



The Islamic University
College of Technical Engineering
Department of Computer Technical Engineering



Fourth Stage

Security

Lecture 13

Asst. Lec. Yousif Samer Mudhafar

Email: yousif.samir19@gmail.com

Lecture Objectives

The student will recognize the following objectives:

➤ **Data Encryption Standard (DES).**

- **Introduction.**
- **History.**
- **Overview.**
- **DES Encryption.**
- **DES Structure.**
- **DES Function.**



Introduction

The **Data Encryption Standard (DES)** is a symmetric-key block cipher published by the **National Institute of Standards and Technology (NIST)**.

History

In 1973, NIST published a request for proposals for a national symmetric-key cryptosystem. A proposal from IBM, a modification of a project called Lucifer, was accepted as DES. DES was published in the Federal Register in March 1975 as a draft of the Federal Information Processing Standard (FIPS). After the publication, the draft was criticized severely for two reasons. First, critics questioned the small key length (only 56 bits), which could make the cipher vulnerable to a brute-force attack. Second, critics were concerned about some hidden design behind the internal structure of DES. They were suspicious that some part of the structure (the S-boxes) may have some hidden trapdoor that would allow the National Security Agency (NSA) to decrypt the messages without the need for the key. Later IBM designers mentioned that the internal structure was designed to prevent differential cryptanalysis. DES was finally published as FIPS 46 in the Federal Register in January 1977. NIST, however, defines DES as the standard for use in unclassified applications. DES has been the most widely used symmetric-key block cipher since its publication. NIST later issued a new standard (FIPS 46-3) that recommends the use of triple-DES (repeated DES cipher three times) for future applications.

Overview

DES is a block cipher, as shown in **Figure 1**.

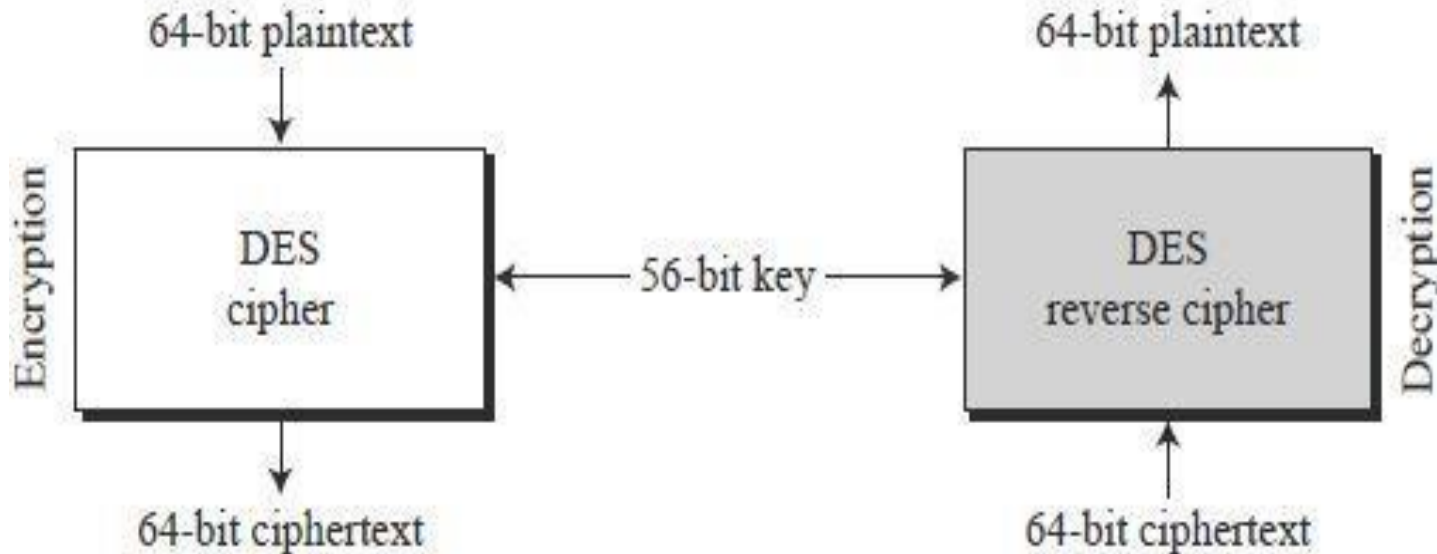


Figure 1: Encryption and decryption with DES.

At the encryption site, DES takes a 64-bit plaintext and creates a 64-bit Ciphertext; at the decryption site, DES takes a 64-bit Ciphertext and creates a 64-bit block of plaintext. The same 56-bit cipher key is used for both encryption and decryption.

DES Encryption

The overall scheme for DES encryption is illustrated in **Figure 2**. As with any encryption scheme, there are two inputs to the encryption function: the plaintext to be encrypted and the key. In this case, the plaintext must be 64 bits in length and the key is 56 bits in length.

Looking at the left-hand side of the figure, we can see that the processing of the plaintext proceeds in three phases. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input. This is followed by a phase consisting of 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 a permutation $[IP^{-1}]$ that is the inverse of the initial permutation function, to produce the 64-bit Ciphertext. With the exception of the initial and final permutations, DES has the exact structure of a Feistel cipher, as shown in **Figure 2**.

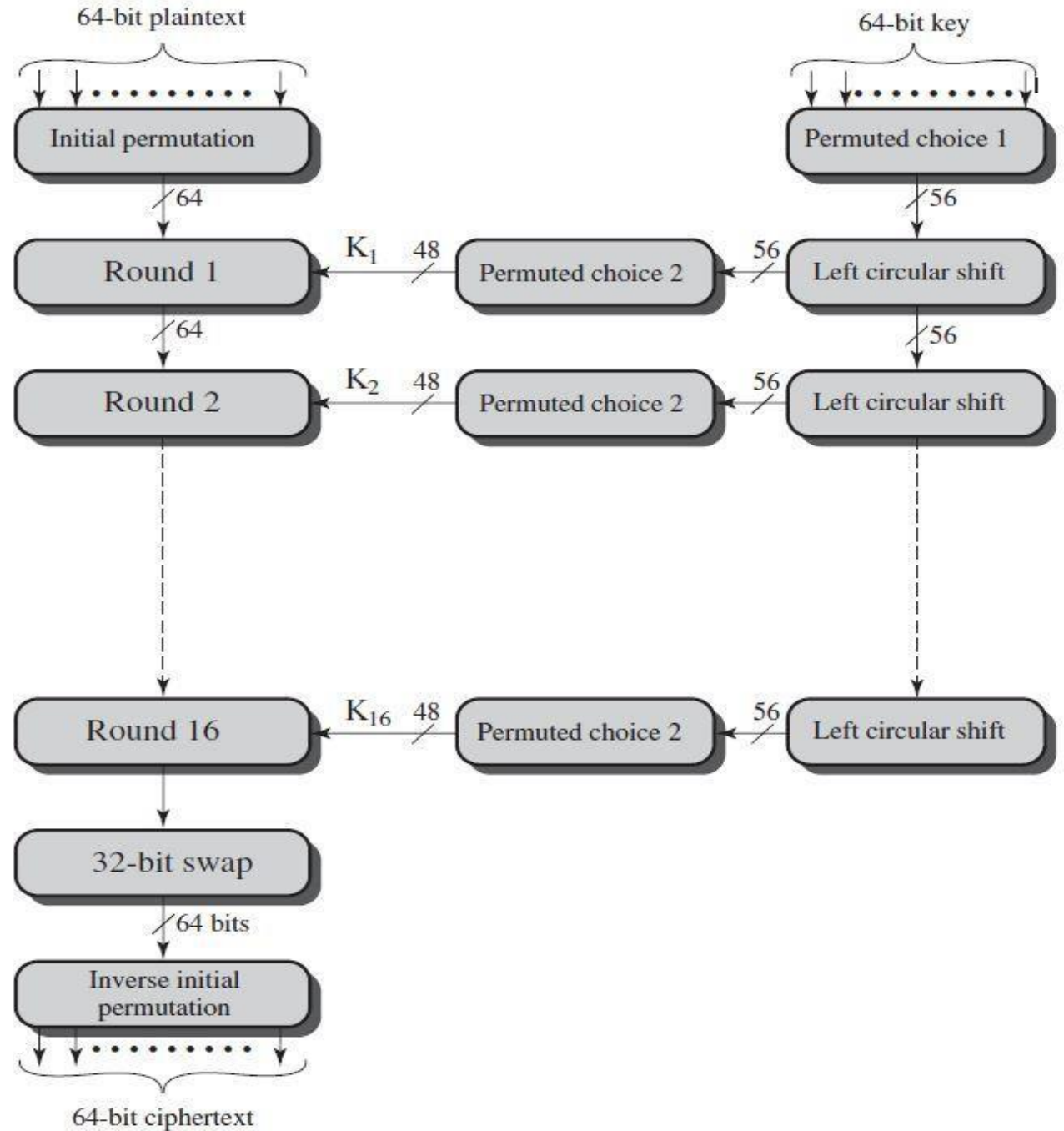


Figure 2: General Depiction of DES Encryption Algorithm.

DES STRUCTURE

Let us concentrate on encryption; later we will discuss decryption. The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Feistel rounds. Each round uses a different 48-bit round key generated from the cipher key according to a predefined algorithm described later in the chapter. **Figure 3** shows the elements of the DES cipher at the encryption site.

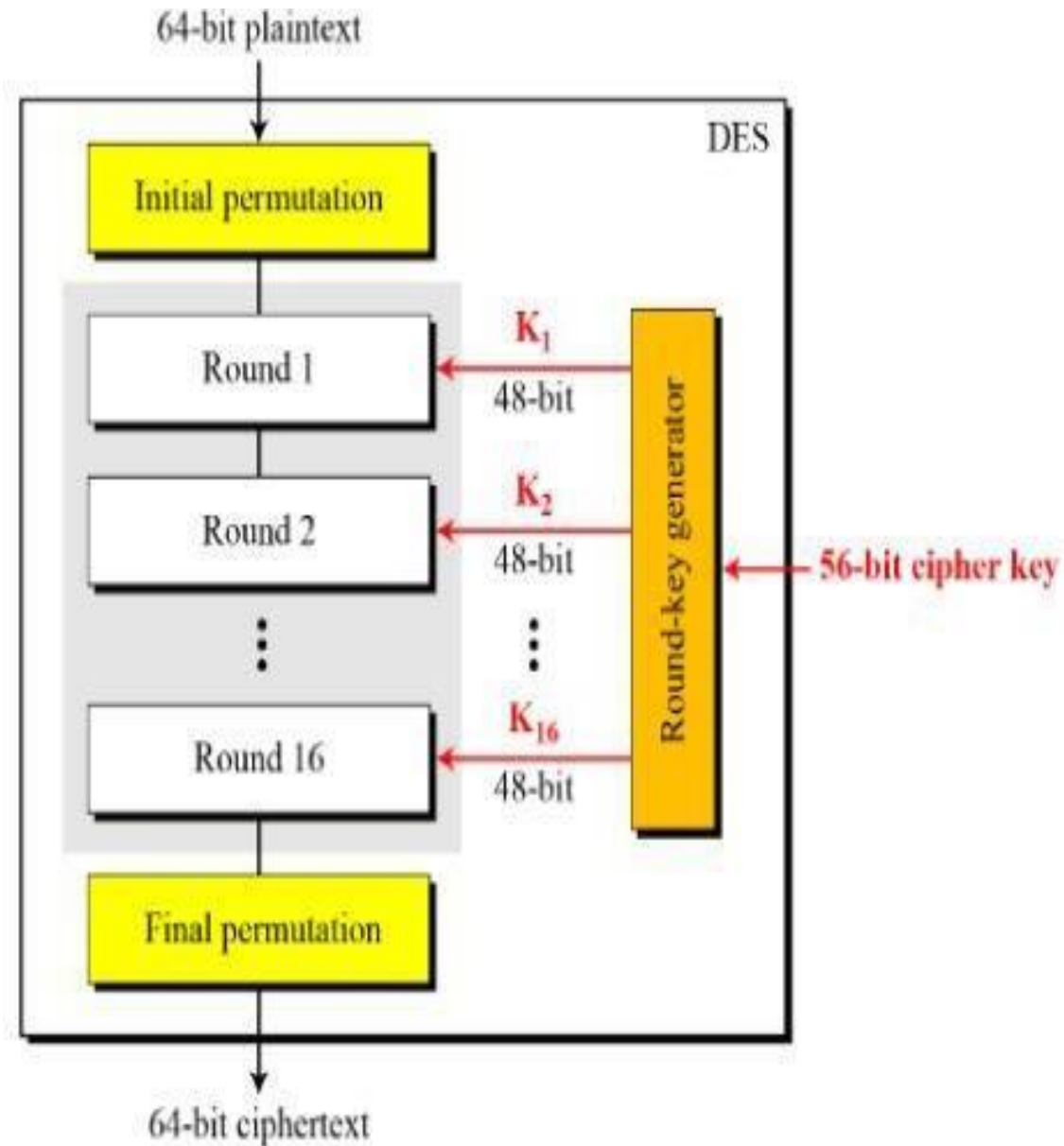


Figure 3: General structure of DES.

Initial Permutation (IP) and Final Permutation or Inverse Initial Permutation (IP^{-1})

Figure 4 shows the initial and final permutations (P-boxes). Each of these permutations takes a 64-bit input and permutes them according to a predefined rule. We have shown only a few input ports and the corresponding output ports. These permutations are keyless straight permutations that are the inverse of each other. For example, in the initial permutation, the 58th bit in the input becomes the first bit in the output. Similarly, in the final permutation, the first bit in the input becomes the 58th bit in the output. In other words, if the rounds between these two permutations do not exist, the 58th bit entering the initial permutation is the same as the 58th bit leaving the final permutation.

The permutation rules for these P-boxes are shown in **Table 1 and Table 2**. Each side of the table can be thought of as a 64-element array. Note that, as with any permutation table we have discussed so far, the value of each element defines the input port number, and the order (index) of the element defines the output port number.

These two permutations have no cryptography significance in DES. Both permutations are keyless and predetermined. The reason they are included in DES is not clear and has not been revealed by the DES designers. The guess is that DES was designed to be implemented in hardware (on chips) and that these two complex permutations may thwart a software simulation of the mechanism.

Initial Permutation (IP) and Final Permutation or Inverse Initial Permutation (IP^{-1})

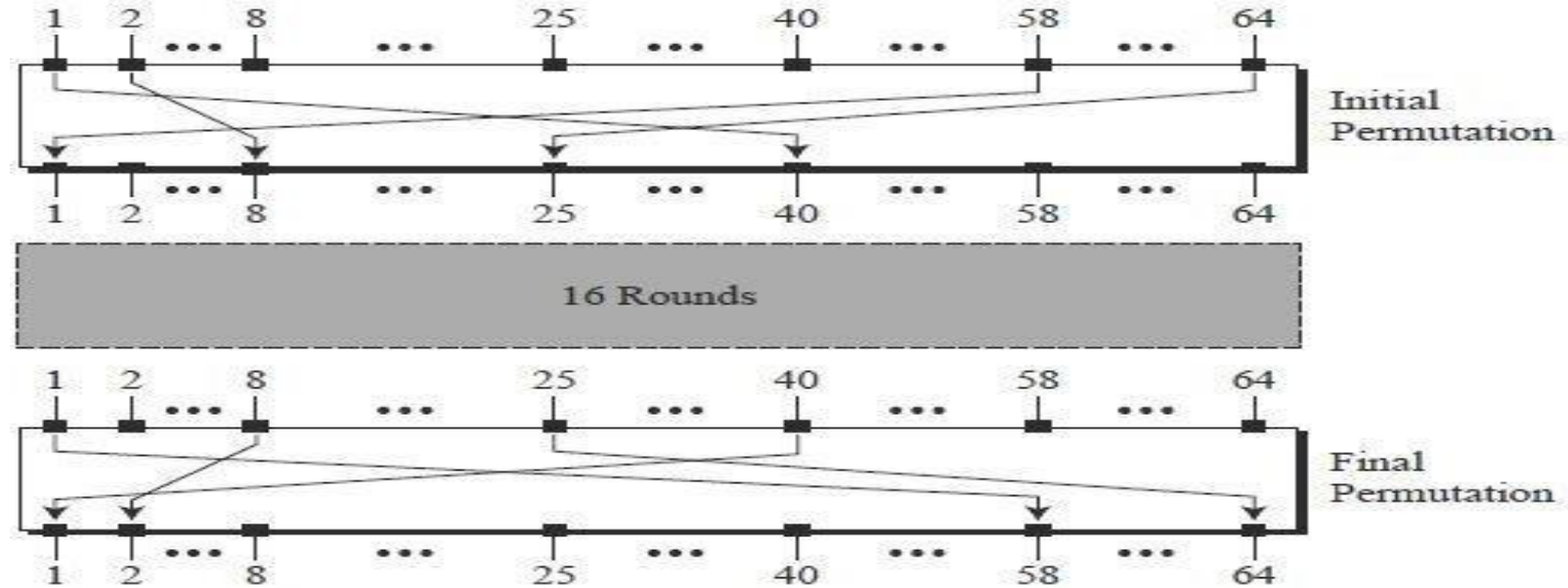


Figure 4: Initial and final permutation steps in DES.

Table 1: 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

Table 2: Final Permutation or 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

Rounds

DES uses 16 rounds. Each round of DES is a Feistel cipher, as shown in **Figure 5**.

The round takes L_{I-1} and R_{I-1} from previous round (or the initial permutation box) and creates L_I and R_I , which go to the next round (or final permutation box). We can assume that each round has two cipher elements (mixer and swapper). Each of these elements is invertible. The swapper is obviously invertible. It swaps the left half of the text with the right half. The mixer is invertible because of the XOR operation. All noninvertible elements are collected inside the function $f(R_{I-1}, K_I)$.

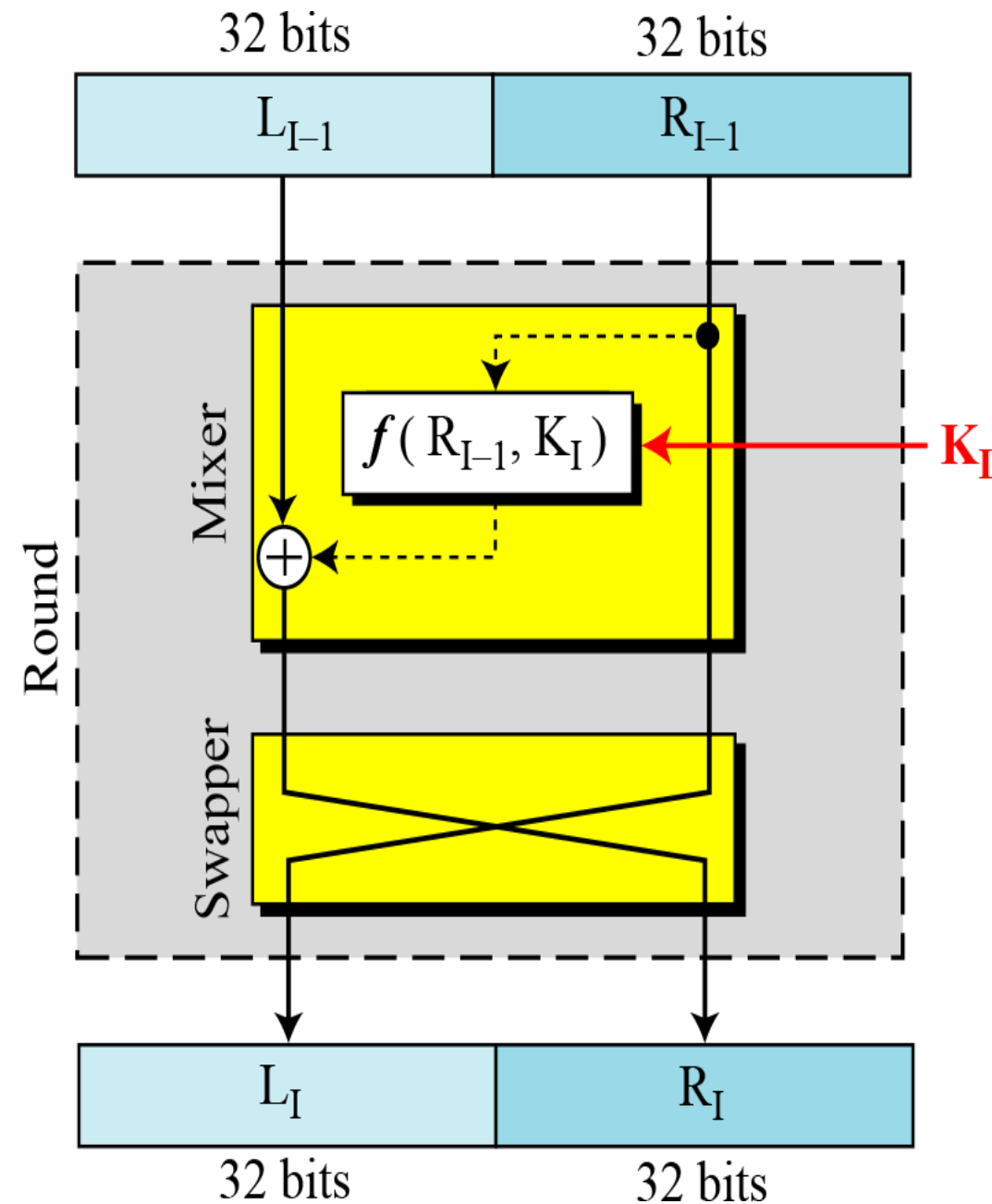


Figure 5: A round in DES (encryption site).

Feistel Encryption and Decryption

FEISTEL CIPHER STRUCTURE The left-hand side of **Figure 6** depicts the structure proposed by Feistel. The inputs to the encryption algorithm are a plaintext block of length bits and a key .The plaintext block is divided into two halves, and . The two halves of the data pass through rounds of processing and then combine to produce the Ciphertext block. Each round has as inputs and derived from the previous round, as well as a subkey derived from the overall . In general, the subkeys are different from and from each other. In **Figure 6**, 16 rounds are used, although any number of rounds could be implemented.

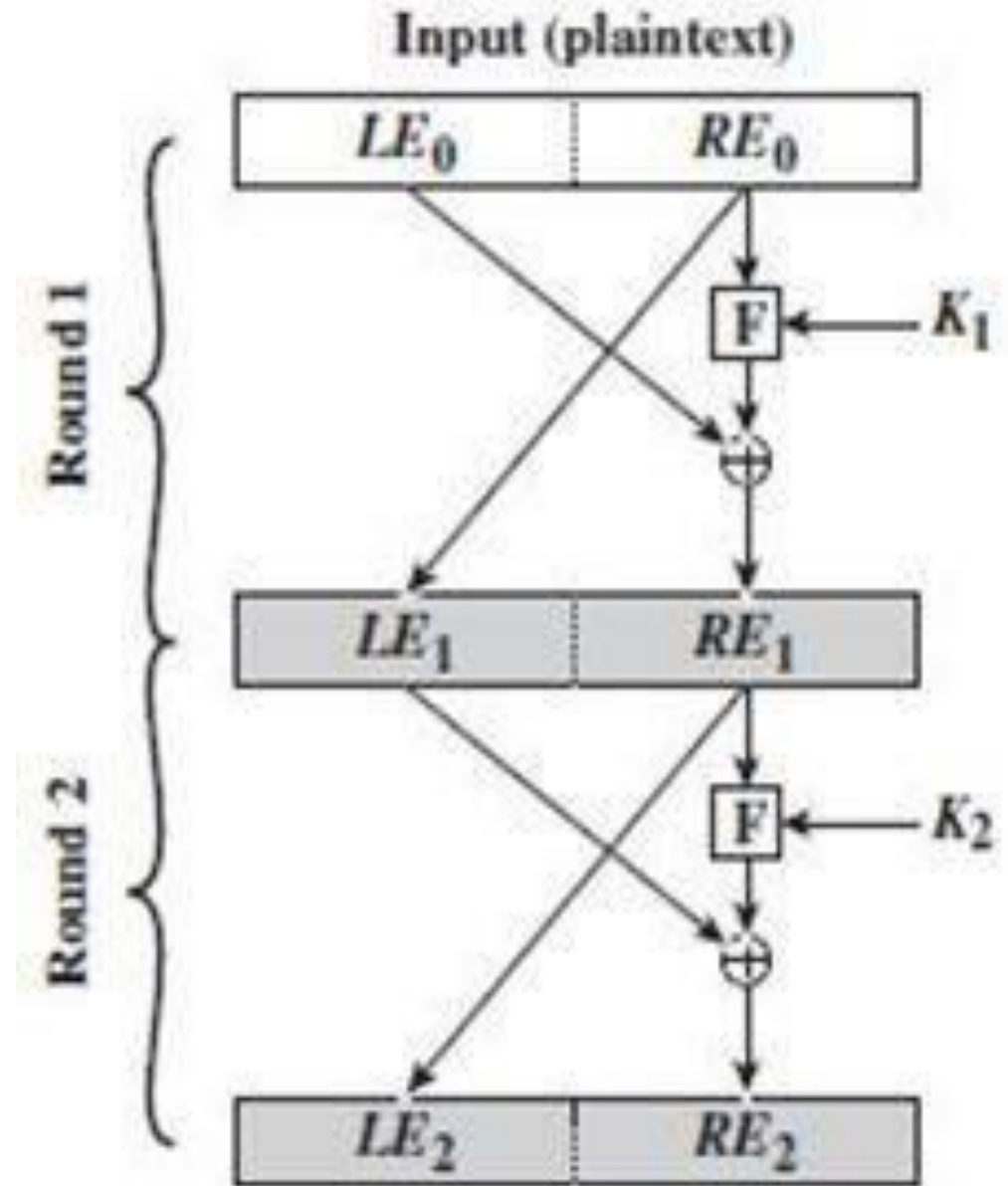


Figure 6: Feistel Encryption (2 rounds).

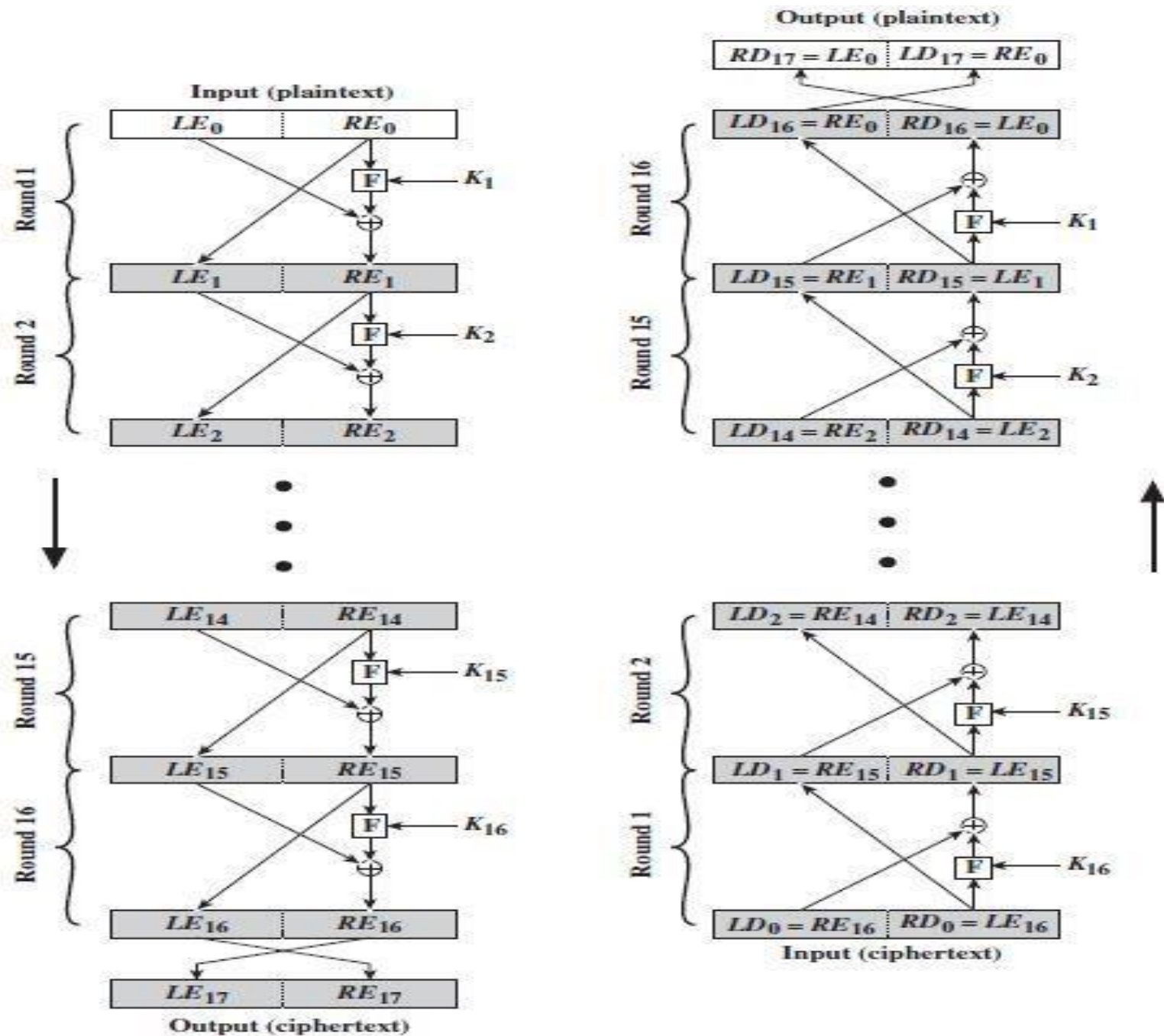


Figure 7: Feistel Encryption and Decryption (16 rounds).

Feistel Encryption and Decryption

All rounds have the same structure. A **substitution** is performed on the left half of the data. This is done by applying a round function F to the right half of the data and then taking the exclusive-OR of the output of that function and the left half of the data. The round function has the same general structure for each round but is parameterized by the round subkey K_I . Another way to express this is to say that F is a function of right-half block of bits and a subkey of bits, which produces an output value of length w bits: $F(RE_i, K_{I+1})$.

Permutation is performed that consists of the interchange of the two halves of the data. This structure is a particular form of the substitution-permutation network (SPN) proposed by Shannon.

DES Function

The heart of DES is the DES function. The DES function applies a 48-bit key to the rightmost 32 bits (R_{I-1}) to produce a 32-bit output. This function is made up of four sections: an expansion D-box, a whitener (that adds key), a group of S-boxes, and a straight D-box as shown in **Figure 8**.

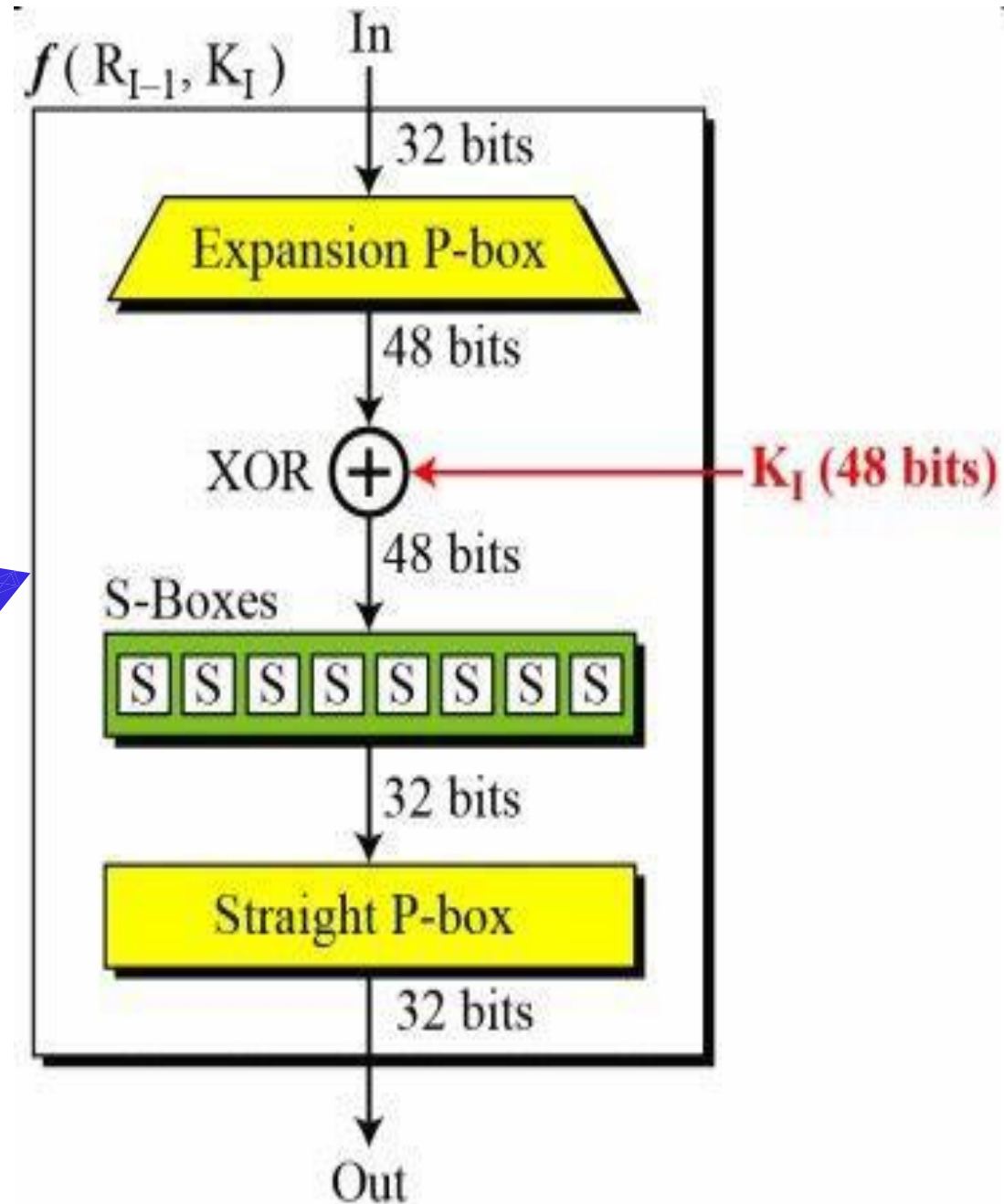
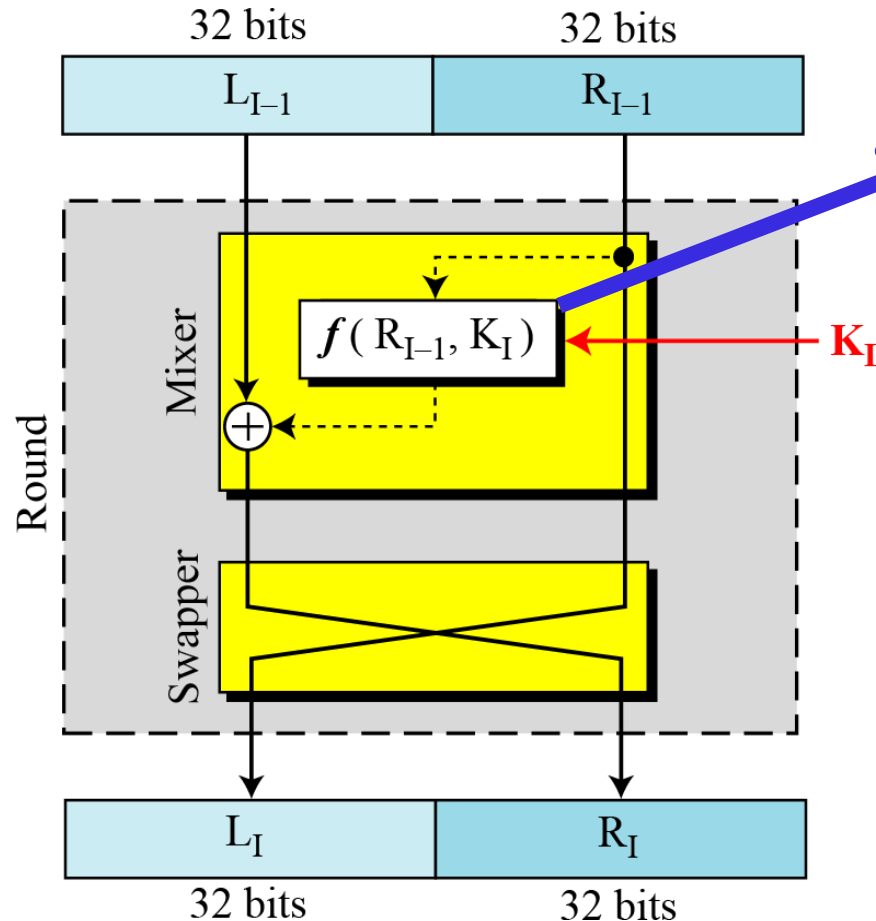


Figure 8: DES Function.

Expansion P-box OR Expansion Permutation

Since R_{I-1} is a 32-bit input and K_I is a 48-bit key, we first need to expand R_{I-1} to 48 bits. R_{I-1} is divided into 8 4-bit sections. Each 4-bit section is then expanded to 6 bits. This expansion permutation follows a predetermined rule. For each section, input bits 1, 2, 3, and 4 are copied to output bits 2, 3, 4, and 5, respectively. Output bit 1 comes from bit 4 of the previous section; output bit 6 comes from bit 1 of the next section. If sections 1 and 8 can be considered adjacent sections, the same rule applies to bits 1 and 32. **Figure 9** shows the input and output in the expansion permutation.

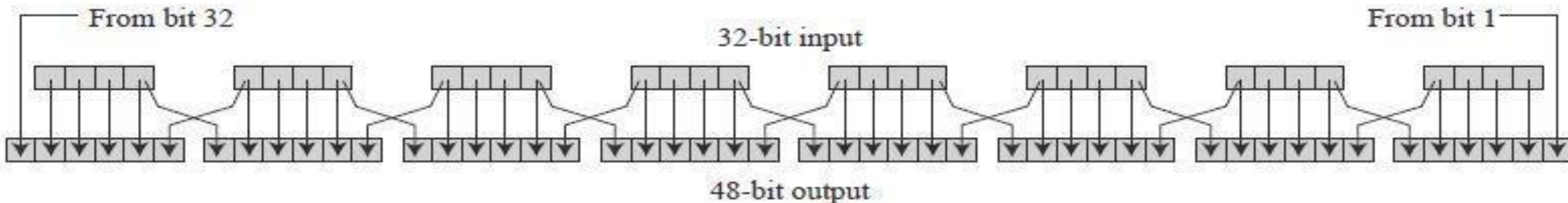


Figure 9: Expansion Permutation.

Although the relationship between the input and output can be defined mathematically, DES uses **Table 3** to define this P-box. Note that the number of output ports is 48, but the value range is only 1 to 32. Some of the inputs go to more than one output. For example, the value of input bit 5 becomes the value of output bits 6 and 8.

Table 3: Expansion P-box 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

Whitener (XOR)

After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key. Note that both the right section and the key are 48-bits in length. Also note that the round key is used only in this operation.

S-Boxes

The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. See **Figure 10**.

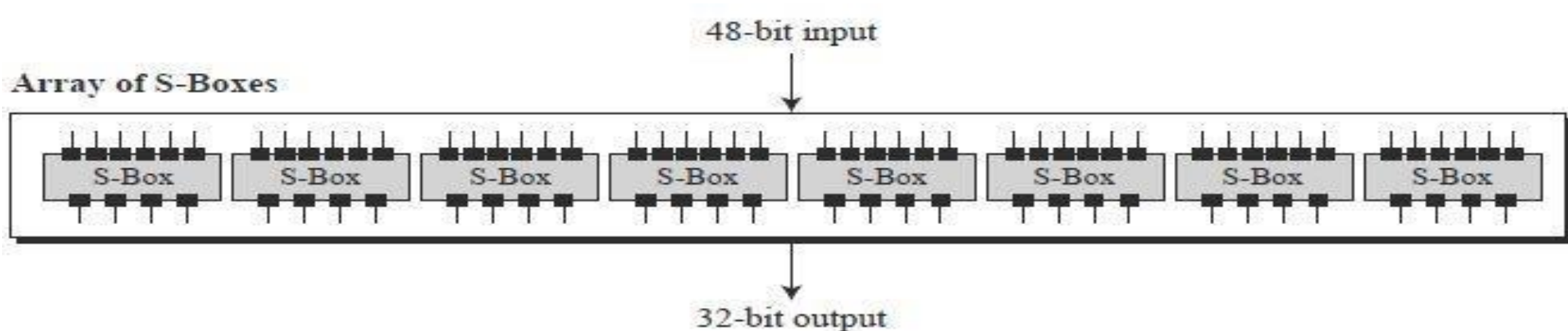


Figure 10: S-boxes.

S-Boxes

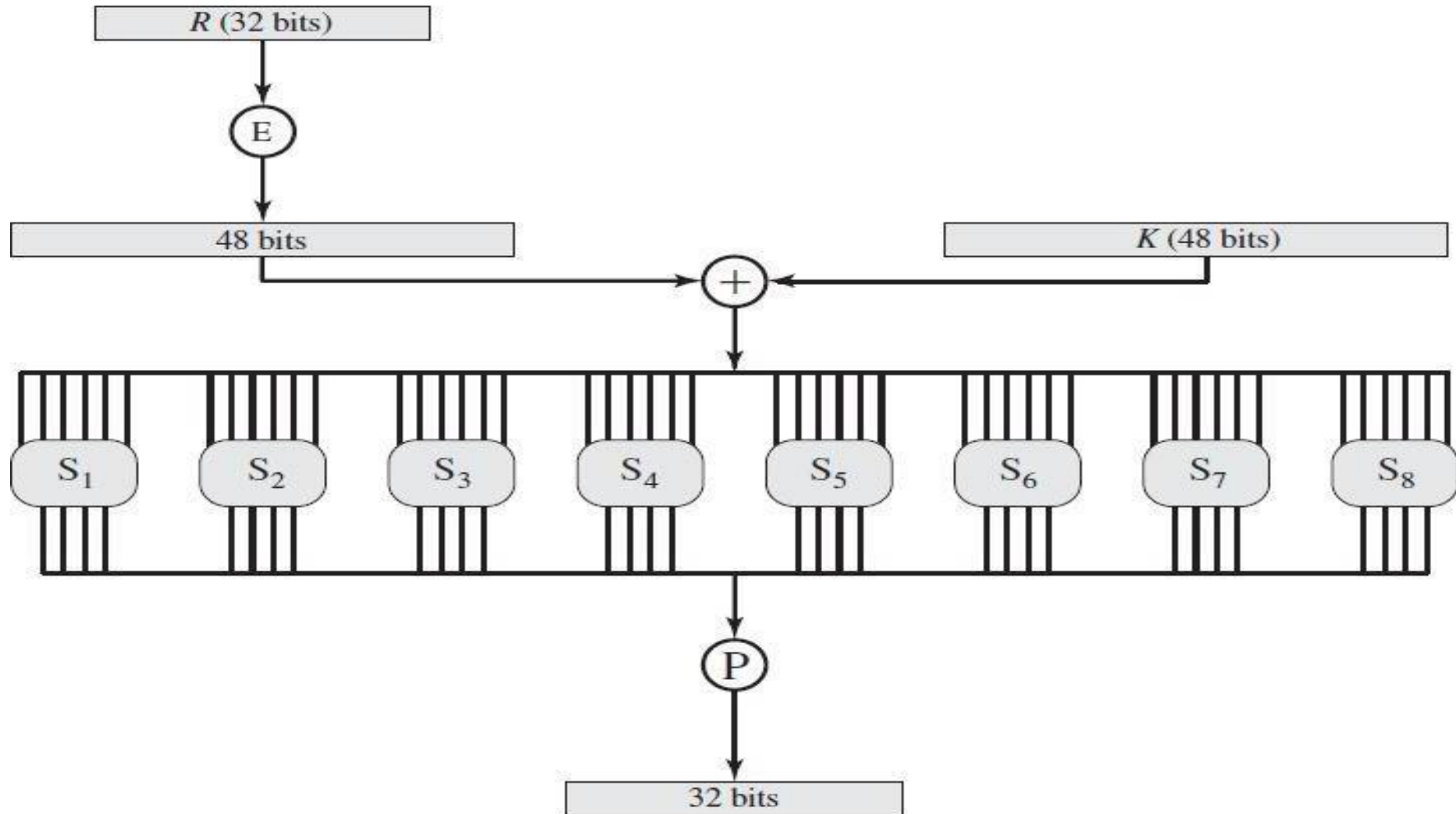


Figure 11: S-boxes.

S-Boxes

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; when these are combined the result is a 32-bit text. The substitution in each box follows a pre-determined rule based on a 4-row by the 16-column table. 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 as shown in **Figure 12**. This will become clear in the examples.

Because each S-box has its own table, we need eight tables, as shown in **Tables 4 to 11**, to define the output of these boxes. The values of the inputs (row number and column number) and the values of the outputs are given as decimal numbers to save space. These need to be changed to binary.

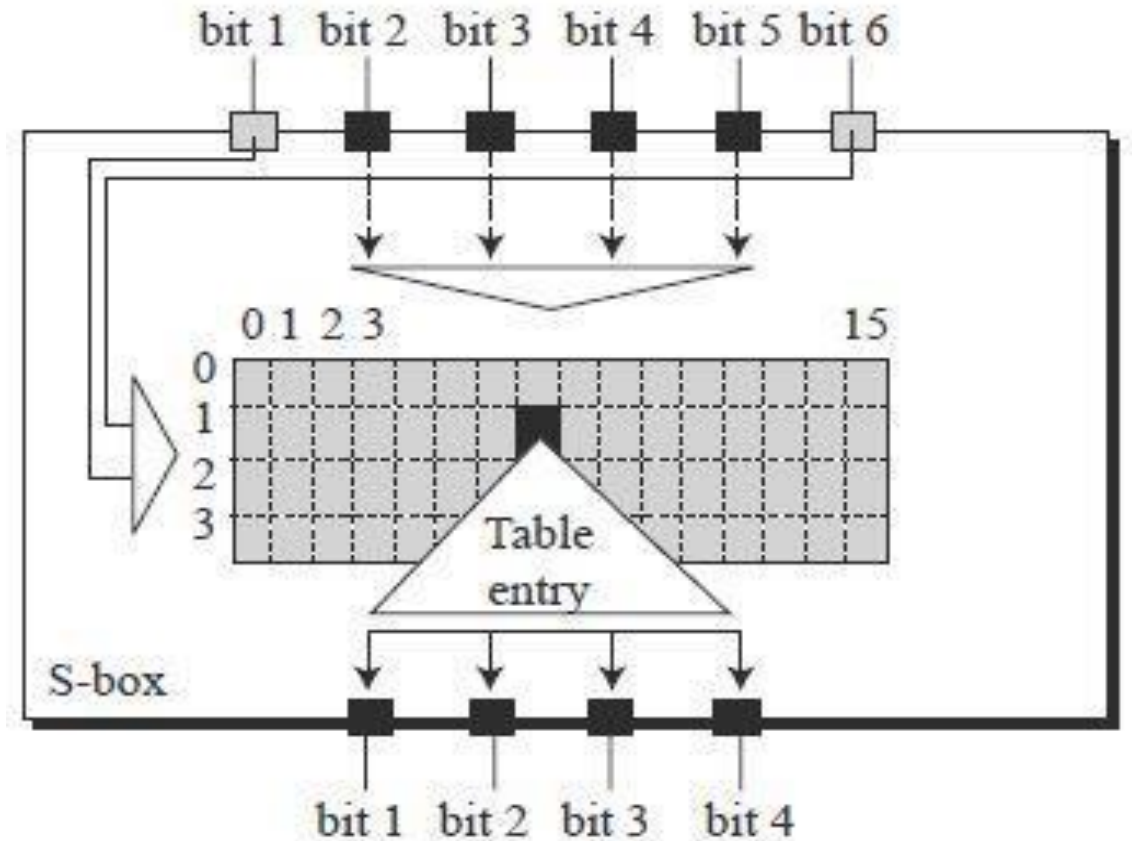


Figure 12: S-box rule.

S-Boxes

Table 4: 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

Table 5: 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-Boxes

Table 6: 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

Table 7: 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-Boxes

Table 8: 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

Table 9: 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-Boxes

Table 10: 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

Table 11: 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

Permutation Function (P) or Straight P-box

The last operation in the DES function is a permutation with a 32-bit input and a 32-bit output. The input/output relationship for this operation is shown in **Table 12** and follows the same general rule as previous tables. For example, the seventh bit of the input becomes the second bit of the output.

Table 12: Permutation Function (P) or Straight P-box.

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

Single Round of DES Algorithm

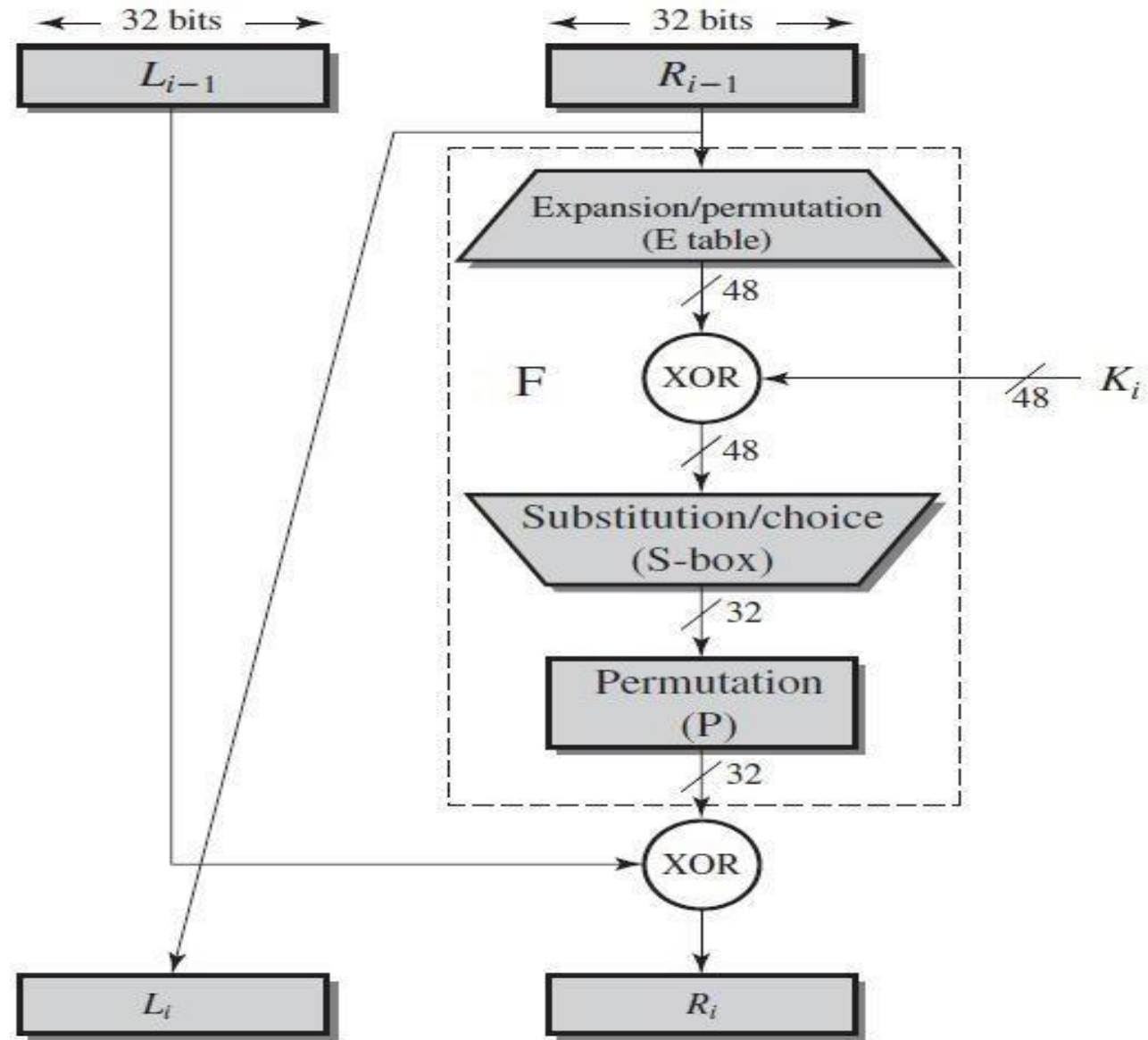


Figure 13: Single Round of DES Algorithm.