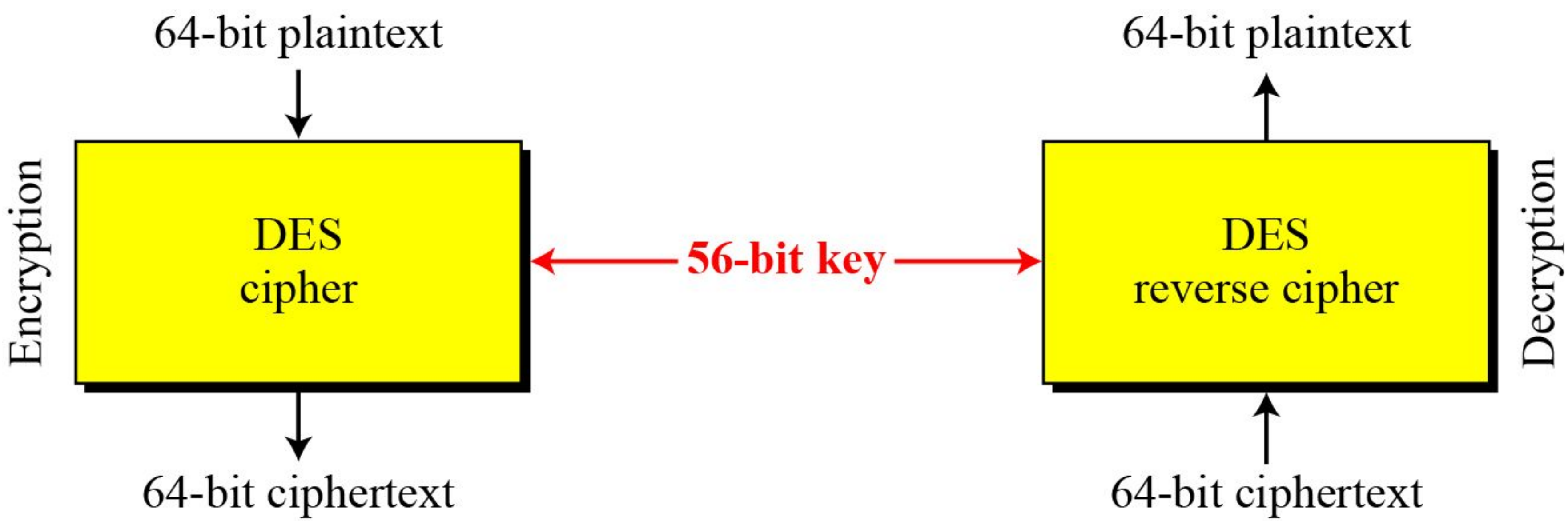
6.1.2 Overview

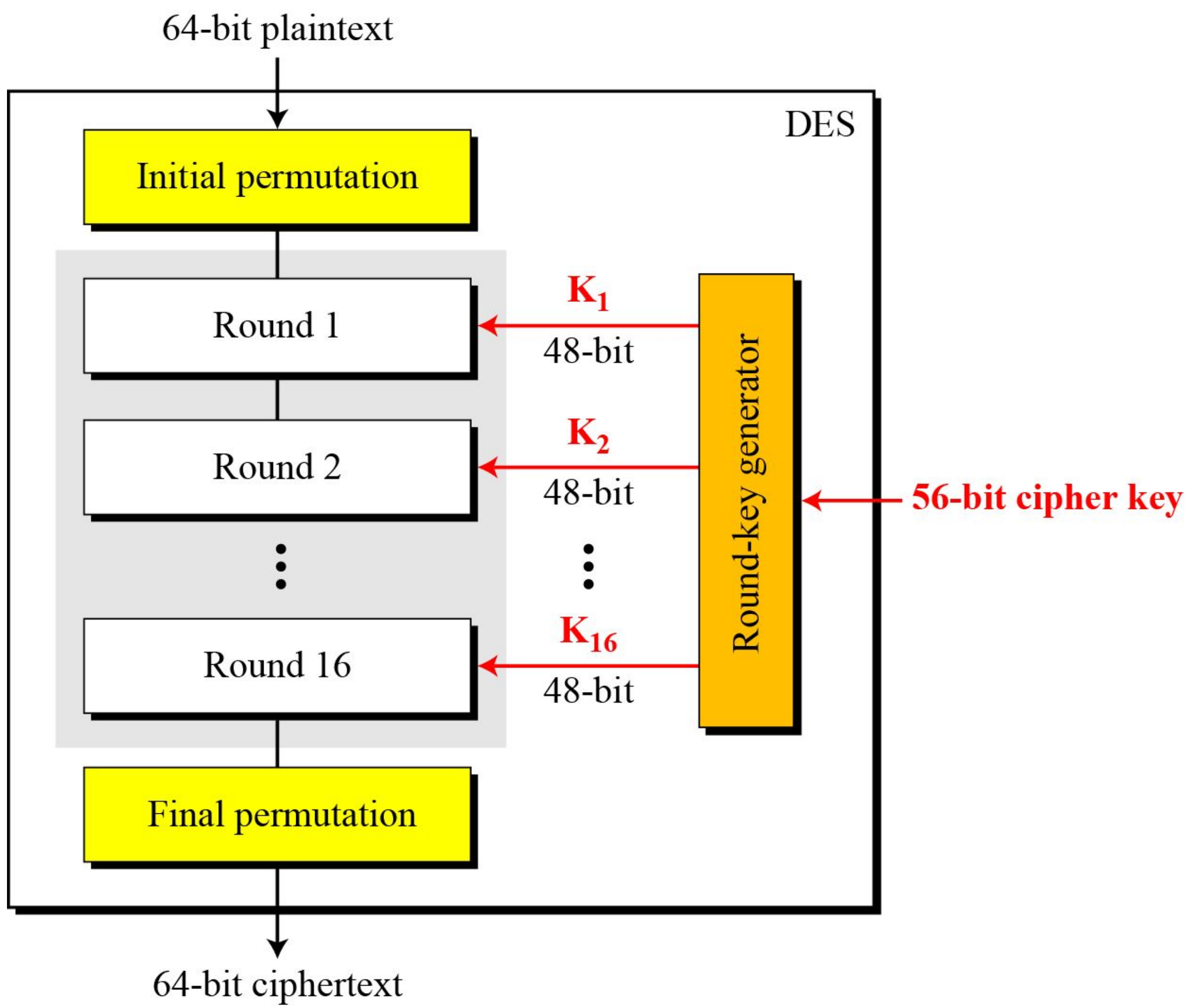
DES is a block cipher, as shown in Figure 6.1.

Figure 6.1 *Encryption and decryption with DES*



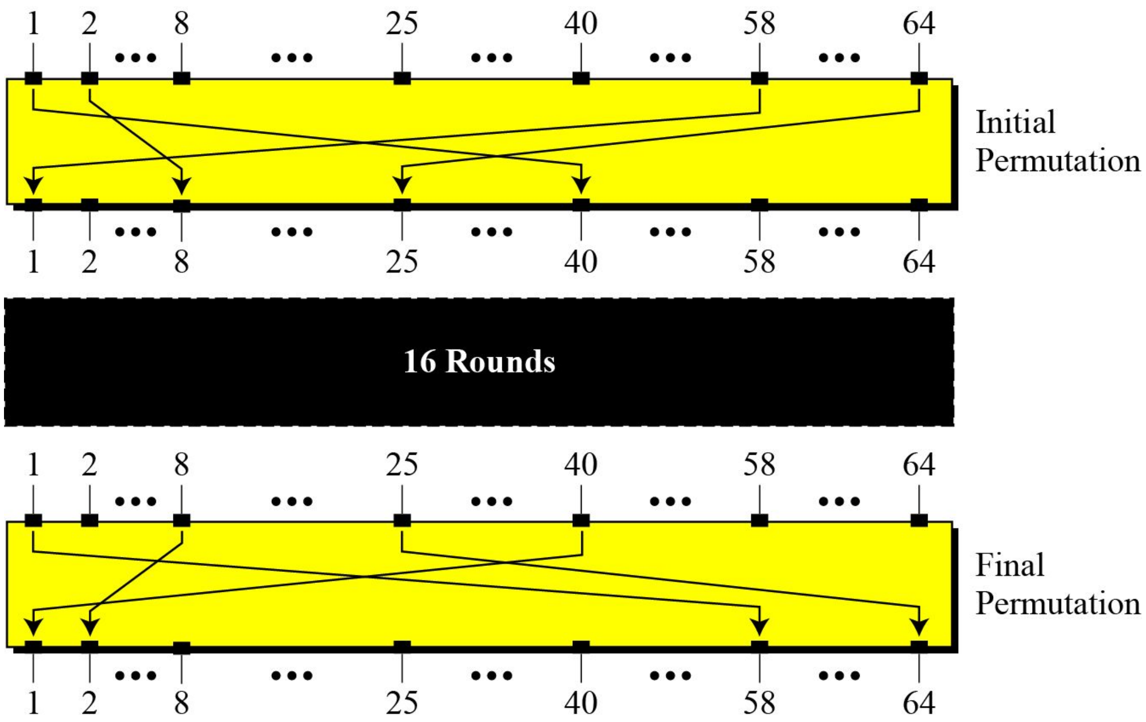
DES Structure

The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Feistel rounds.



6.2.1 Initial and Final Permutations

Figure 6.3 ***Initial and final permutation steps in DES***



6.2.1 Continue

Table 6.1 ***Initial and final permutation tables***

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

How to read this table?

The 58^{th} bit of input x will be the first bit of output IP(x), the 50^{th} bit of x is the second bit of IP(x), etc.

The initial and final permutations are straight P-boxes that are inverses of each other. They have no cryptography significance in DES.

6.2.1 Continued

Example 6.1

Find the output of the initial permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

Solution

Only bit 25 and bit 64 are 1s; the other bits are 0s. In the final permutation, bit 25 becomes bit 64 and bit 63 becomes bit 15. The result is

0x0002 0000 0000 0001

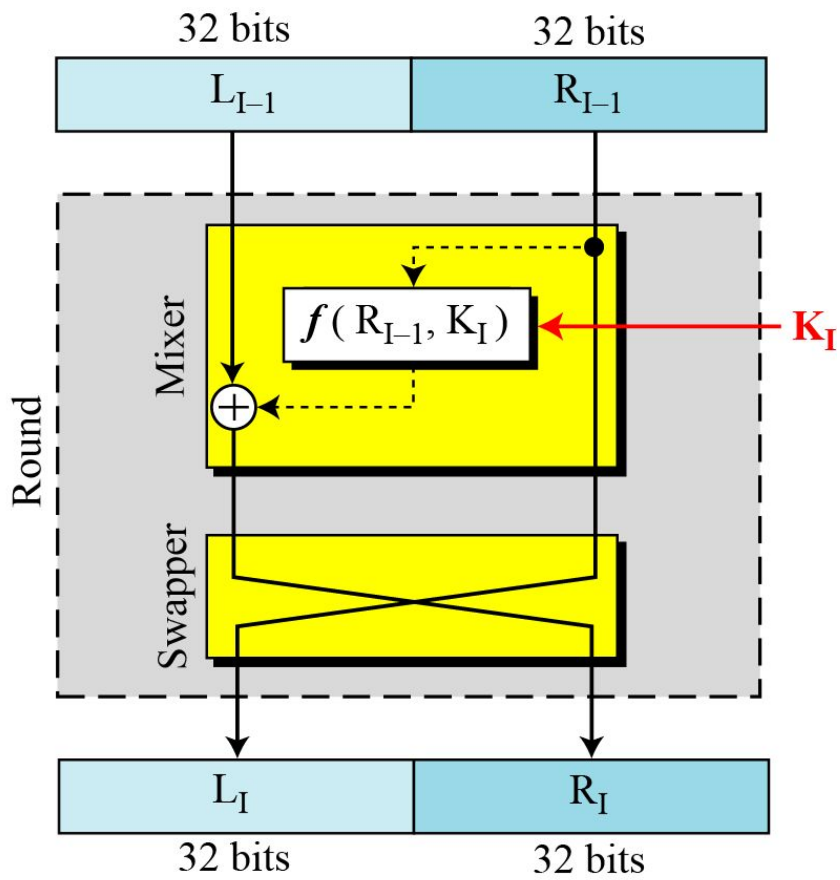
6.2.2 Rounds

Figure 6.4
A round in DES (encryption site)

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

–Separate message block into two 32-bit halves, L_i and R_i

- –Introduce **confusion** by using a “complex” nonlinear function f
- –f has two inputs: R_i and a 48-bit round key, K_i
- –Introduce **diffusion** by “adding” L_i and the output of f



$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus f(R_i, K_{i+1})$$

6.2.2 Continued

DES Function

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

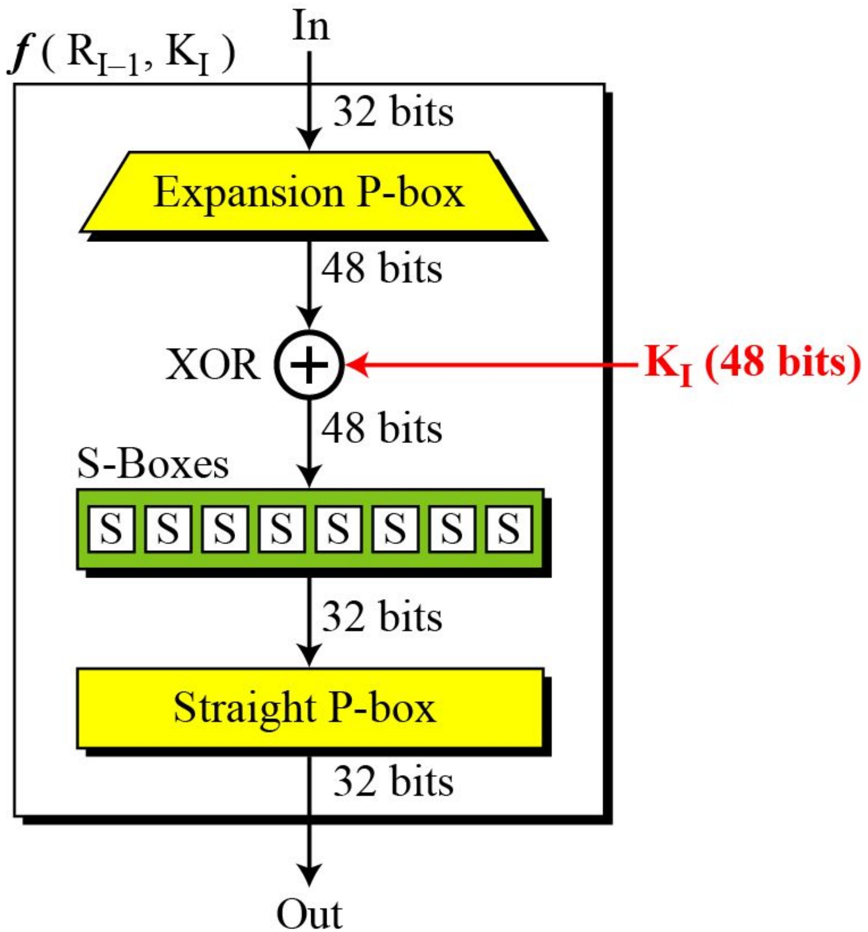


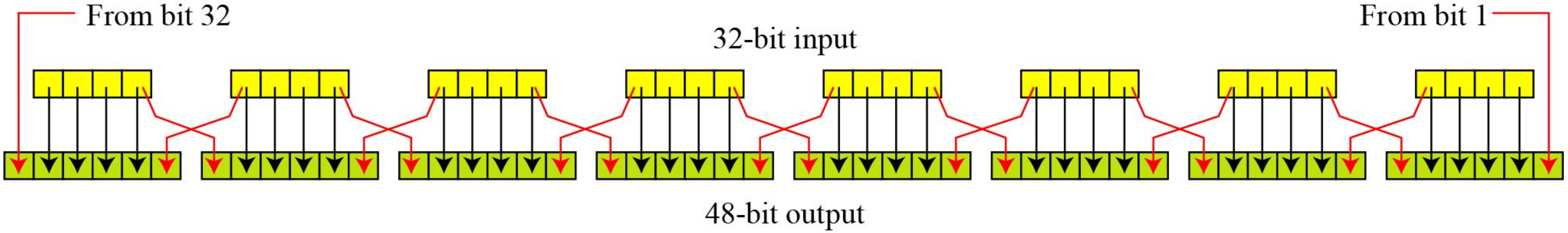
Figure 6.5
DES function

6.2.2 Continue

Expansion P-box

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.

Figure 6.6 *Expansion permutation*



6.2.2 Continue

Although the relationship between the input and output can be defined mathematically, DES uses Table 6.2 to define this P-box.

Table 6.6 ***Expansion P-box table***

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

6.2.2 Continue

Whitener (XOR)

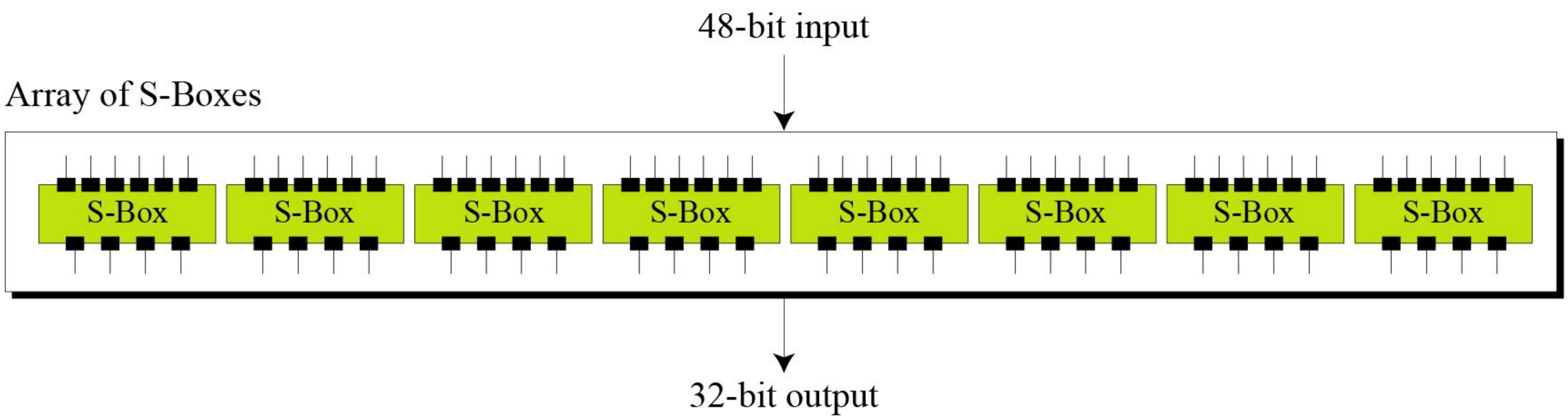
After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key. Note that both the right section and the key are 48-bits in length. Also note that the round key is used only in this operation.

6.2.2 Continue

S-Boxes

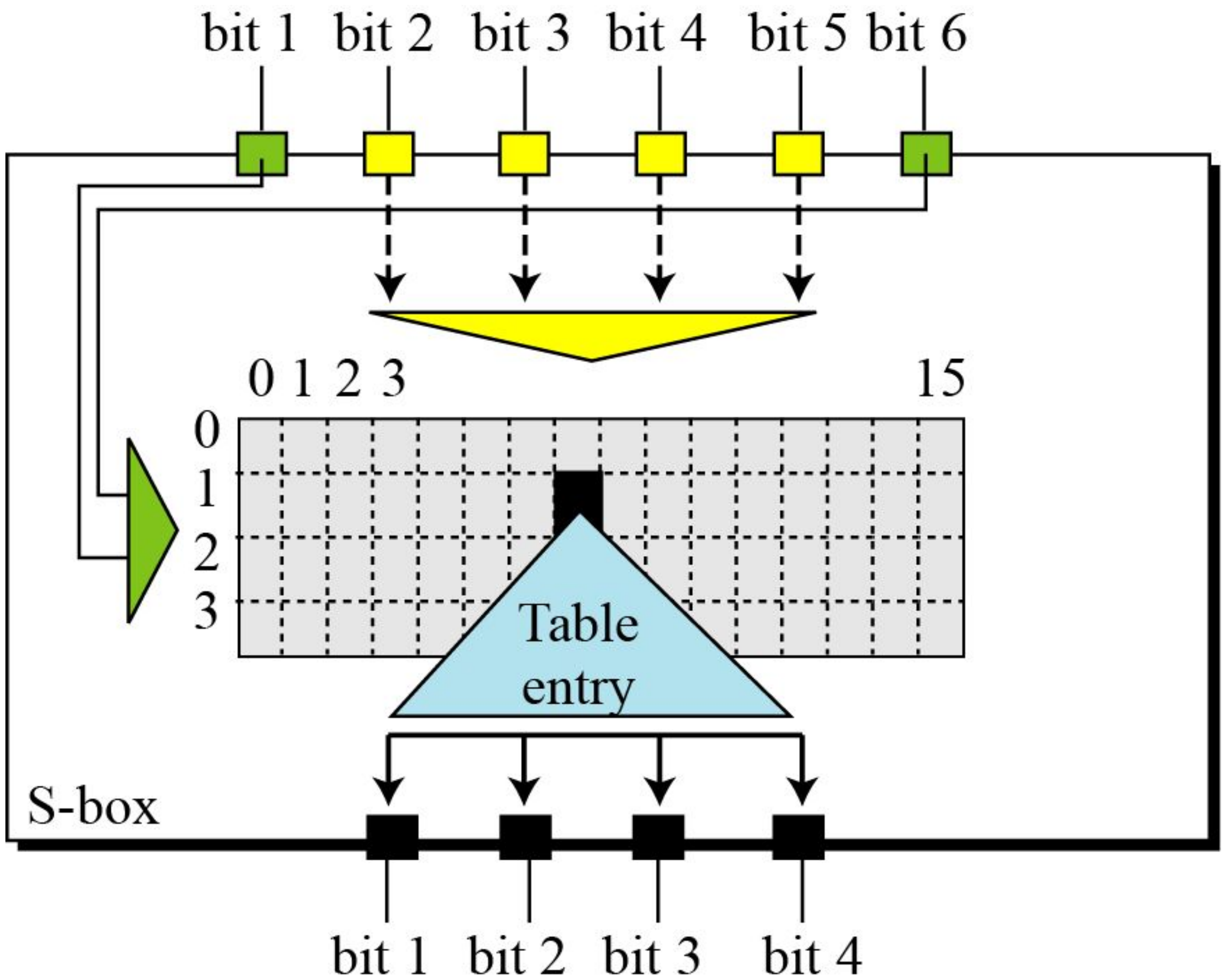
The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. See Figure 6.7.

Figure 6.7 *S-boxes*



6.2.2 Continue

Figure 6.8 ***S-box rule***



6.2.2 Continue

Table 6.3 shows the permutation for S-box 1. For the rest of the boxes see the textbook.

Table 6.3 ***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

6.2.2 Continued

Example 6.3

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

Solution

If we write the first and the sixth bits together, we get 11 in binary, which is 3 in decimal. The remaining bits are 0001 in binary, which is 1 in decimal. We look for the value in row 3, column 1, in Table 6.3 (S-box 1). The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

6.2.2 Continue

Straight Permutation

Table 6.11 *Straight permutation table*

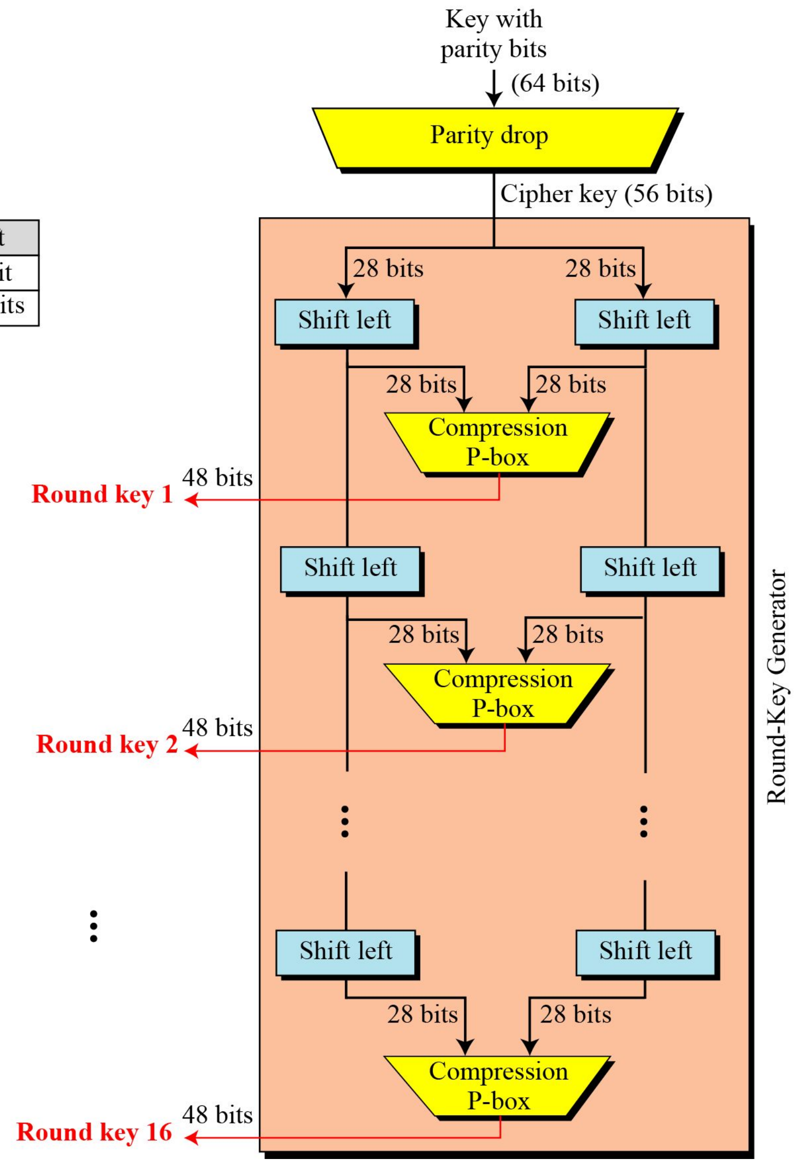
16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

6.2.3 Key Generation

Figure 6.10
Key generation

Shifting

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



The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key.

6.2.3 Continued

Table 6.12 *Parity-bit drop table*

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

64 □ 56

Table 6.13 *Number of bits shifts*

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

6.2.3 Continued

Table 6.14 *Key-compression table*

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

56 □ 48

6.3.1 Properties

Two desired properties of a block cipher are the avalanche effect and the completeness.

Example 6.7

To check the avalanche effect in DES, let us encrypt two plaintext blocks (with the same key) that differ only in one bit and observe the differences in the number of bits in each round.

Plaintext: 0000000000000000 Key: 22234512987ABB23
Ciphertext: 4789FD476E82A5F1

Plaintext: 000000000000000**1** Key: 22234512987ABB23
Ciphertext: 0A4ED5C15A63FEA3

6.3.1 *Continued*

Example 6.7 *Continued*

Although the two plaintext blocks differ only in the rightmost bit, the ciphertext blocks differ in 29 bits. This means that changing approximately 1.5 percent of the plaintext creates a change of approximately 45 percent in the ciphertext.

Table 6.17 *Number of bit differences for Example 6.7*

Rounds	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
Bit differences	1	6	20	29	30	33	32	29	32	39	33	28	30	31	30	29

6.3.1 Continued

Completeness effect

Completeness effect means that each bit of the ciphertext needs to depend on many bits on the plaintext.

6.3.2 Design Criteria

S-Boxes

The design provides confusion and diffusion of bits from each round to the next.

P-Boxes

They provide diffusion of bits.

Number of Rounds

DES uses sixteen rounds of Feistel ciphers. the ciphertext is thoroughly a random function of plaintext and ciphertext.

6.3.3 DES Weaknesses

During the last few years critics have found some weaknesses in DES.

Weaknesses in Cipher Design

1. ***Weaknesses in S-boxes***

- *Two specifically chosen inputs to an S-box can create same output*

2. ***Weaknesses in P-boxes***

- *initial and final permutations have no security benefits*
- *the first and fourth bits of every 4-bit series are repeated*

3. ***Weaknesses in Key***

- *Weak keys create same 16 round keys*
- *Semi-weak keys create 2 different round keys*
- *Possible weak keys create 4 distinct round keys*
- *Key complement*

6.3.3 DES Weaknesses

- *There are four weak keys.*
- *After parity drop operation, a key consists either of all 0s, all 1s, or half 0s and half 1s.*
- *Weak keys create same 16 round keys.*

Table 6.18 *Weak keys*

Keys before parities drop (64 bits)	*Actual key (56 bits)*
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

6.3.3 Continued

Example 6.8

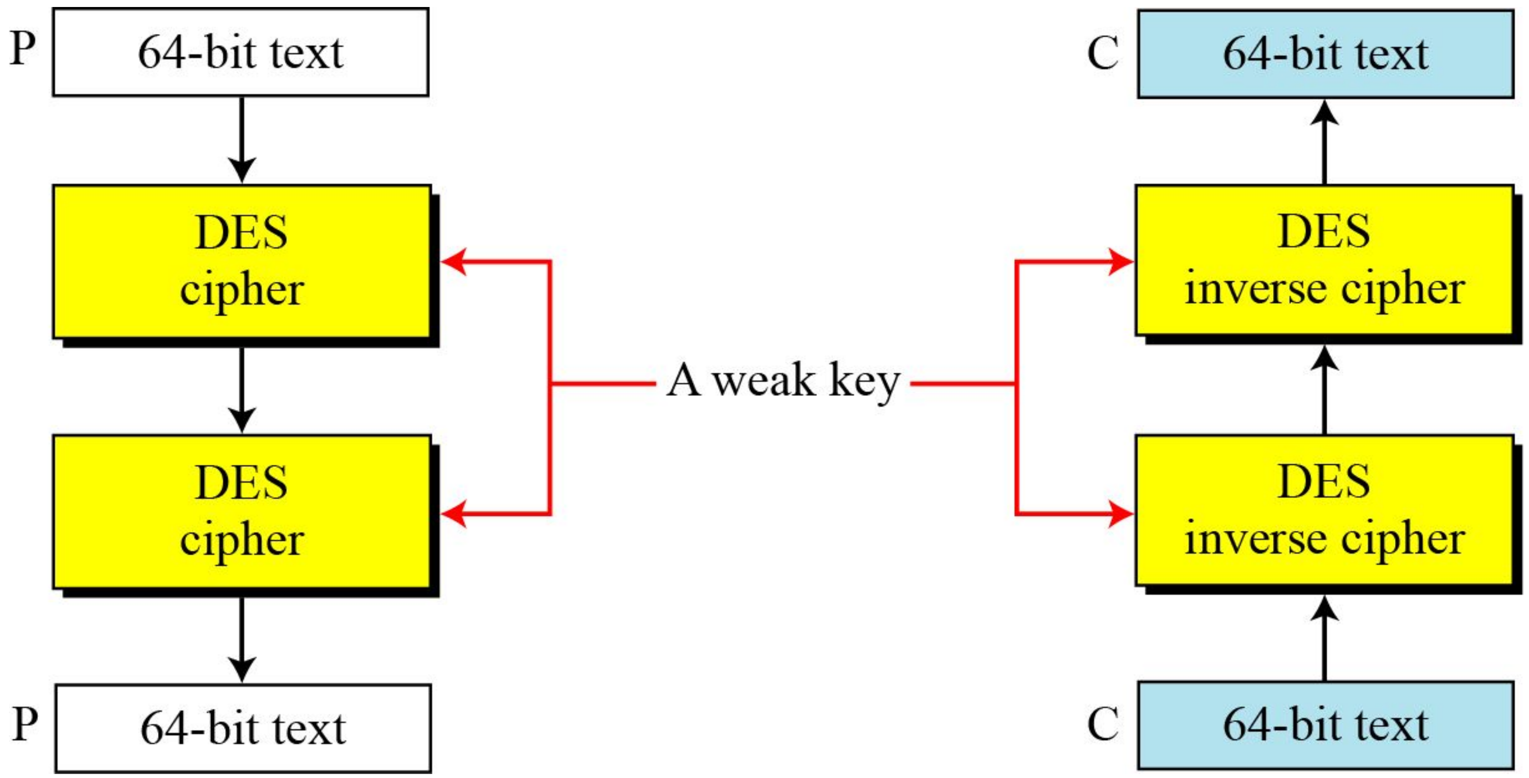
Let us try the first weak key in Table 6.18 to encrypt a block two times. After two encryptions
with the same key the original plaintext block is created. Note that we have used the encryption algorithm two times, not one encryption followed by another decryption.

Key: 0x0101010101010101
Plaintext: *0x1234567887654321* Ciphertext: 0x814FE938589154F7

Key: 0x0101010101010101
Plaintext: 0x814FE938589154F7 Ciphertext: *0x1234567887654321*

6.3.3 Continued

Figure 6.11 *Double encryption and decryption with a weak key*



$$E_k(E_k(P)) = P$$

6.3.3 Continued

Semi-weak keys create only 2 different round keys; k1, k2

Table 6.19 *Semi-weak keys*

First key in the pair	*Second key in the pair*
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
01E0 01E1 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

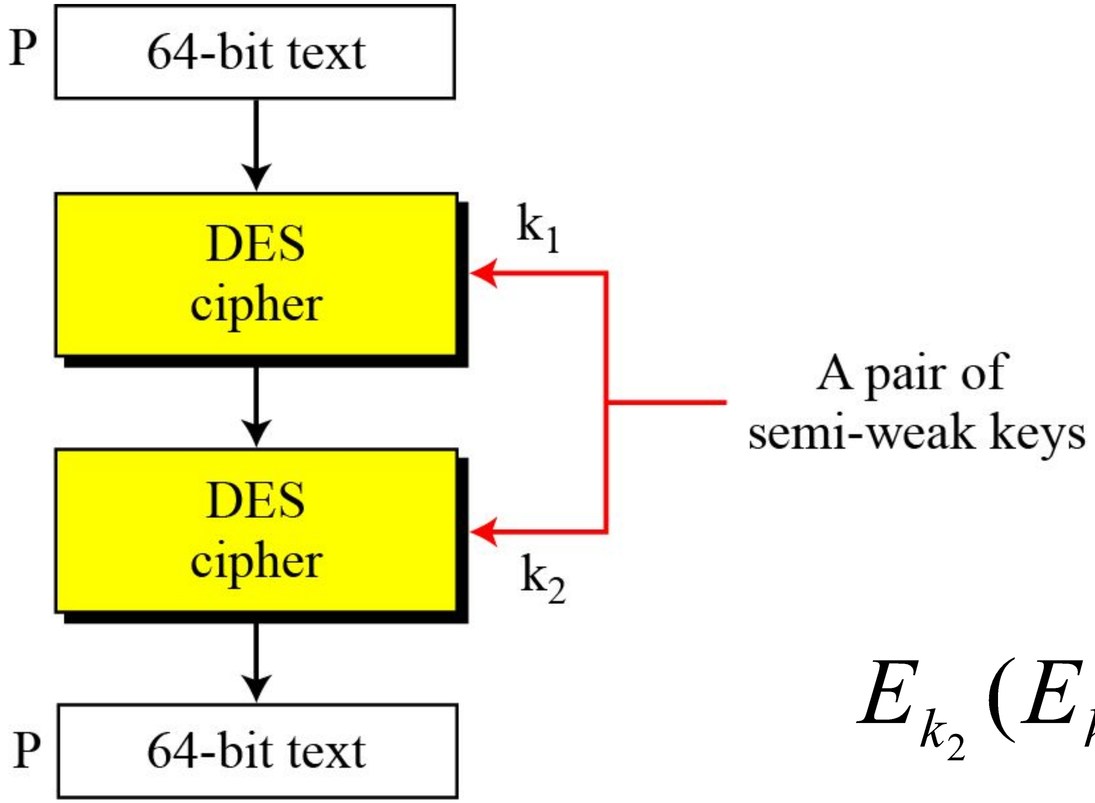
6.3.3 Continued

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

Semi-week keys create 2 different round keys

6.3.3 Continued

Figure 6.12 *A pair of semi-weak keys in encryption and decryption*



$$E_{k_2}(E_{k_1}(P)) = P$$

6.3.3 Continued

Example 6.9

What is the probability of randomly selecting a weak, a semi-weak, or a possible weak key?

Solution

DES has a key domain of 2^{56}. The total number of the above keys are 64 (4 + 12 + 48). The probability of choosing one of these keys is 8.8×10^{-16}, almost impossible.

6.3.3 Continued

Key Complement In the key domain (2^{56}), definitely half of the keys are complement of the other half. A **key complement** can be made by inverting (changing 0 to 1 or 1 to 0) each bit in the key. Does a key complement simplify the job of the cryptanalysis? It happens that it does. Eve can only half of the possible keys (2^{55}) to perform brute-force attack. This is because

$$C = E(K, P) \rightarrow \overline{C} = E(\overline{K}, \overline{P})$$

In other words, if we encrypt the complement of plaintext with the complement of the key, we get the complement of the ciphertext. Eve does not have to test all 2^{56} possible keys, she can test only half of them and then complement the result.

6.3.3 Continued

Example 6.10

Let us test the claim about the complement keys. We have used an arbitrary key and plaintext to find the corresponding ciphertext. If we have the key complement and the plaintext, we can obtain the complement of the previous ciphertext (Table 6.20).

Table 6.20 *Results for Example 6.10*

	Original	*Complement*
Key	1234123412341234	EDCBEDCBEDCBEDCB
Plaintext	12345678ABCDEF12	EDCBA987543210ED
Ciphertext	E112BE1DEFC7A367	1EED41E210385C98