aboved combined in 22 hours (DES cracker + 100,000 computers)
- => double DES, triple DES, AES

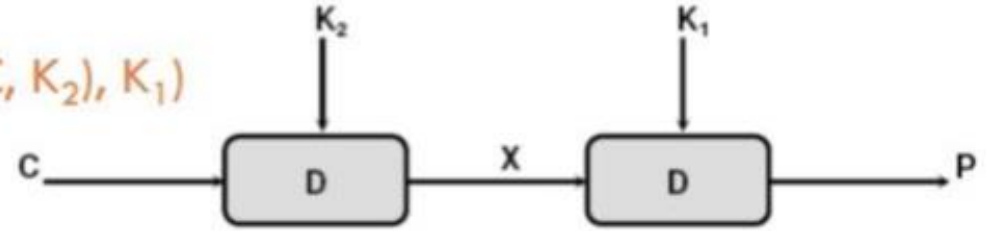
Double DES

$C = E(E(P, K_1), K_2)$



a- Mã hoá

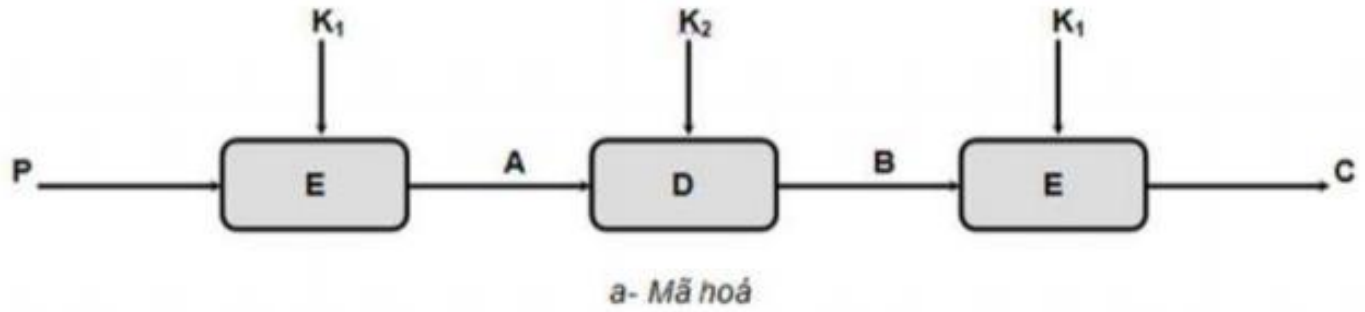
$P = D(D(C, K_2), K_1)$



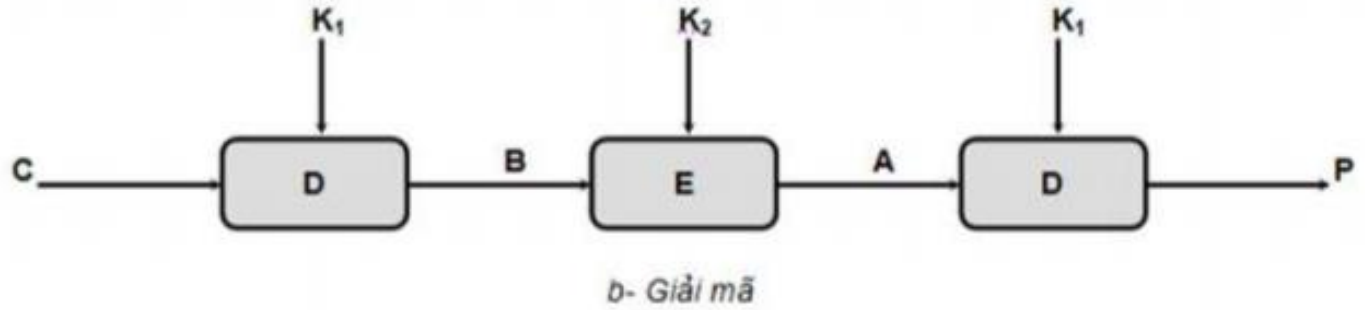
b- Giải mã

Triple DES

$C = E(D(E(P, K_1), K_2), K_1)$



a- Mã hoá



b- Giải mã

$P = D(E(D(C, K_1), K_2, K_1))$

Thanks