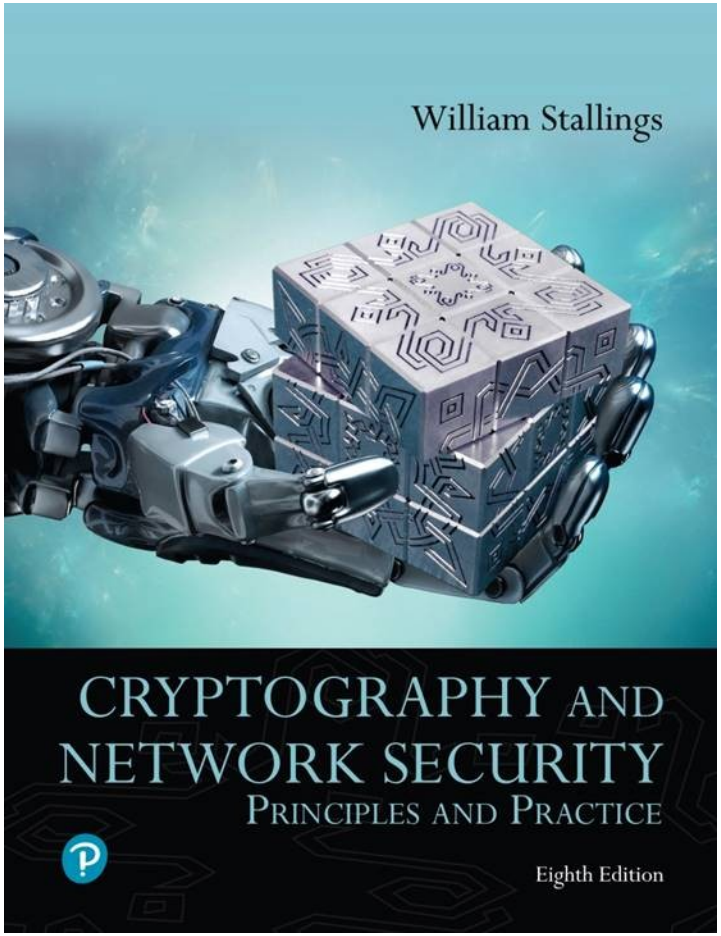
Cryptography and Network Security: Principles and Practice

Eighth Edition



Chapter 4

Block Ciphers and the Data Encryption Standard

Stream Cipher (1 of 2)

- Encrypts a digital data stream one bit or one byte at a time.
  - Examples:
    - Autokeyed Vigenère cipher
    - Vernam cipher
- In the ideal case, a one-time pad version of the Vernam cipher would be used, in which the key stream is as long as the plaintext bit stream.
  - If the cryptographic keystream is random, then this cipher is unbreakable by any means other than acquiring the key stream.
    - Key stream must be provided to both users in advance via some independent and secure channel
    - This introduces insurmountable logistical problems if the intended data traffic is very large

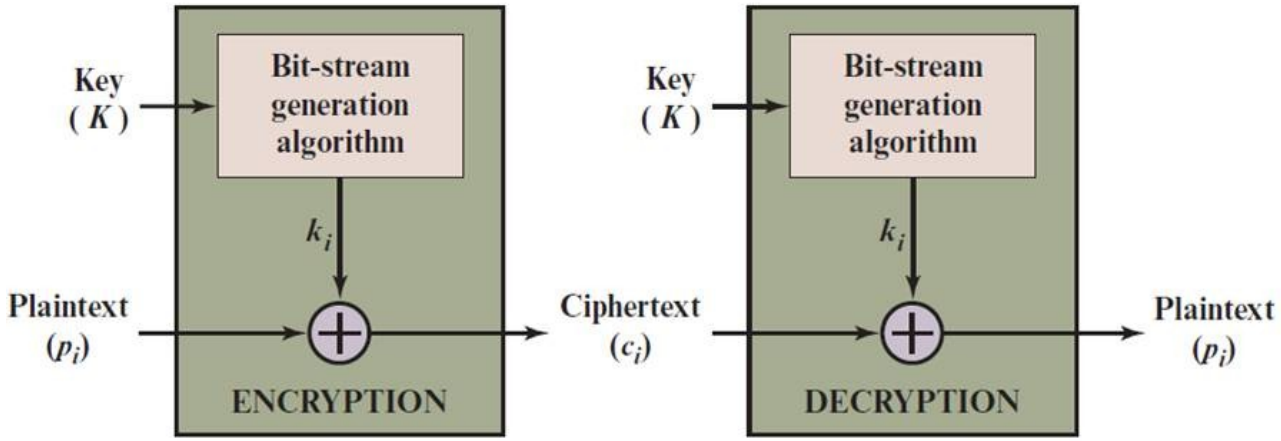
Stream Cipher (2 of 2)

- For practical reasons, the bit-stream generator must be implemented as an algorithmic procedure so that the cryptographic bit stream can be produced by both users.
  - It must be computationally impractical to predict future portions of the bit stream based on previous portions of the bit stream.
  - The two users need only share the generating key and each can produce the key stream..

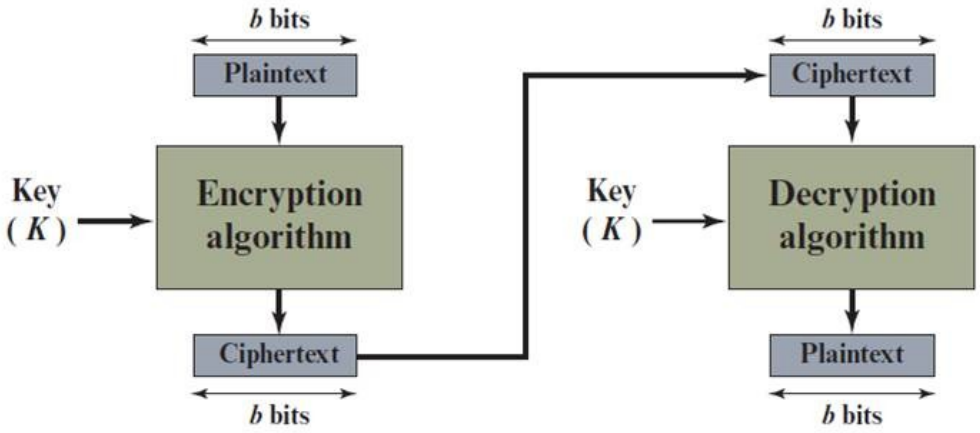
Block Cipher

- A block of plaintext is treated as a whole and used to produce a ciphertext block of equal length
- Typically, a block size of 64 or 128 bits is used
- As with a stream cipher, the two users share a symmetric encryption key.
- Most network-based symmetric cryptographic applications make use of block ciphers.

Figure 4.1 Stream Cipher and Block Cipher



(a) Stream cipher using algorithmic bit-stream generator



(b) Block cipher

Figure 4.2 General *n*-bit-*n*-bit Block Substitution (shown with *n* = 4)

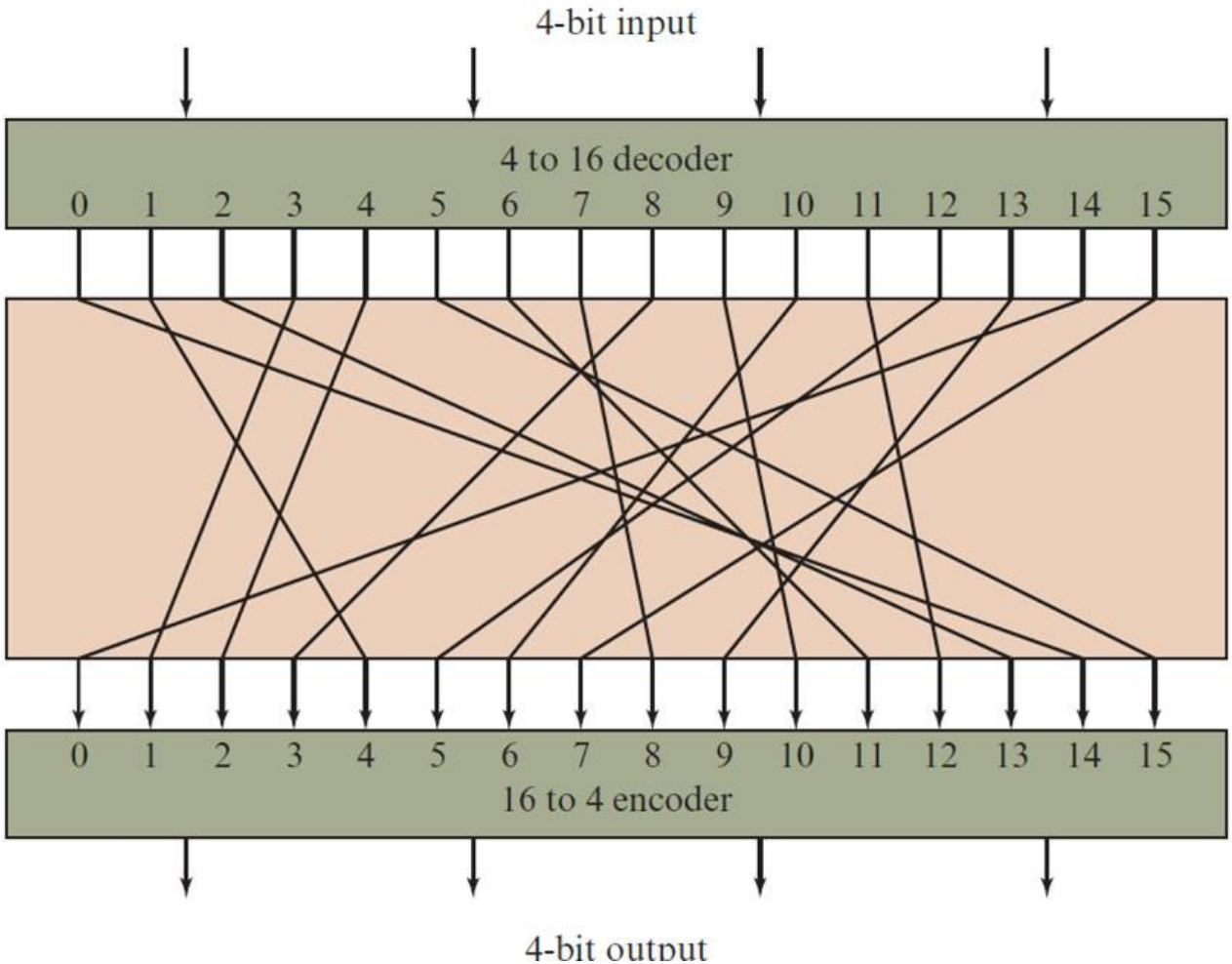


Table 4.1 Encryption and Decryption Tables for Substitution Cipher of Figure 4.2

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

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

Figure 5.2 *A transposition block cipher modeled as a permutation*

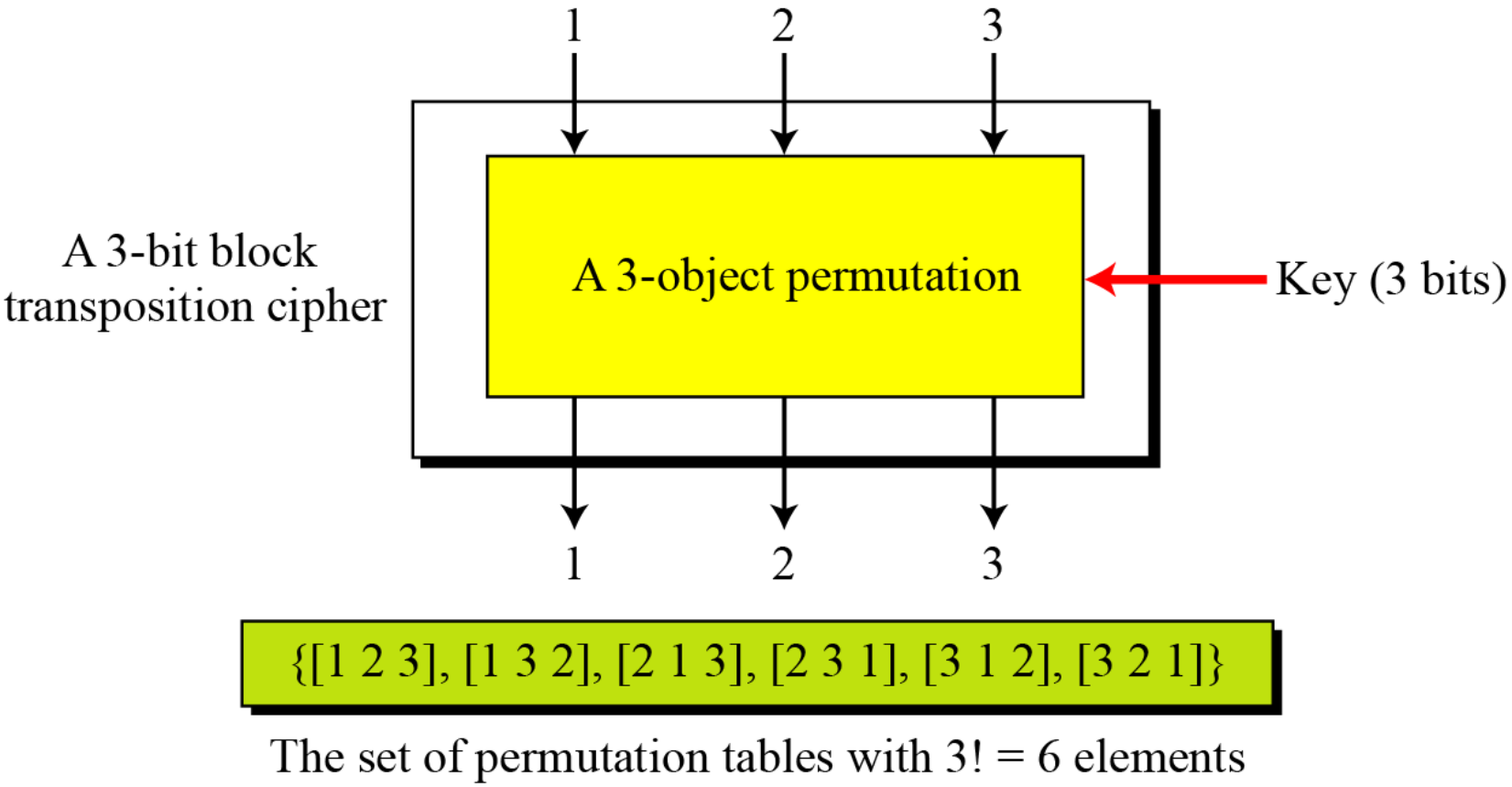
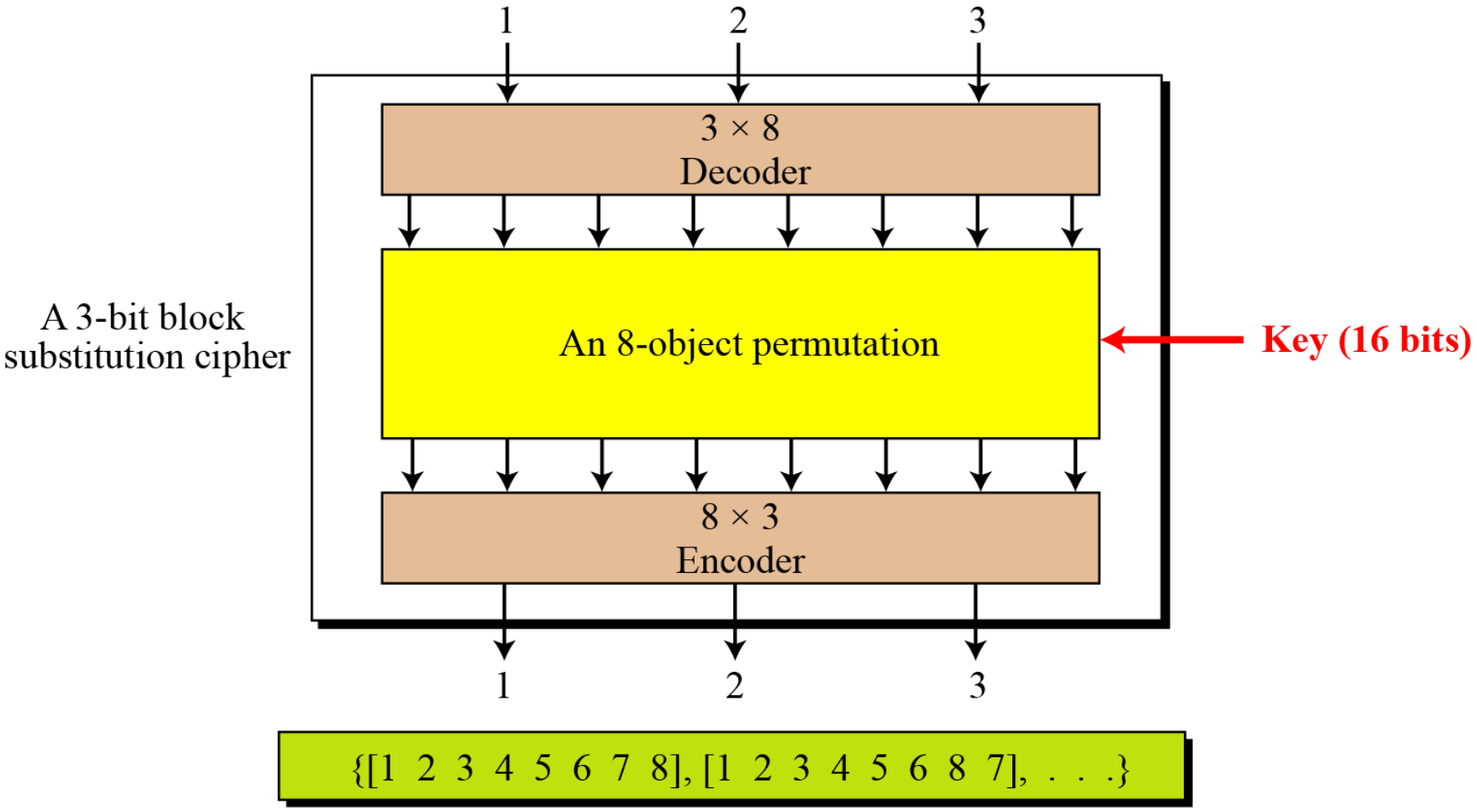


Figure 5.3 *A substitution block cipher model as a permutation*



Note

A full-size key *n*-bit transposition cipher or a substitution block cipher can be modeled as a permutation, but their key sizes are different:

- **Transposition: the key is $\lceil \log_2 n! \rceil$ bits long.**
- **Substitution: the key is $\lceil \log_2 (2n)! \rceil$ bits long.**

Note

A partial-key cipher is a group under the composition operation if it is a subgroup of the corresponding full-size key cipher.

5.1.1 Substitution or Transposition

A modern block cipher can be designed to act as a substitution cipher or a transposition cipher.

Note

To be resistant to exhaustive-search attack, a modern block cipher needs to be designed as a substitution cipher.

Components of a Modern Block Cipher

Modern block ciphers normally are keyed substitution ciphers in which the key allows only partial mappings from the possible inputs to the possible outputs.

P-Boxes

A P-box (permutation box) parallels the traditional transposition cipher for characters. It transposes bits.

S-Boxes

An S-box (substitution box) can be thought of as a miniature substitution cipher.

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.

Feistel Cipher

- Feistel proposed the use of a cipher that alternates substitutions and permutations
- Substitutions
  - Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements
- Permutation
  - No elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed
- Is a practical application of a proposal by Claude Shannon to develop a product cipher that alternates confusion and diffusion functions
- Is the structure used by many significant symmetric block ciphers currently in use

Diffusion and Confusion

- Terms introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system
  - Shannon's concern was to thwart cryptanalysis based on statistical analysis
- **Diffusion**
  - The statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext
  - This is achieved by having each plaintext digit affect the value of many ciphertext digits
- **Confusion**
  - Seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible
  - Even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key

5.1.4 Continued

Diffusion

The idea of diffusion is to hide the relationship between the ciphertext and the plaintext.

Note

Diffusion hides the relationship between the ciphertext and the plaintext.

5.1.4 Continued

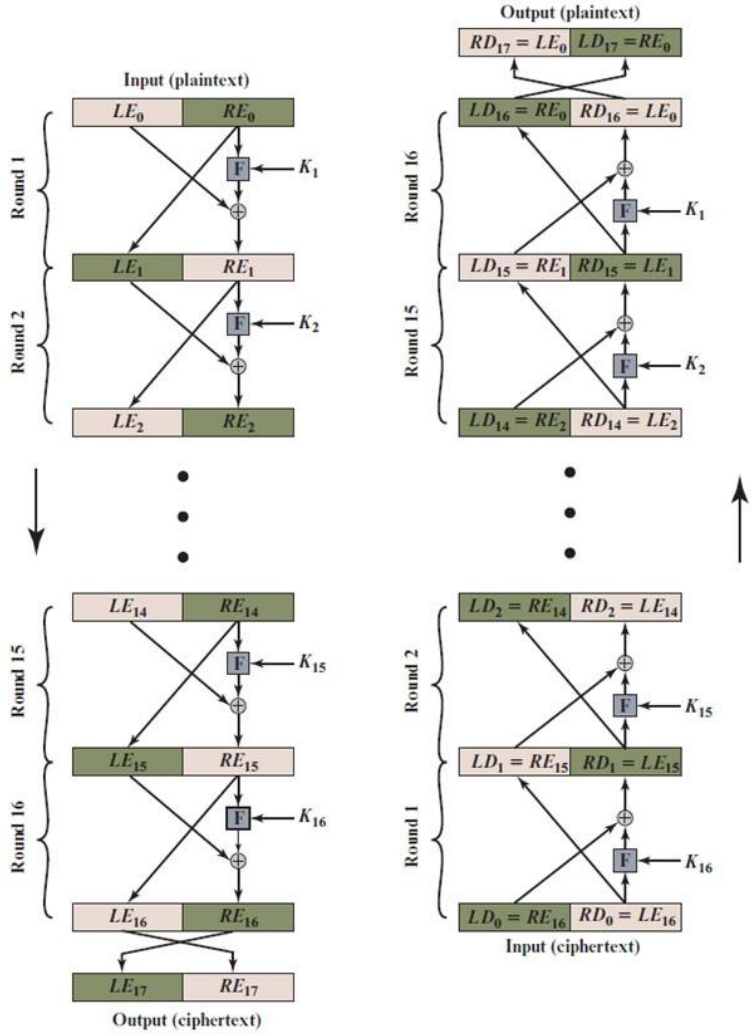
Confusion

The idea of confusion is to hide the relationship between the ciphertext and the key.

Note

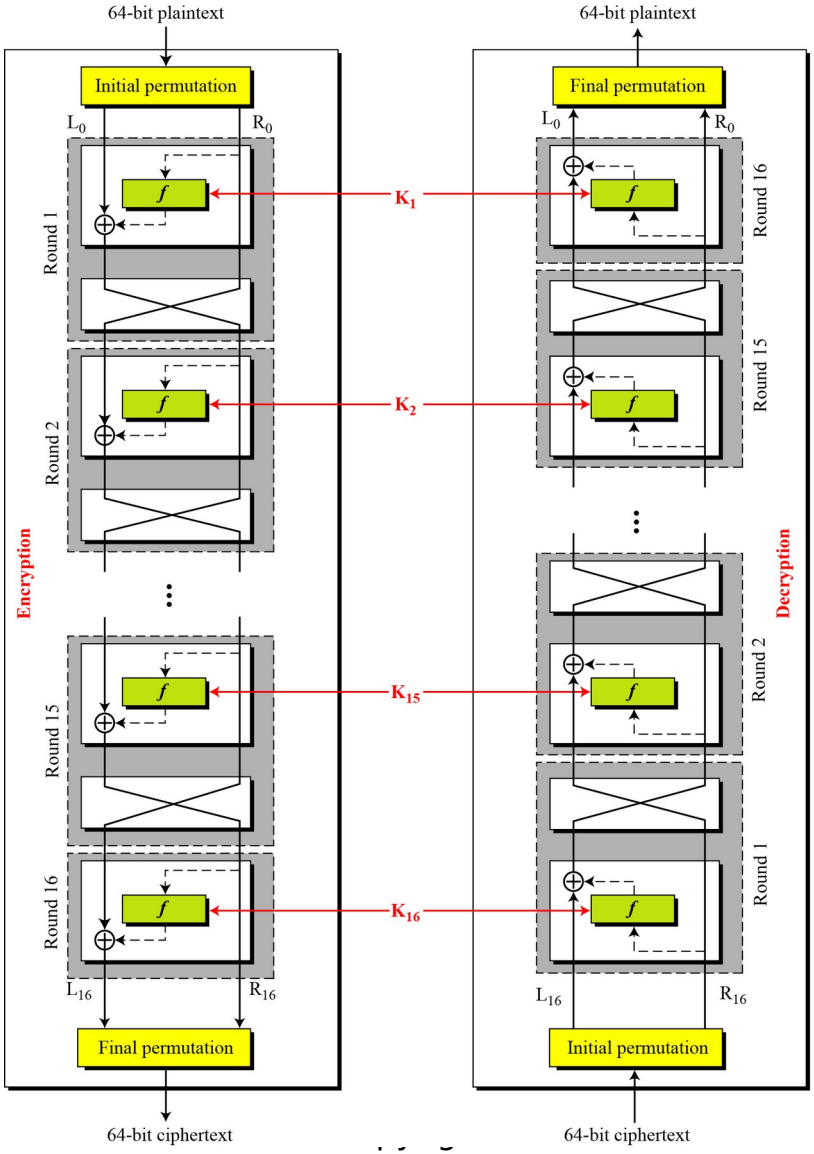
Confusion hides the relationship between the ciphertext and the key.

Figure 4.3 Feistel Encryption and Decryption (16 rounds)



6.2.3 Continued

Figure 6.9 *DES cipher and reverse cipher for the first approach*



6.2.3 Cipher and Reverse Cipher

Using mixers and swappers, we can create the cipher and reverse cipher, each having 16 rounds.

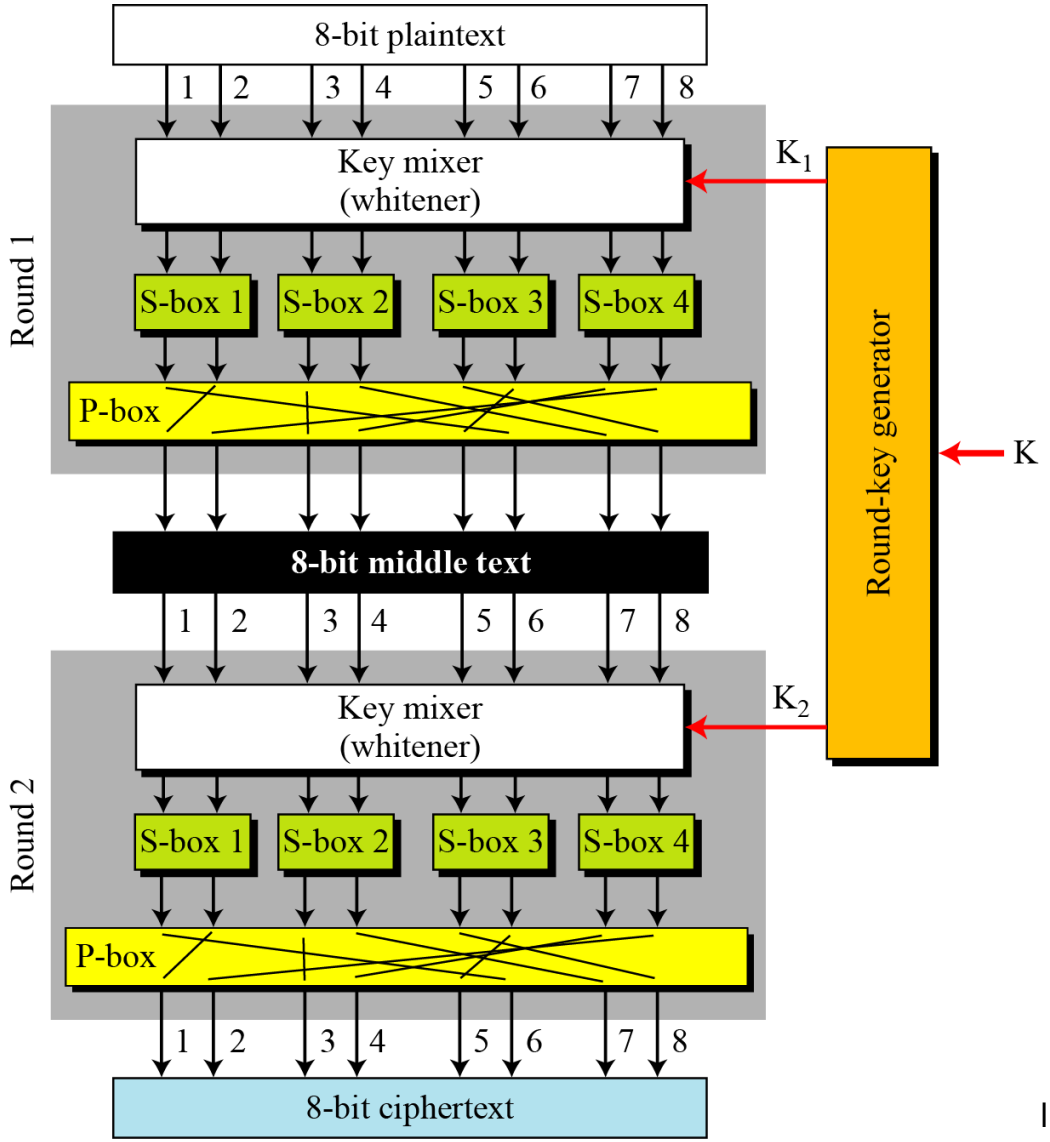
First Approach

To achieve this goal, one approach is to make the last round (round 16) different from the others; it has only a mixer and no swapper.

Note

In the first approach, there is no swapper in the last round.

Figure *A product cipher made of two rounds*



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.

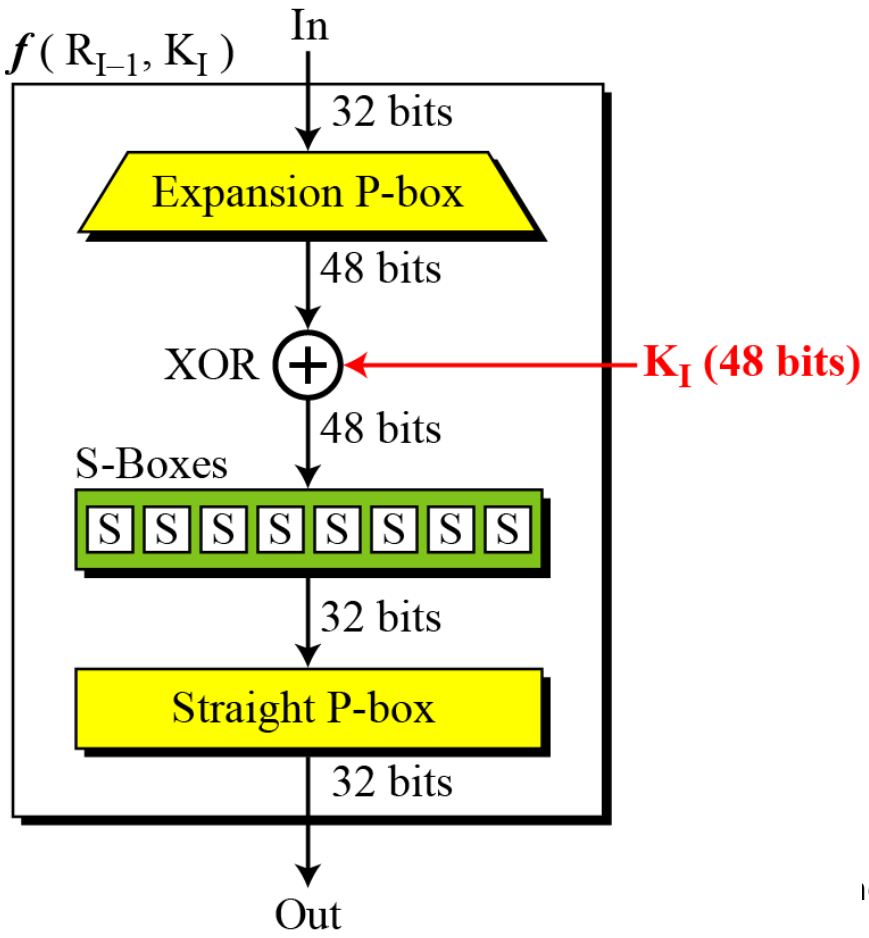
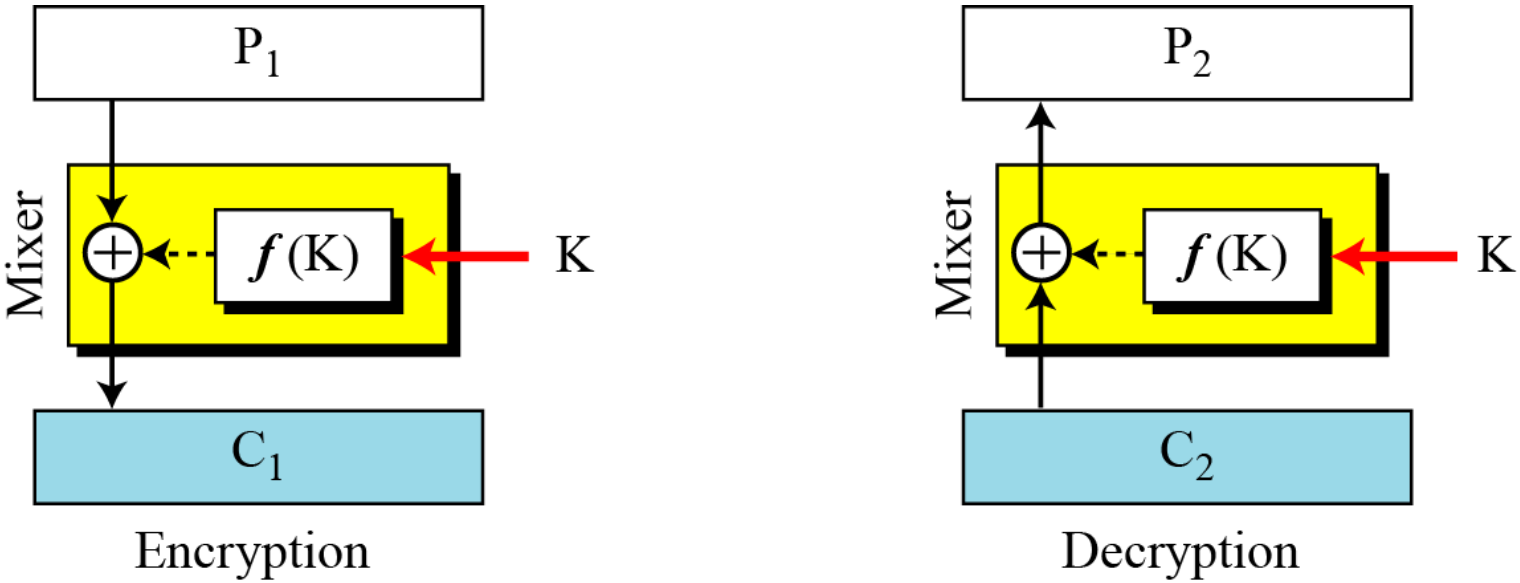


Figure 6.5
DES function

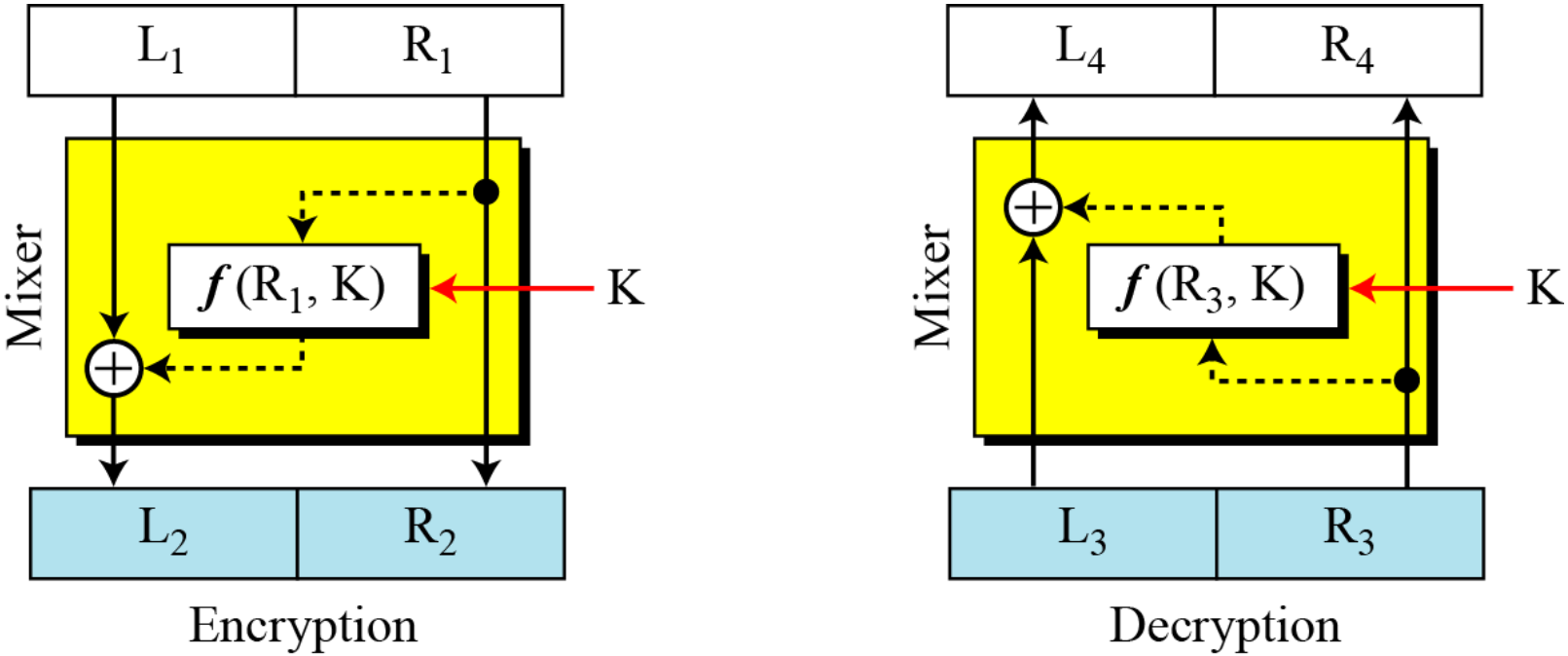
Figure *The first thought in Feistel cipher design*



Note

Diffusion hides the relationship between the ciphertext and the plaintext.

Figure *Improvement of the previous Feistel design*



6.2.2 Rounds

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

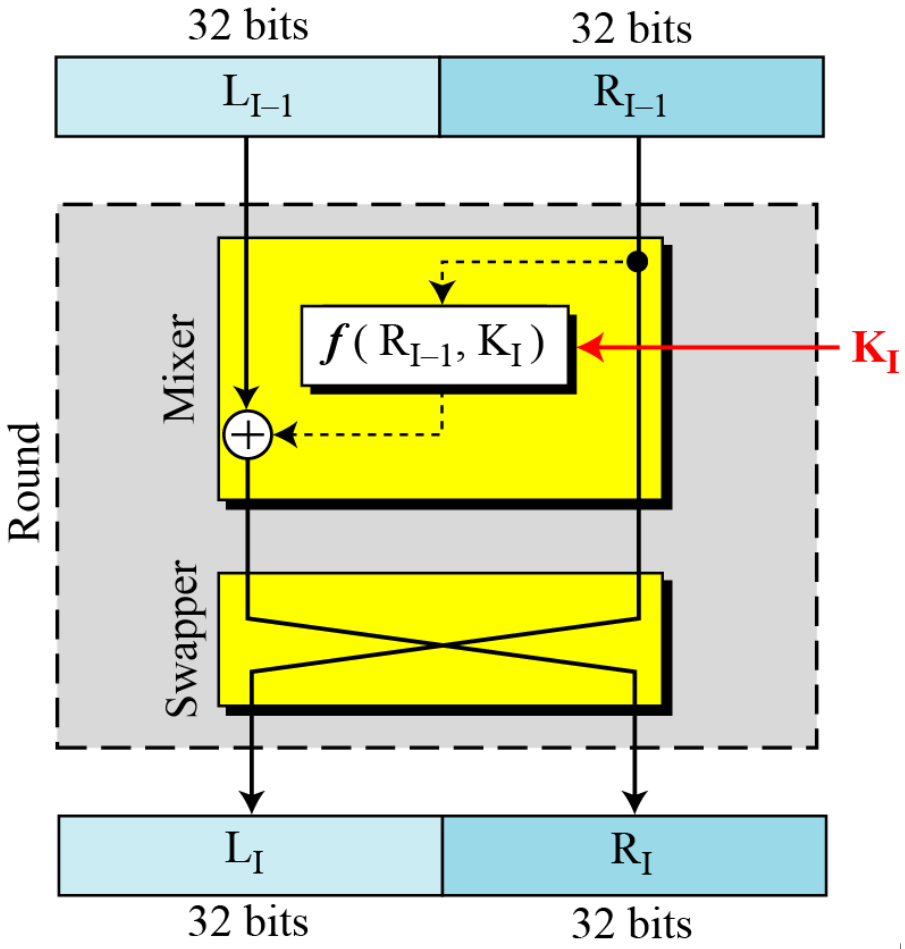


Figure 6.4
A round in DES (encryption site)

Feistel Example

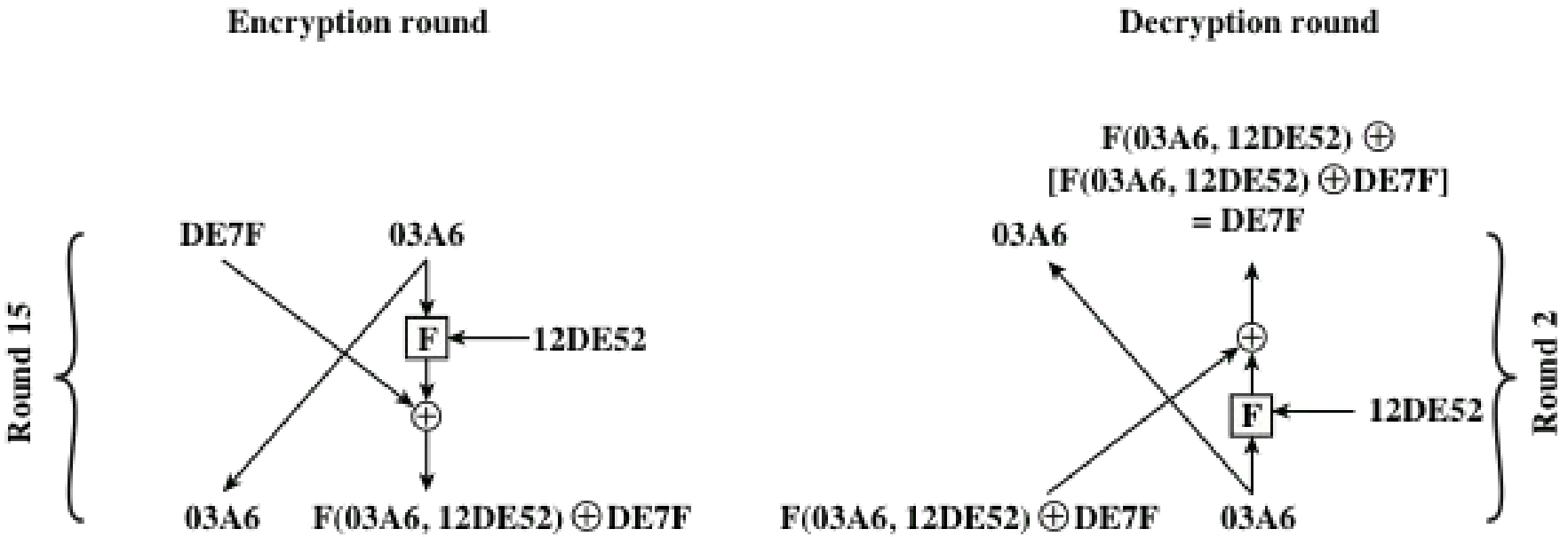
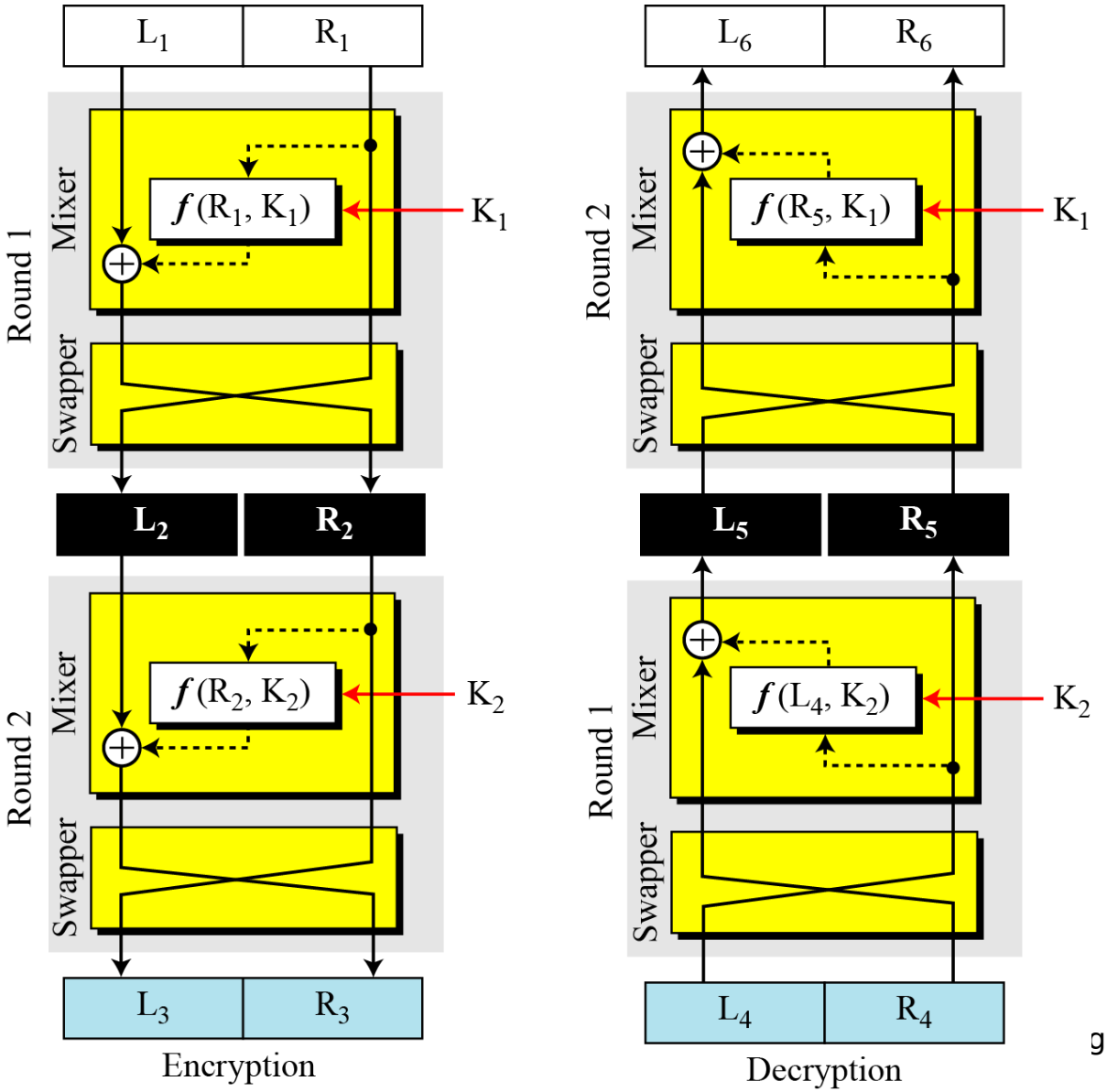


Figure 4.4 Feistel Example

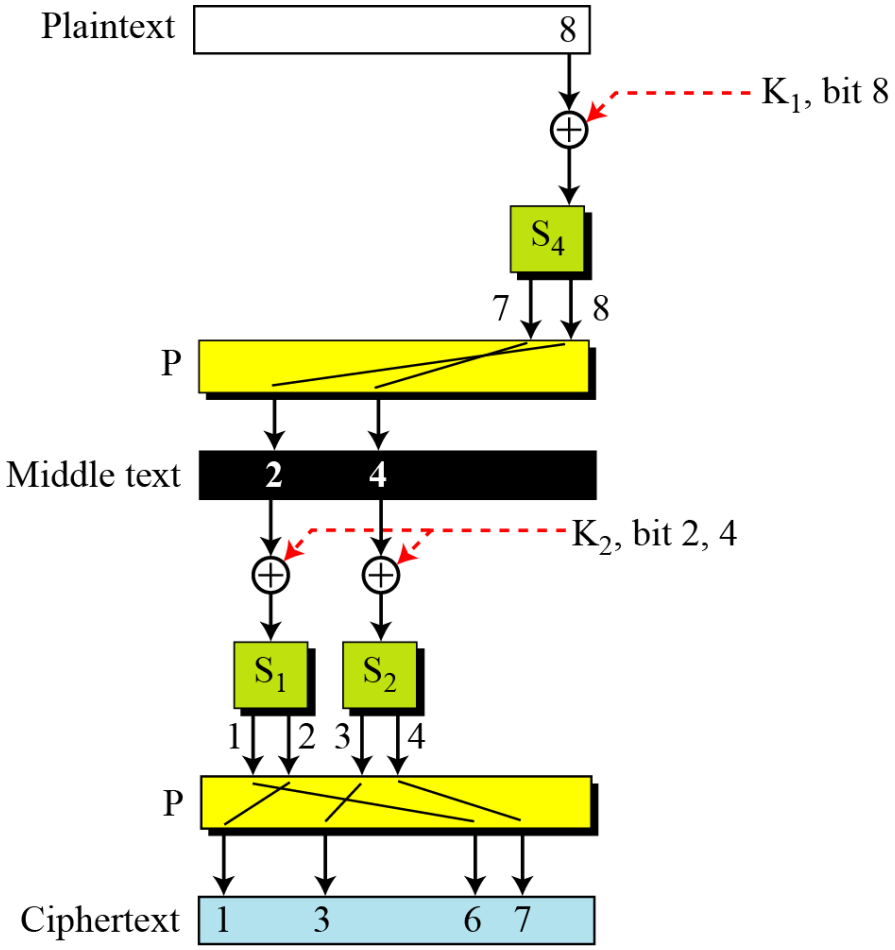
Continued

Figure *Final design of a Feistel cipher with two rounds*



Continued

Figure *Diffusion and confusion in a block cipher*

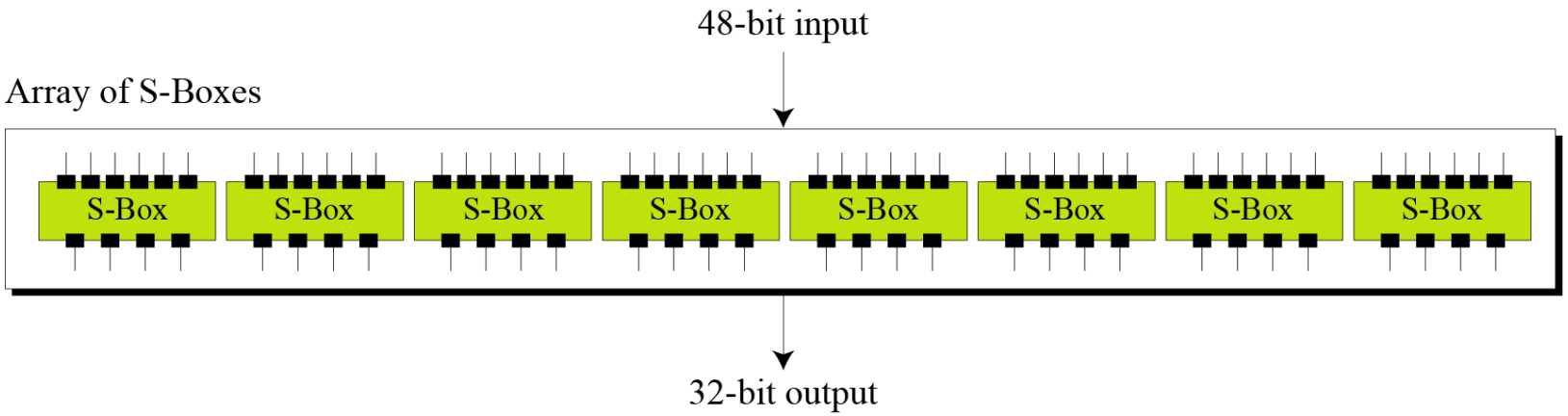


6.2.2 Continue

S-Boxes

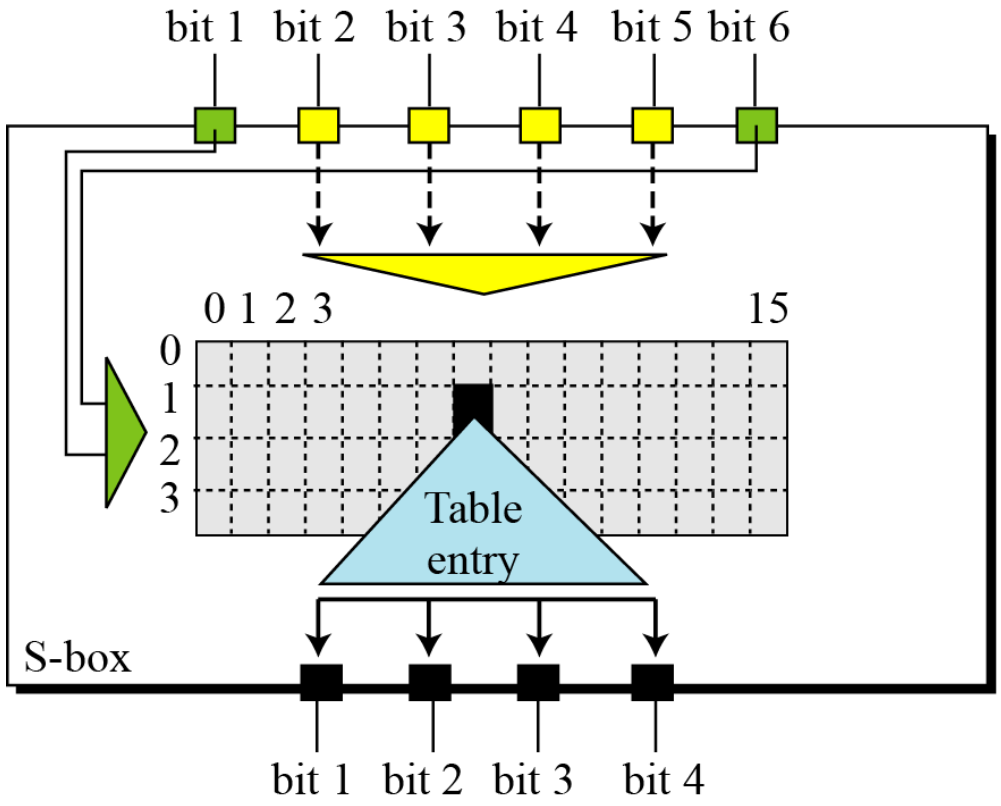
The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. See Figure 6.7.

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

Feistel Cipher Design Features (1 of 2)

- 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 speeds
- Number of rounds
  - The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security
- Subkey generation algorithm
  - Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis

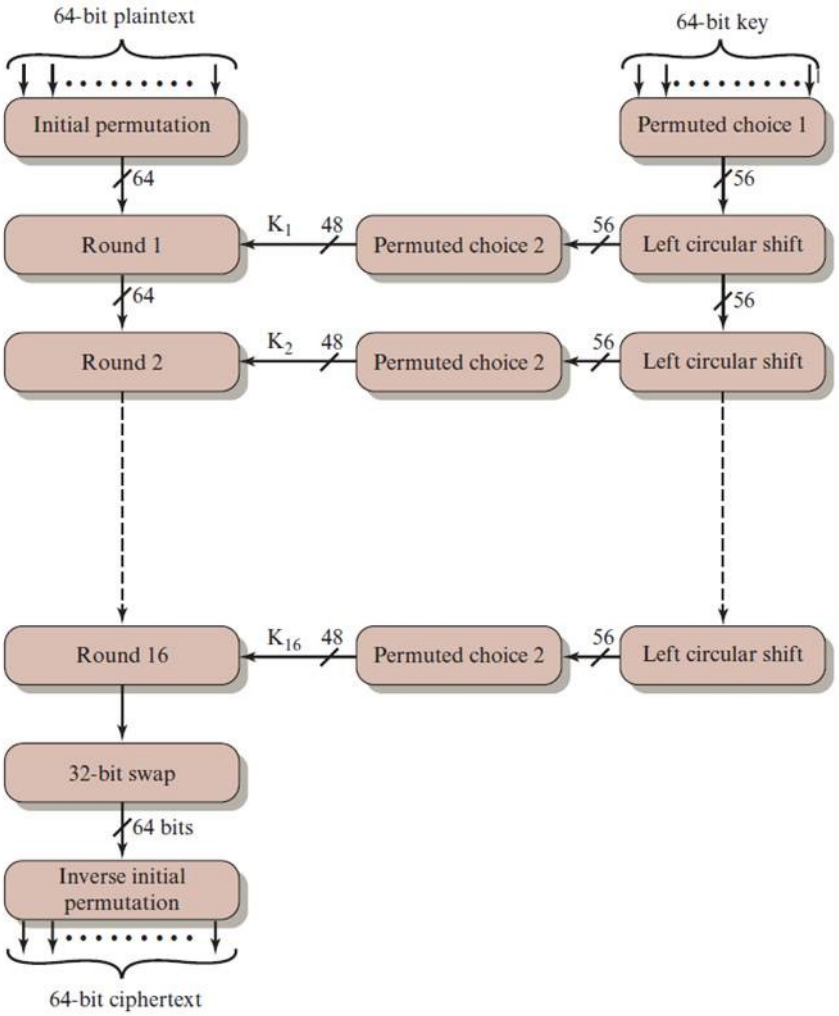
Feistel Cipher Design Features (2 of 2)

- Round function F
  - Greater complexity generally means greater resistance to cryptanalysis
- Fast software encryption/decryption
  - In many cases, encrypting is embedded in applications or utility functions in such a way as to preclude a hardware implementation; accordingly, the speed of execution of the algorithm becomes a concern
- Ease of analysis
  - If the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength

Data Encryption Standard (DES)

- Issued in 1977 by the National Bureau of Standards (now NIST) as Federal Information Processing Standard 46
- Was the most widely used encryption scheme until the introduction of the Advanced Encryption Standard (AES) in 2001
- Algorithm itself is referred to as the Data Encryption Algorithm (DEA)
  - Data are encrypted in 64-bit blocks using a 56-bit 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

Figure 4.5 General Depiction of DES Encryption Algorithm



6.2.3 Continued

Shifting

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

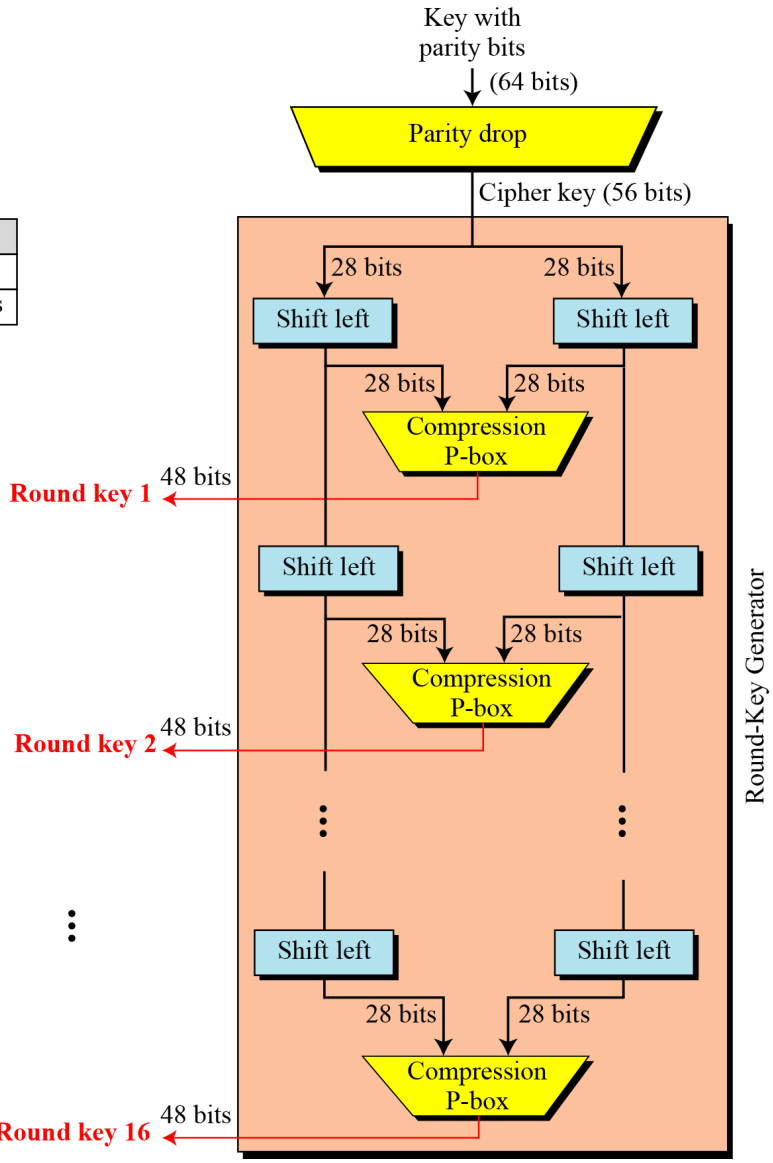


Figure 6.10
Key generation

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

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

6.2.4 Examples

Example 6.5

We choose a random plaintext block and a random key, and determine what the ciphertext block would be (all in hexadecimal):

Plaintext: 123456ABCD132536 Key: AABB09182736CCDD
CipherText: C0B7A8D05F3A829C

Table 6.15 *Trace of data for Example 6.5*

Plaintext: 123456ABCD132536			
After initial permutation: 14A7D67818CA18AD After splitting: L_0=14A7D678 R_0=18CA18AD			
Round	*Left*	*Right*	*Round Key*
Round 1	18CA18AD	5A78E394	194CD072DE8C
Round 2	5A78E394	4A1210F6	4568581ABCCE
Round 3	4A1210F6	B8089591	06EDA4ACF5B5
Round 4	B8089591	236779C2	DA2D032B6EE3

6.2.4 Continued

Example 6.5 Continued

Table 6.15 *Trace of data for Example 6.5 (Conintued*

Round 5	236779C2	A15A4B87	69A629FEC913
Round 6	A15A4B87	2E8F9C65	C1948E87475E
Round 7	2E8F9C65	A9FC20A3	708AD2DDB3C0
Round 8	A9FC20A3	308BEE97	34F822F0C66D
Round 9	308BEE97	10AF9D37	84BB4473DCCC
Round 10	10AF9D37	6CA6CB20	02765708B5BF
Round 11	6CA6CB20	FF3C485F	6D5560AF7CA5
Round 12	FF3C485F	22A5963B	C2C1E96A4BF3
Round 13	22A5963B	387CCDAA	99C31397C91F
Round 14	387CCDAA	BD2DD2AB	251B8BC717D0
Round 15	BD2DD2AB	CF26B472	3330C5D9A36D
Round 16	19BA9212	CF26B472	181C5D75C66D
After combination: 19BA9212CF26B472			
Ciphertext: C0B7A8D05F3A829C			*(after final permutation)*

6.2.4 Continued

Example 6.6

Let us see how Bob, at the destination, can decipher the ciphertext received from Alice using the same key. Table 6.16 shows some interesting points.

Ciphertext: C0B7A8D05F3A829C			
After initial permutation: 19BA9212CF26B472 After splitting: L_0=19BA9212 R_0=CF26B472			
Round	*Left*	*Right*	*Round Key*
Round 1	CF26B472	BD2DD2AB	181C5D75C66D
Round 2	BD2DD2AB	387CCDAA	3330C5D9A36D
. . .	. . .	. . .	. . .
Round 15	5A78E394	18CA18AD	4568581ABCCE
Round 16	14A7D678	18CA18AD	194CD072DE8C
After combination: 14A7D67818CA18AD			
Plaintext:123456ABCD132536 (after final permutation)			

Table 4.2 DES Example

Round	*Ki*	*Li*	*Ri*
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP–1		da02ce3a	89ecac3b

Note: DES subkeys are shown as eight 6-bit values in hex format

Table 4.3 Avalanche Effect in DES: Change in Plaintext

Round		δ
	02468aceeeca86420 12468aceeeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33

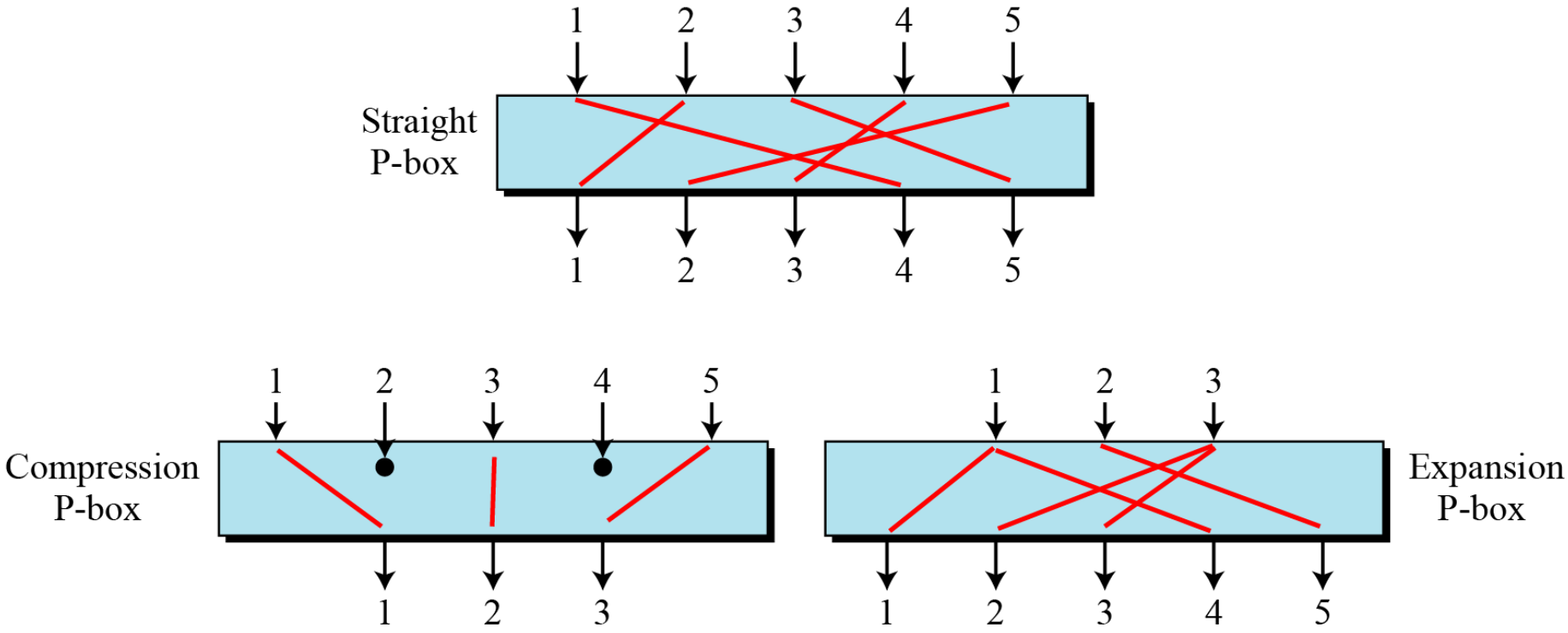
Round		δ
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

Table 4.4 Avalanche Effect in DES: Change in Key

Round		δ
	02468aceeeca86420 02468aceeeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32

Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP−1	da02ce3a89ecac3b ee92b50606b62b0b	30

Figure 5.4 *Three types of P-boxes*



5.1.3 Continued

P-Boxes: Invertibility

Note

A straight P-box is invertible, but compression and expansion P-boxes are not.

5.1.3 Continued

Example 5.7

Figure 5.6 shows how to invert a permutation table represented as a one-dimensional table.

Figure 5.6 *Inverting a permutation table*

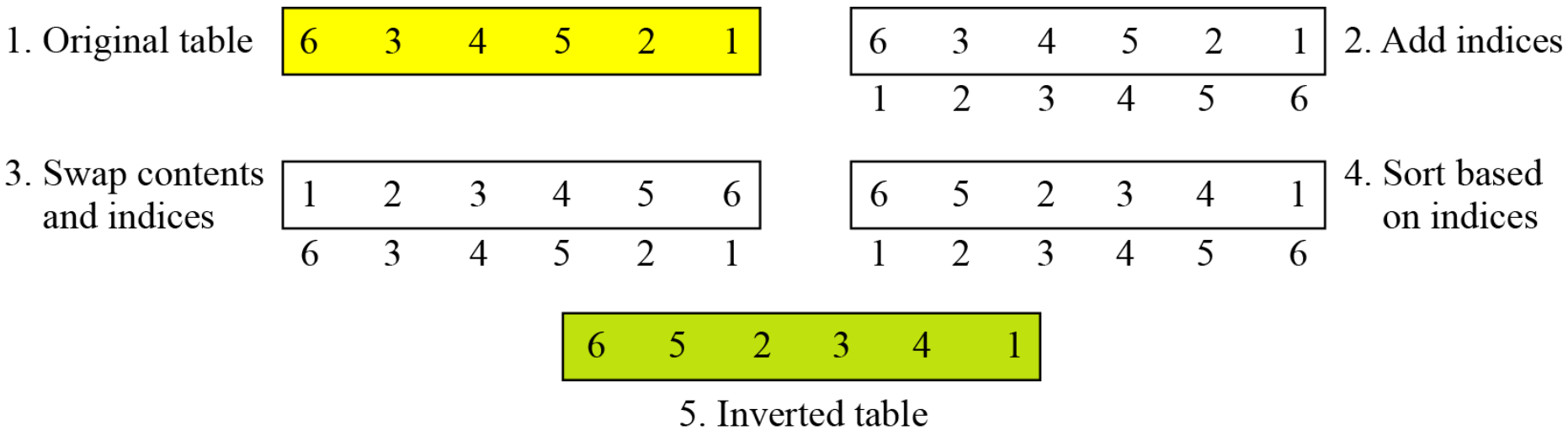
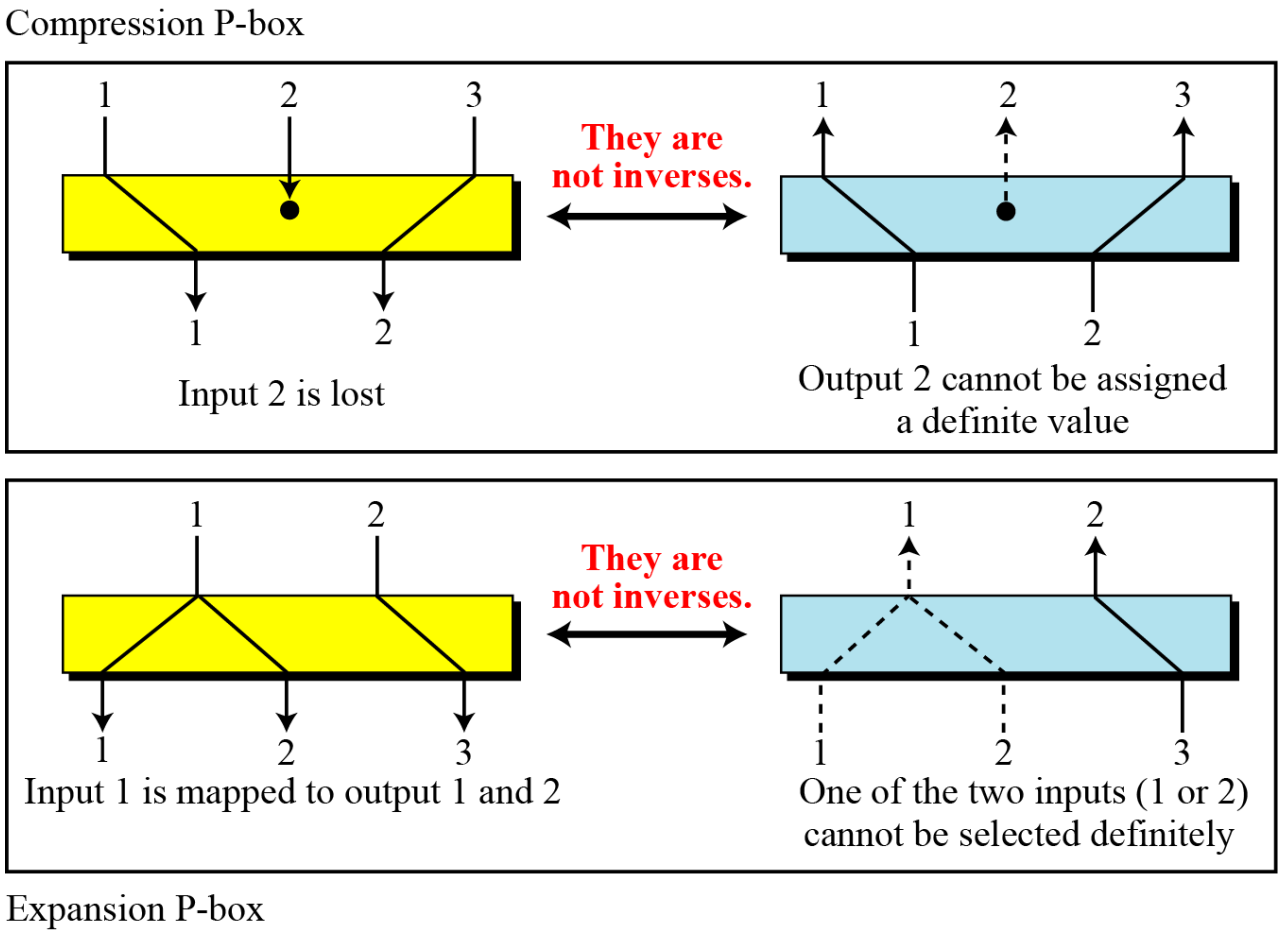


Figure 5.7 *Compression and expansion P-boxes are non-invertible*

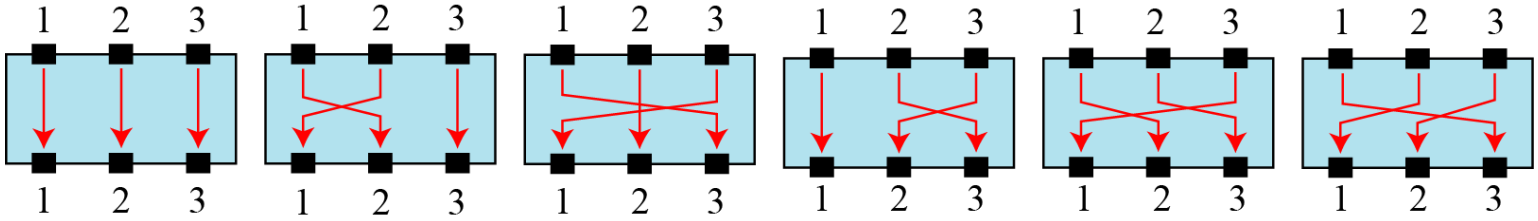


5.1.3 Continued

Example 5.5

Figure 5.5 shows all 6 possible mappings of a 3 × 3 P-box.

Figure 5.5 *The possible mappings of a 3 × 3 P-box*



5.1.3 Continued

Straight P-Boxes

Table 5.1 *Example of a permutation table for a straight P-box*

58	50	42	34	26	18	10	02	60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06	64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01	59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05	63	55	47	39	31	23	15	07

5.1.3 Continued

Compression P-Boxes

A compression P-box is a P-box with n inputs and m outputs where m < n.

Table 5.2 *Example of a 32 × 24 permutation table*

01	02	03	21	22	26	27	28	29	13	14	17
18	19	20	04	05	06	10	11	12	30	31	32

5.1.3 Continued

Expansion P-Boxes

An expansion P-box is a P-box with n inputs and m outputs where m > n.

Table 5.3 *Example of a 12 × 16 permutation table*

01	09	10	11	12	01	02	03	03	04	05	06	07	08	09	12

S-Box

An S-box (substitution box) can be thought of as a miniature substitution cipher.

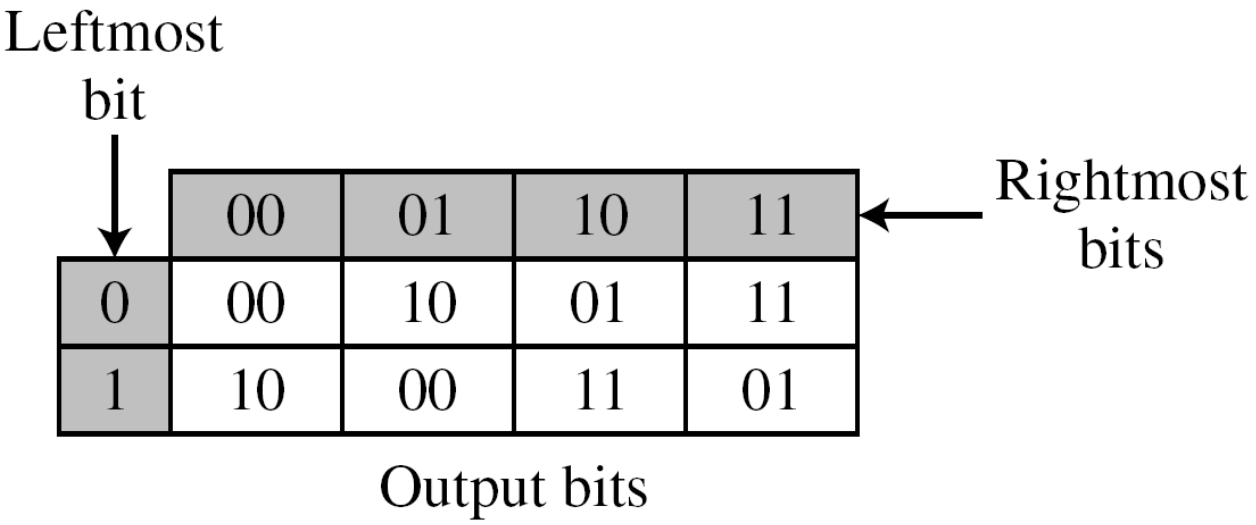
Note

An S-box is an $m \times n$ substitution unit, where m and n are not necessarily the same.

Continued

Example 5.10

The following table defines the input/output relationship for an S-box of size 3 × 2. The leftmost bit of the input defines the row; the two rightmost bits of the input define the column. The two output bits are values on the cross section of the selected row and column.



Based on the table, an input of 010 yields the output 01. An input of 101 yields the output of 00.

Continued

S-Boxes: Invertibility

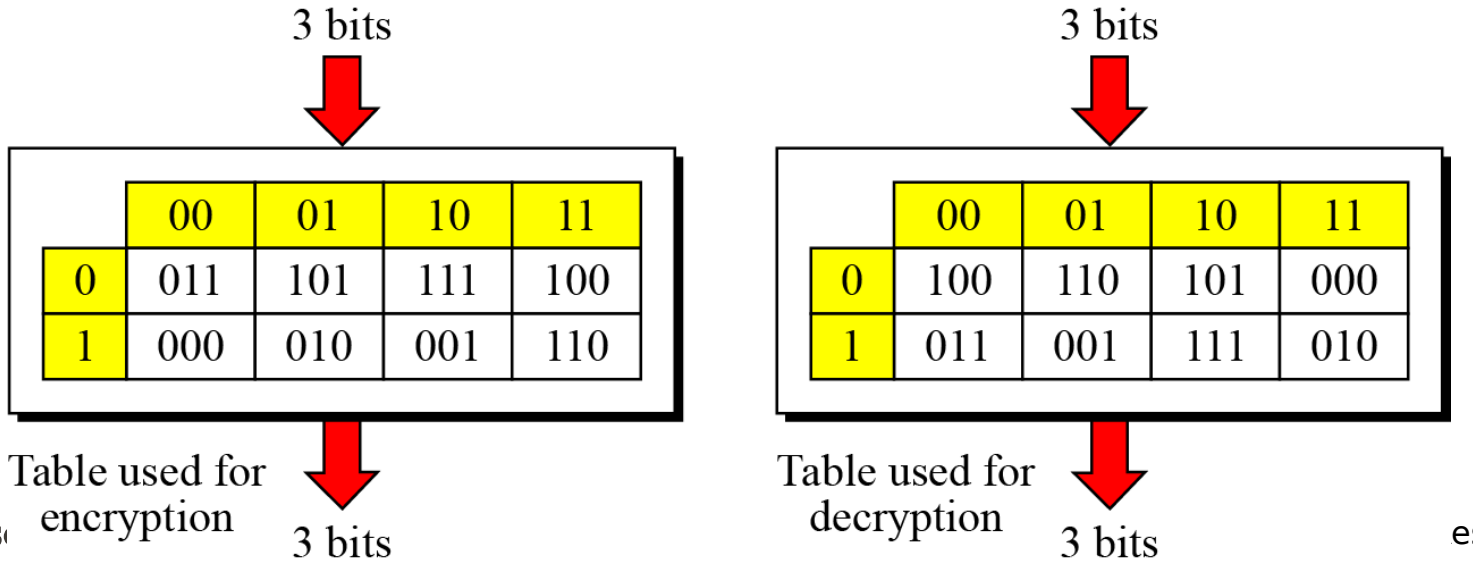
An S-box may or may not be invertible. In an invertible S-box, the number of input bits should be the same as the number of output bits.

Continued

Example 5.11

Figure 5.8 shows an example of an invertible S-box. For example, if the input to the left box is 001, the output is 101. The input 101 in the right table creates the output 001, which shows that the two tables are inverses of each other.

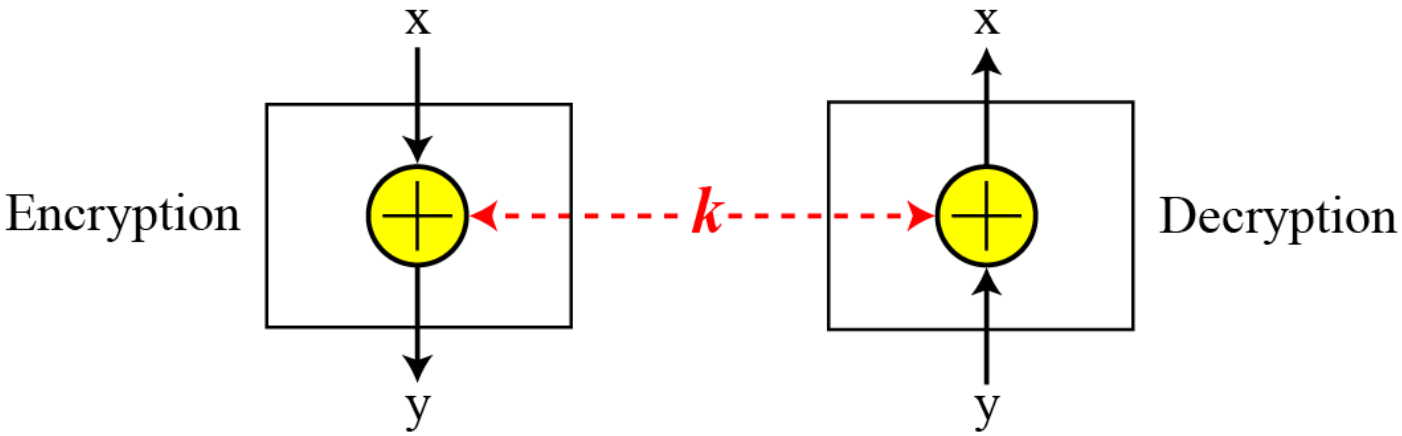
Figure 5.8 *S-box tables for Example 5.11*



Exclusive-Or

An important component in most block ciphers is the exclusive-or operation.

Figure 5.9 *Invertibility of the exclusive-or operation*



5.1.3 Continued

Example 5.12

This is a trivial example. The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

Solution

The function extracts the first and second bits to get 11 in binary or 3 in decimal. The result of squaring is 9, which is 1001 in binary.

$$\textbf{Encryption: } C = P \oplus f(K) = 0111 \oplus 1001 = 1110$$

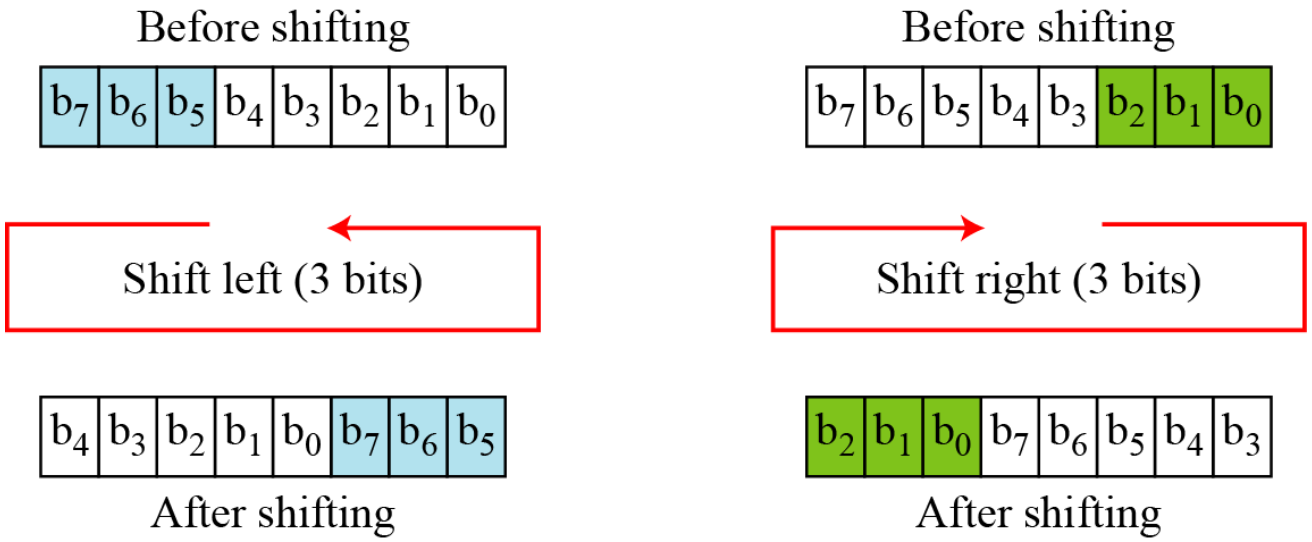
$$\textbf{Decryption: } P = C \oplus f(K) = 1110 \oplus 1001 = 0111$$

5.1.3 Continued

Circular Shift

Another component found in some modern block ciphers is the circular shift operation.

Figure 5.10 *Circular shifting an 8-bit word to the left or right*

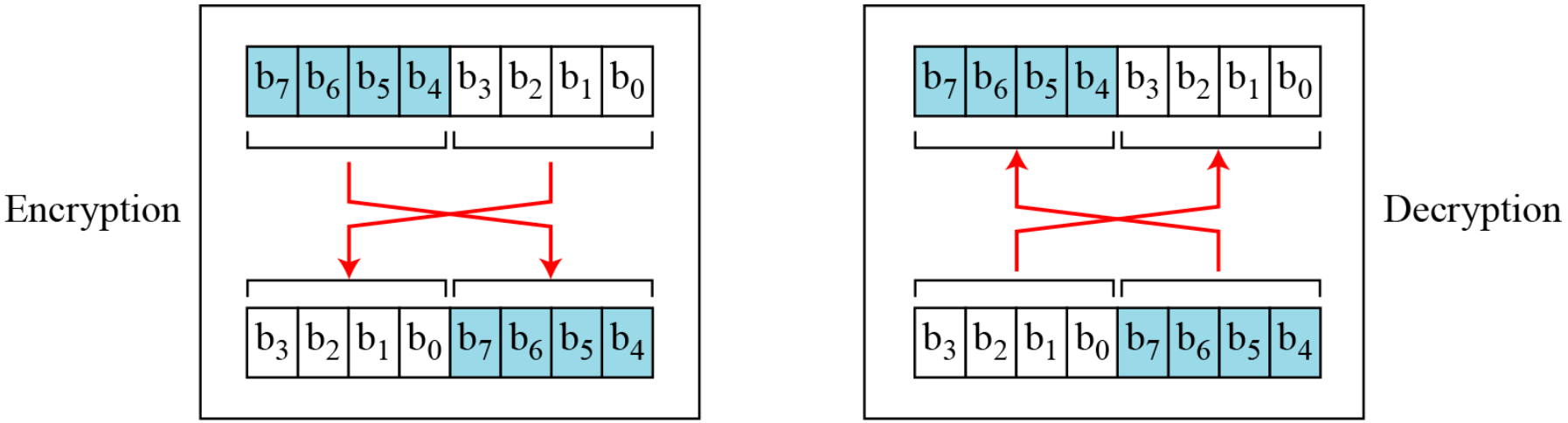


5.1.3 Continued

Swap

The swap operation is a special case of the circular shift operation where k = n/2.

Figure 5.11 *Swap operation on an 8-bit word*



5.1.3 Continued

Split and Combine

Two other operations found in some block ciphers are split and combine.

Figure 5.12 ***Split and combine operations on an 8-bit word***

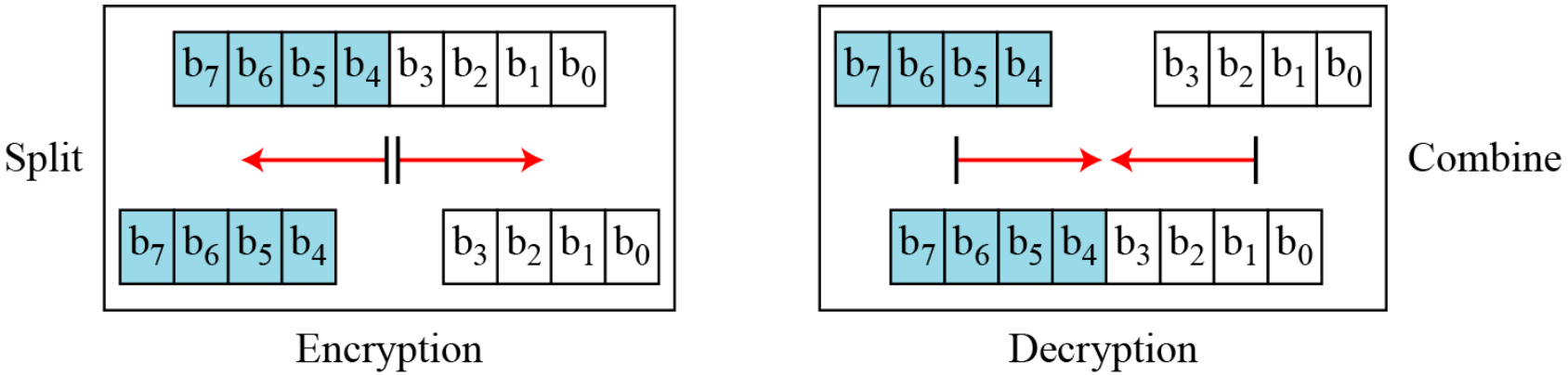
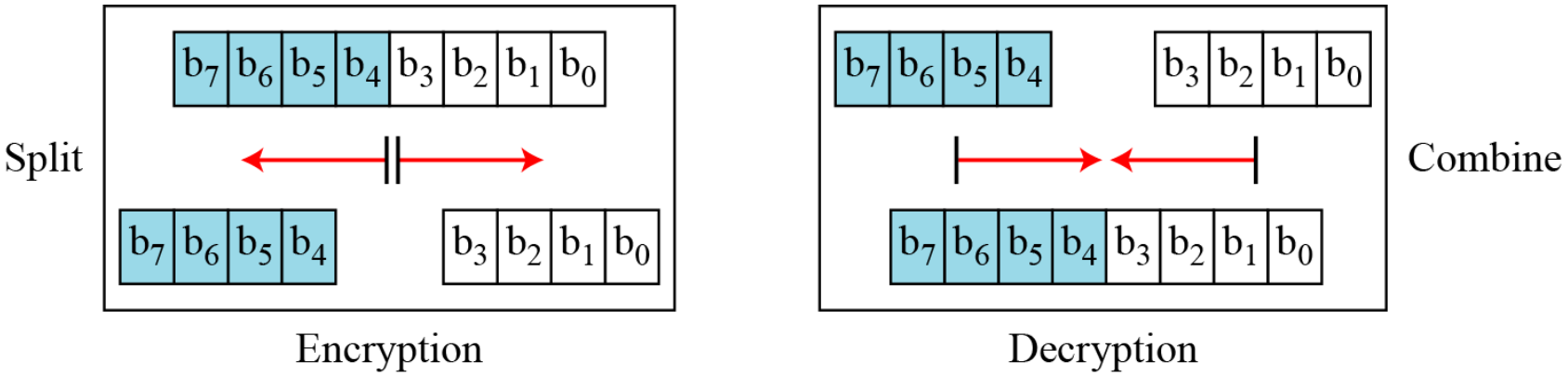


Figure 5.12 *Split and combine operations on an 8-bit word*



5.1.4 Product Ciphers

Shannon introduced the concept of a product cipher. A product cipher is a complex cipher combining substitution, permutation, and other components discussed in previous sections.

5.1.4 Continued

Rounds

Diffusion and confusion can be achieved using iterated product ciphers where each iteration is a combination of S-boxes, P-boxes, and other components.

Table 4.5 Average Time Required for Exhaustive Key Search

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns = 6.3×10^9 years	6.3×10^6 years

5.1.5 Two Classes of Product Ciphers

Modern block ciphers are all product ciphers, but they are divided into two classes.

1. ***Feistel ciphers***

2. ***Non-Feistel ciphers***

5.1.5 Continued

Feistel Ciphers

Feistel designed a very intelligent and interesting cipher that has been used for decades. A Feistel cipher can have three types of components: self-invertible, invertible, and noninvertible.

Self-invertible Feistel cipher. $f(f(x))=x$
Example: $1010 \oplus 1100 = 0110$ and $0110 \oplus 1100 = 1010$.
Invertible Feistel cipher.
Example: product cipher.

5.1.5 Continued

Non-Feistel Ciphers

- ***A non-Feistel cipher uses only invertible components.***
- ***A component in the encryption cipher has the corresponding component in the decryption cipher.***

5.1.6 Attacks on Block Ciphers

Attacks on traditional ciphers can also be used on modern block ciphers, but today's block ciphers resist most of the attacks discussed in Chapter 3.

5.1.5 Continued

Differential Cryptanalysis

Eli Biham and Adi Shamir introduced the idea of differential cryptanalysis. This is a chosen-plaintext attack.

5.1.6 Continued

Example 5.13

Assume that the cipher is made only of one exclusive-or operation, as shown in Figure 5.18. Without knowing the value of the key, Eve can easily find the relationship between plaintext differences and ciphertext differences if by plaintext difference we mean P1 ⊕ P2 and by ciphertext difference, we mean C1⊕ C2. The following proves that C1 ⊕ C2 = P1 ⊕ P2:

$$C_1 = P_1 \oplus K \quad C_2 = P_2 \oplus K \quad \rightarrow \quad C_1 \oplus C_2 = P_1 \oplus K \oplus P_2 \oplus K = P_1 \oplus P_2$$

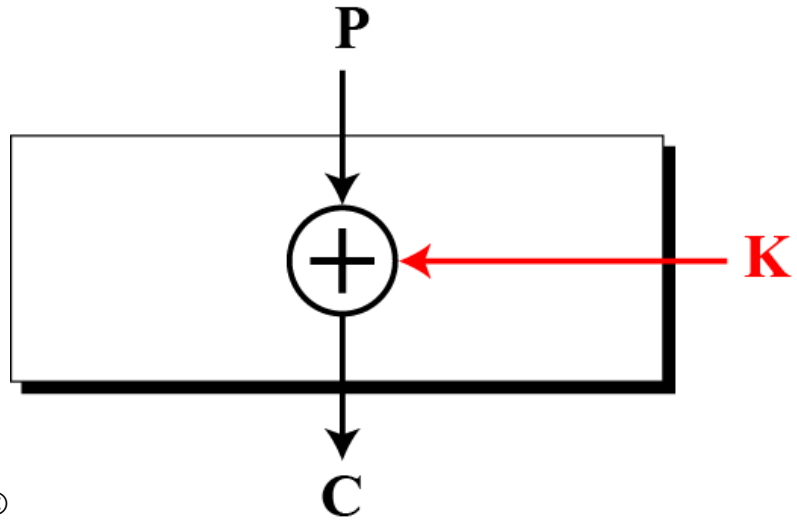


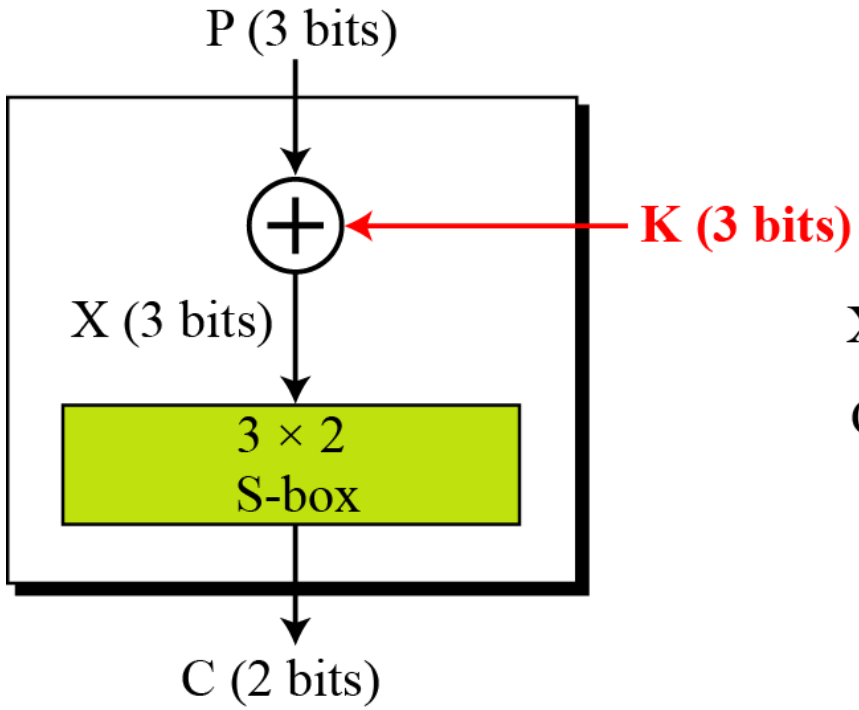
Figure 5.18 *Diagram for Example 5.13*

5.1.6 Continued

Example 5.14

We add one S-box to Example 5.13, as shown in Figure 5.19.

Figure 5.19 *Diagram for Example 5.14*



X	000	001	010	011	100	101	110	111
C	11	00	10	10	01	00	11	00

S-box table

5.1.6 Continued

Example 5.14 Continued

Eve now can create a probabilistic relationship as shown in Table 5.4.

Table 5.4 *Differential input/output*

$P_1 \oplus P_2$ \ $C_1 \oplus C_2$	*00*	*01*	*10*	*11*
000	**8**			
001	**2**	**2**		**4**
010	**2**	**2**	**4**	
011		**4**	**2**	**2**
100	**2**	**2**	**4**	
101		**4**	**2**	**2**
110	**4**		**2**	**2**
111			**2**	**6**

5.1.6 Continued

Example 5.15

The heuristic result of Example 5.14 can create probabilistic information for Eve as shown in Table 5.5.

Table 5.5 *Differential distribution table*

$P_1 \oplus P_2$ \ $C_1 \oplus C_2$	*00*	*01*	*10*	*11*
000	**1**	**0**	**0**	**0**
001	**0.25**	**0.25**	**0**	**0.50**
010	**0.25**	**0.25**	**0.50**	**0**
011	**0**	**0.50**	**0.25**	**0.25**
100	**0.25**	**0.25**	**0.50**	**0**
101	**0**	**0.50**	**0.25**	**0.25**
110	**0.50**	**0**	**0.25**	**0.25**
111	**0**	**0**	**0.25**	**0.75**

5.1.6 Continued

Example 5.16

Looking at Table 5.5, Eve knows that if $P_1 \oplus P_2 = 001$, then $C_1 \oplus C_2 = 11$ with the probability of 0.50 (50 percent). She tries $C_1 = 00$ and gets $P_1 = 010$ (chosen-ciphertext attack). She also tries $C_2 = 11$ and gets $P_2 = 011$ (another chosen-ciphertext attack). Now she tries to work backward, based on the first pair, P_1 and C_1,

$$C_1 = 00 \rightarrow X_1 = 001 \quad \text{or} \quad X_1 = 111$$
$$\text{If } X_1 = 001 \rightarrow K = X_1 \oplus P_1 = 011 \qquad \text{If } X_1 = 111 \rightarrow K = X_1 \oplus P_1 = 101$$

$$C_2 = 11 \rightarrow X_2 = 000 \quad \text{or} \quad X_1 = 110$$
$$\text{If } X_2 = 000 \rightarrow K = X_2 \oplus P_2 = 011 \qquad \text{If } X_2 = 110 \rightarrow K = X_2 \oplus P_2 = 101$$

The two tests confirm that K = 011 or K =101.

5.1.6 Continued

Note

Differential cryptanalysis is based on a nonuniform differential distribution table of the S-boxes in a block cipher.

Linear Cryptanalysis

Linear cryptanalysis was presented by Mitsuru Matsui in 1993. The analysis uses known plaintext attacks.

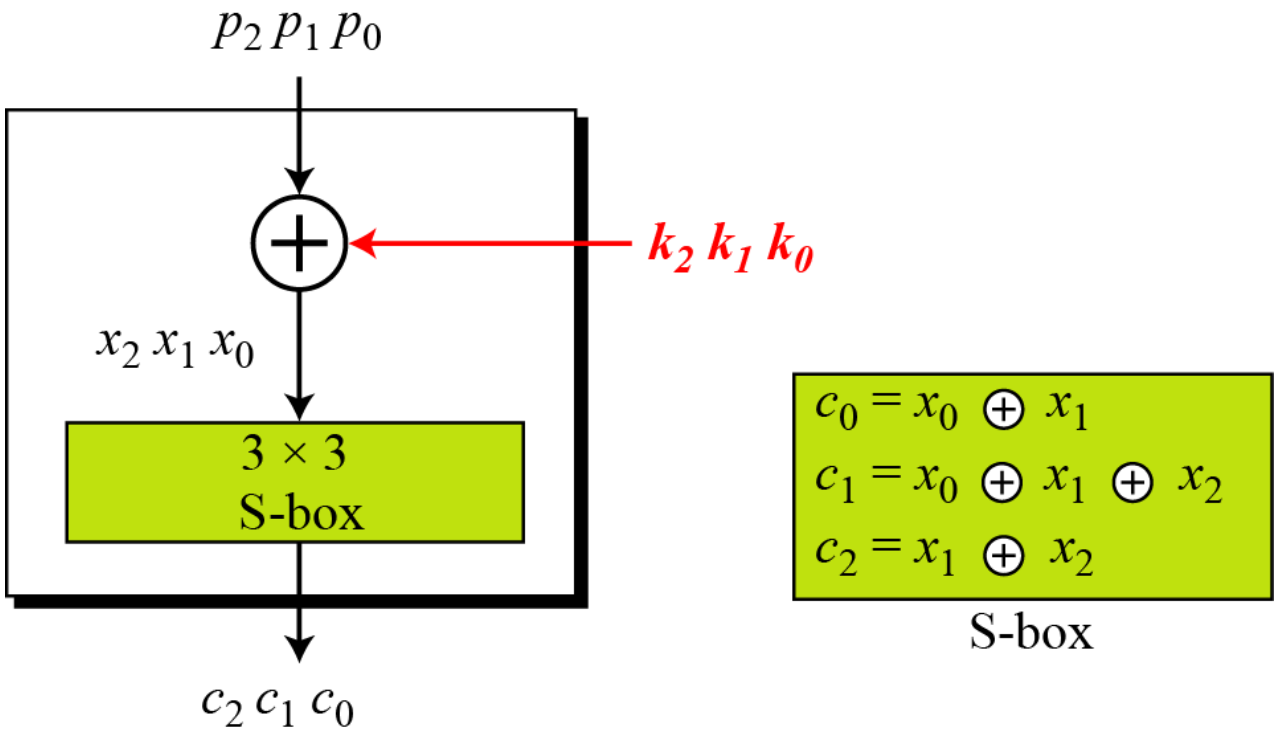


Figure 5.20 *A simple cipher with a linear S-box*

$$c_0 = p_0 \oplus k_0 \oplus p_1 \oplus k_1$$
$$c_1 = p_0 \oplus k_0 \oplus p_1 \oplus k_1 \oplus p_2 \oplus k_2$$
$$c_2 = p_1 \oplus k_1 \oplus p_2 \oplus k_2$$

Solving for three unknowns, we get.

$$k_1 = (p_1) \oplus (c_0 \oplus c_1 \oplus c_2)$$
$$k_2 = (p_2) \oplus (c_0 \oplus c_1)$$
$$k_0 = (p_0) \oplus (c_1 \oplus c_2)$$

This means that three known-plaintext attacks can find the values of k_0, k_1, and k_2.

5.1.6 Continued

In some modern block ciphers, it may happen that some S-boxes are not totally nonlinear; they can be approximated, probabilistically, by some linear functions.

$$(k_0 \oplus k_1 \oplus \cdots \oplus k_x) = (p_0 \oplus p_1 \oplus \cdots \oplus p_y) \oplus (c_0 \oplus c_1 \oplus \cdots \oplus c_z)$$

where $1 \le x \le m$, $1 \le y \le n$, and $1 \le z \le n$.

Note

A more detailed linear cryptanalysis is given in Appendix N.

Strength of DES

- Timing attacks
  - One in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts
  - Exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs
  - So far it appears unlikely that this technique will ever be successful against DES or more powerful symmetric ciphers such as triple DES and AES

Block Cipher Design Principles: Number of Rounds

- The greater the number of rounds, the more difficult it is to perform cryptanalysis
- In general, the criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack
- If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search

Block Cipher Design Principles: Design of Function F

- The heart of a Feistel block cipher is the function F
- The more nonlinear F, the more difficult any type of cryptanalysis will be
- The SAC and BIC criteria appear to strengthen the effectiveness of the confusion function

The algorithm should have good avalanche properties

- Strict avalanche criterion (SAC)
  - States that any output bit j of an S-box should change with probability 1/2 when any single input bit i is inverted for all i , j
- Bit independence criterion (BIC)
  - States that output bits j and k should change independently when any single input bit i is inverted for all i , j , and k

Block Cipher Design Principles: Key Schedule Algorithm

- With any Feistel block cipher, the key is used to generate one subkey for each round
- In general, we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key
- It is suggested that, at a minimum, the key schedule should guarantee key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion

6-3 DES ANALYSIS

Critics have used a strong magnifier to analyze DES. Tests have been done to measure the strength of some desired properties in a block cipher.

Topics discussed in this section:

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: 0000000000000001 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.

6.3.3 DES Weaknesses

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

Weaknesses in Cipher Design

1. ***Weaknesses in S-boxes***
2. ***Weaknesses in P-boxes***
3. ***Weaknesses in Key***

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

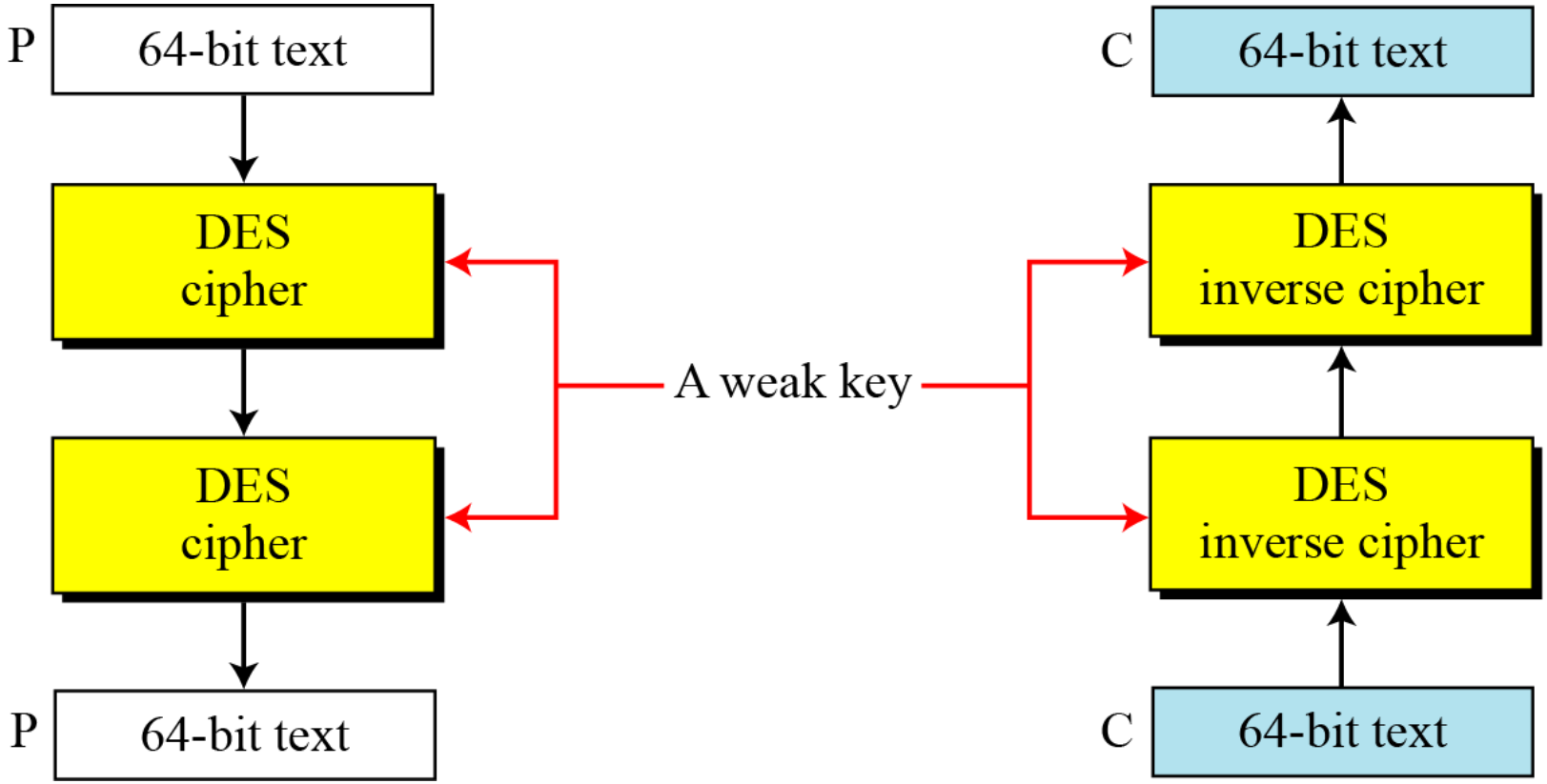
Let us try the first weak key in Table 6.18 to encrypt a block two times. After two encryptions with the same key the original plaintext block is created. Note that we have used the encryption algorithm two times, not one encryption followed by another decryption.

Key: 0x0101010101010101
Plaintext: *0x123456788765432l* Ciphertext: 0x814FE938589154F7

Key: 0x0101010101010101
Plaintext: 0x814FE938589154F7 Ciphertext: *0x123456788765432l*

6.3.3 Continued

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



6.3.3 Continued

Table 6.19 *Semi-weak keys*

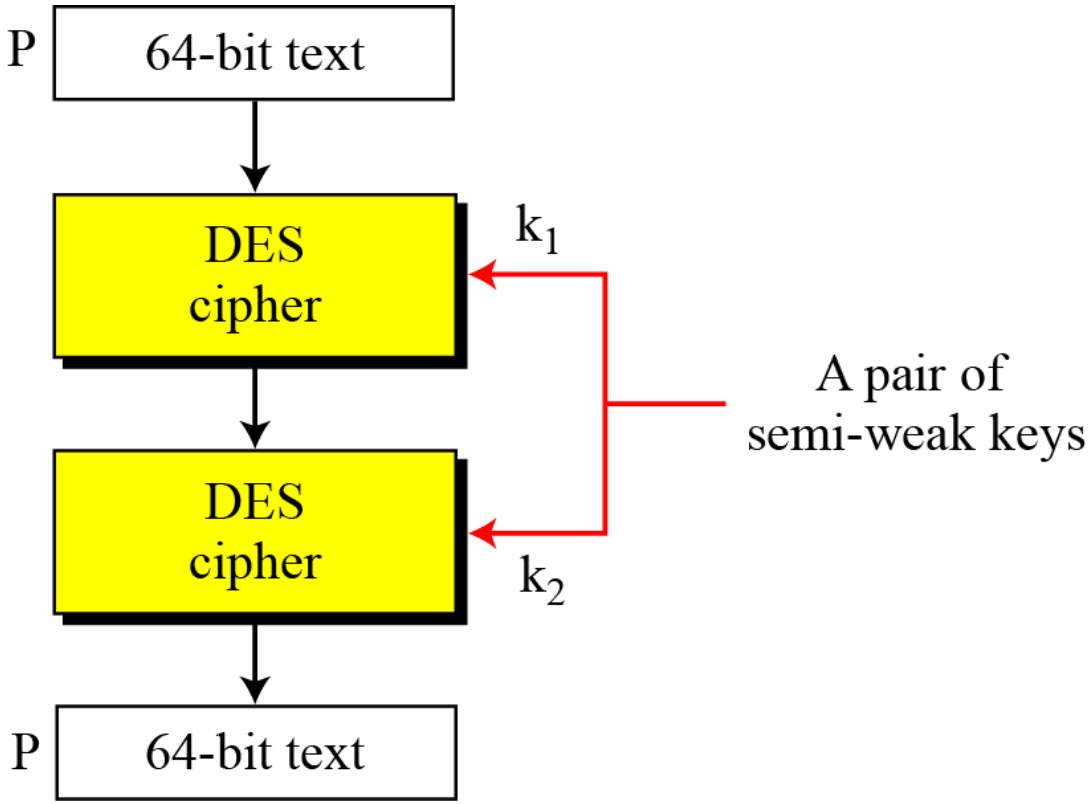
First key in the pair	*Second key in the pair*
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
01E0 01E1 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

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

6.3.3 Continued

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



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 use only half of the possible keys (2^{55}) to perform brute-force attack. This is because

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

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

6-4 Multiple DES

The major criticism of DES regards its key length. Fortunately DES is not a group. This means that we can use double or triple DES to increase the key size.

Topics discussed in this section:

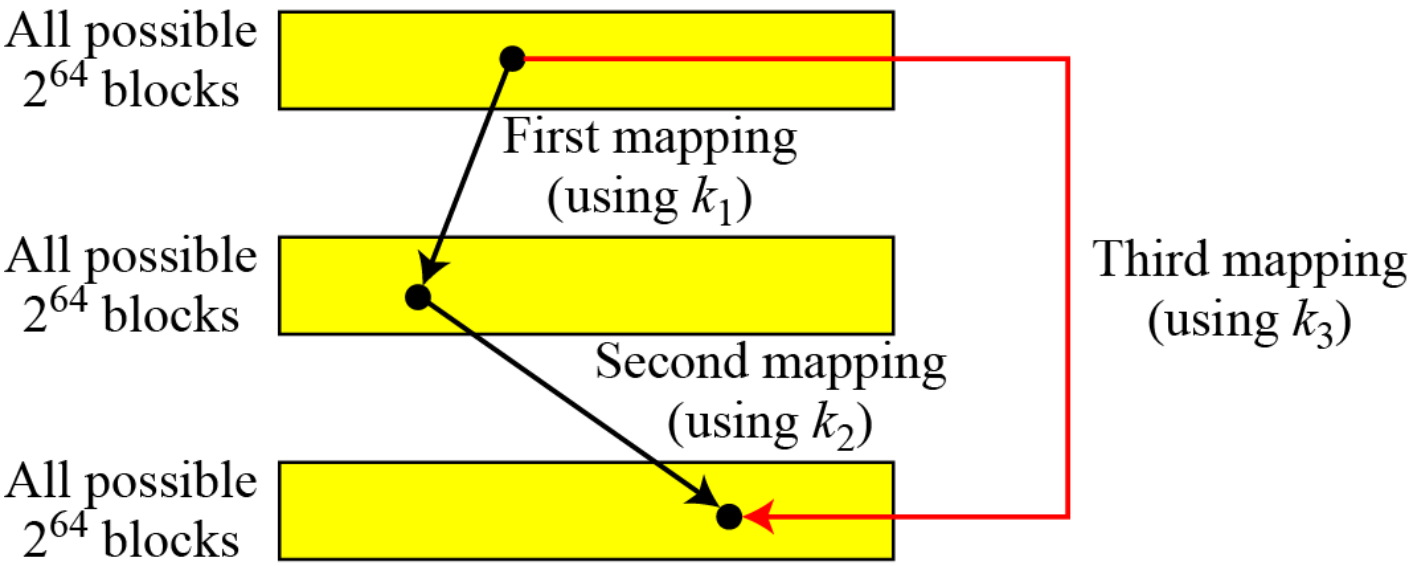
6.4.1 Double DES

6.4.4 Triple DES

6-4 Continued

A substitution that maps every possible input to every possible output is a group.

Figure 6.13 ***Composition of mapping***



6.4.1 Double DES

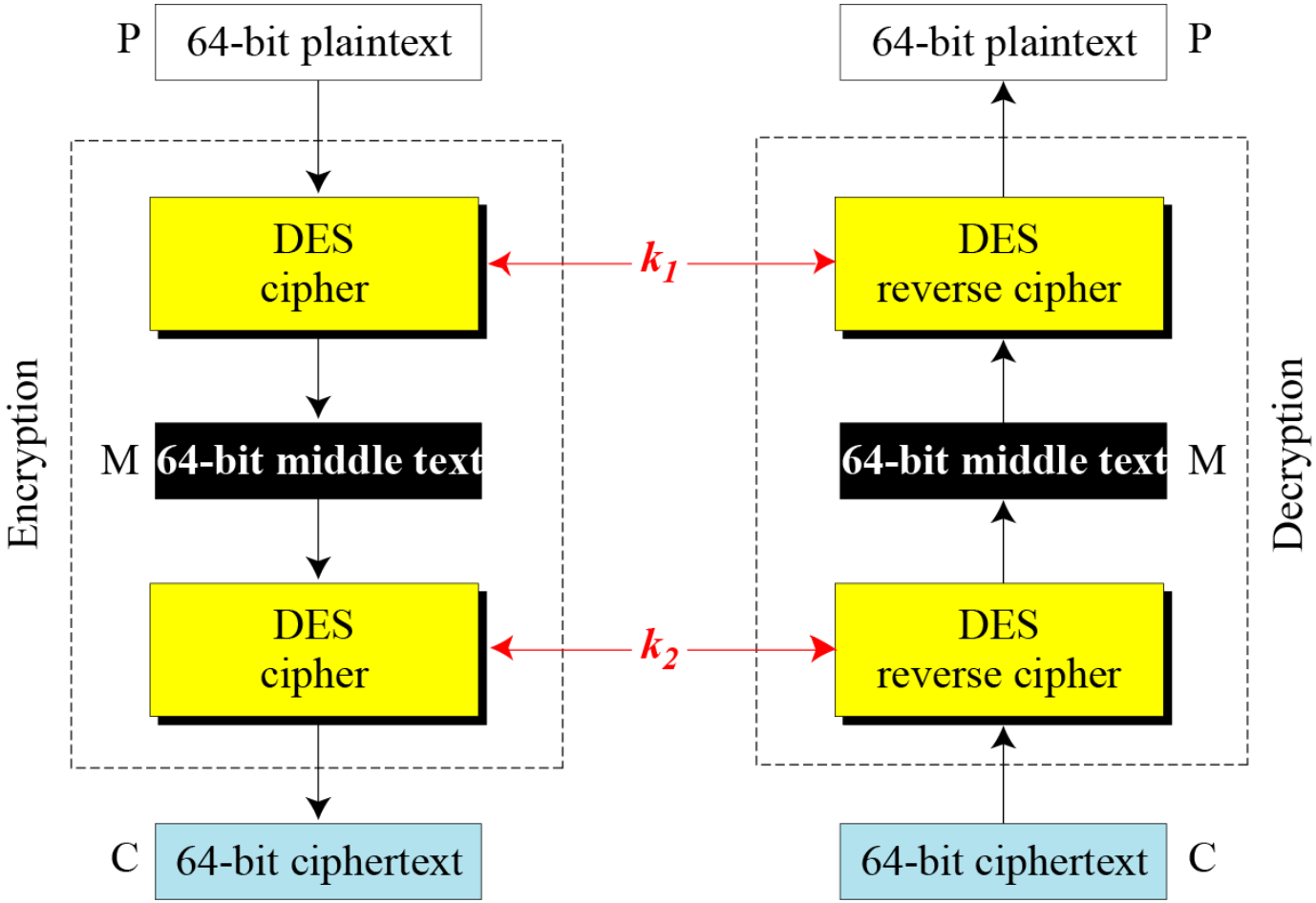
The first approach is to use double DES (2DES).

Meet-in-the-Middle Attack

However, using a known-plaintext attack called meet-in-the-middle attack proves that double DES improves this vulnerability slightly (to 2^{57} tests), but not tremendously (to 2^{112}).

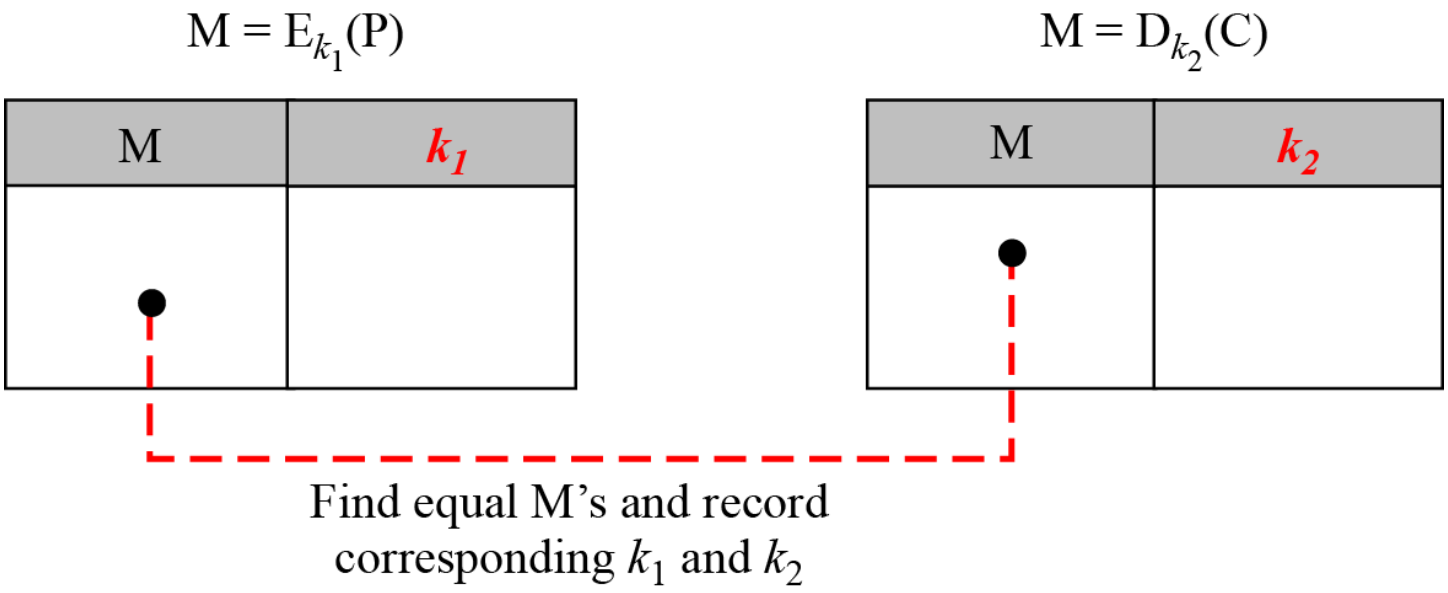
6.4.1 Continued

Figure 6.14 *Meet-in-the-middle attack for double DES*



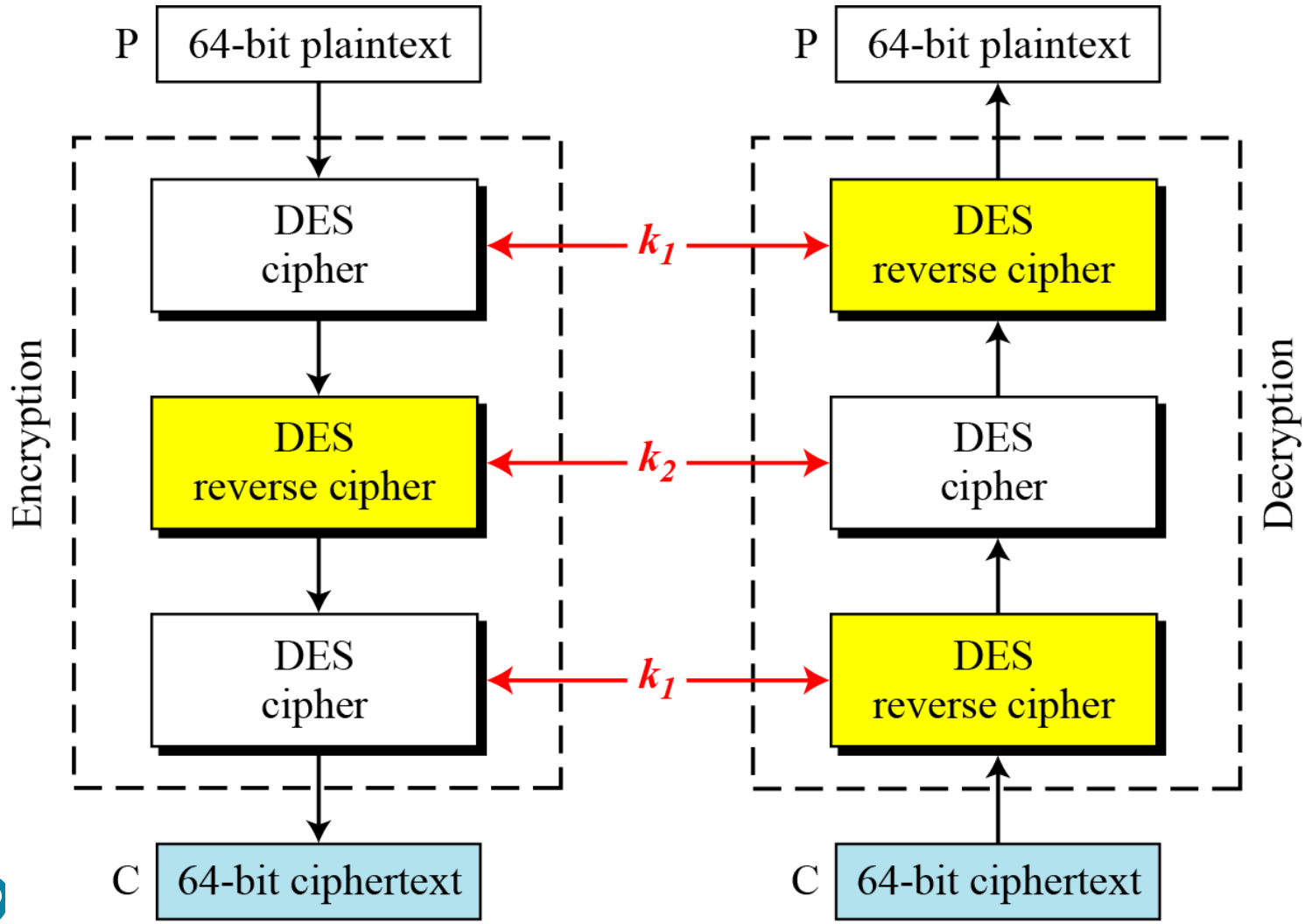
6.4.1 Continued

Figure 6.15 *Tables for meet-in-the-middle attack*



6.4.2 Triple DES

Figure 6.16 *Triple DES with two keys*



6.4.2 Continuous

Triple DES with Three Keys

The possibility of known-plaintext attacks on triple DES with two keys has enticed some applications to use triple DES with three keys. Triple DES with three keys is used by many applications such as PGP (See Chapter 16).

6-5 Security of DES

DES, as the first important block cipher, has gone through much scrutiny. Among the attempted attacks, three are of interest: brute-force, differential cryptanalysis, and linear cryptanalysis.

Topics discussed in this section:

6.5.1 Brute-Force Attack

We have discussed the weakness of short cipher key in DES. Combining this weakness with the key complement weakness, it is clear that DES can be broken using 2^{55} encryptions.

6.5.2 Differential Cryptanalysis

It has been revealed that the designers of DES already knew about this type of attack and designed S-boxes and chose 16 as the number of rounds to make DES specifically resistant to this type of attack.

Note

We show an example of DES differential cryptanalysis in Appendix N.

6.5.3 Linear Cryptanalysis

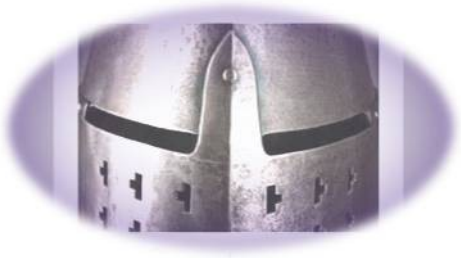
Linear cryptanalysis is newer than differential cryptanalysis. DES is more vulnerable to linear cryptanalysis than to differential cryptanalysis. S-boxes are not very resistant to linear cryptanalysis. It has been shown that DES can be broken using 2^{43} pairs of known plaintexts. However, from the practical point of view, finding so many pairs is very unlikely.

Note

We show an example of DES linear cryptanalysis in Appendix N.

Summary

- Explain the concept of the avalanche effect
- Discuss the cryptographic strength of DES
- Summarize the principal block cipher design principles
- Understand the distinction between stream ciphers and block ciphers
- Present an overview of the Feistel cipher and explain how decryption is the inverse of encryption
- Present an overview of Data Encryption Standard (DES)



Copyright