Chapter 5 – Part A

Introduction to Modern Symmetric-key Ciphers

Assignment 4

Chapter 5 Exercises

- 13, 14
- 20, 21, 22
- 27
- 28, 30
- 32, 33

5-1 MODERN BLOCK CIPHERS

A symmetric-key modern block cipher encrypts an n-bit block of plaintext or decrypts an n-bit block of ciphertext. The encryption or decryption algorithm uses a k-bit key.

Topics discussed in this section:

5.1.1 Substitution or Transposition

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

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

5.1.1 Continued

Example 5.2

Suppose that we have a block cipher where $n = 64$. If there are 10 1's in the ciphertext, how many trial-and-error tests does Eve need to do to recover the plaintext from the intercepted ciphertext in each of the following cases?

a. The cipher is designed as a substitution cipher.

b. The cipher is designed as a transposition cipher.

Solution

a. In the first case, Eve has no idea how many 1's are in the plaintext. Eve needs to try all possible 2^{64} 64-bit blocks to find one that makes sense.

b. In the second case, Eve knows that there are exactly 10 1's in the plaintext. Eve can launch an exhaustive-search attack using only those 64-bit blocks that have exactly 10 1's.

$$\binom{64}{10} = \frac{64!}{(10!)(54!)} = 151{,}473{,}214{,}816$$

(Less than 3 min...)

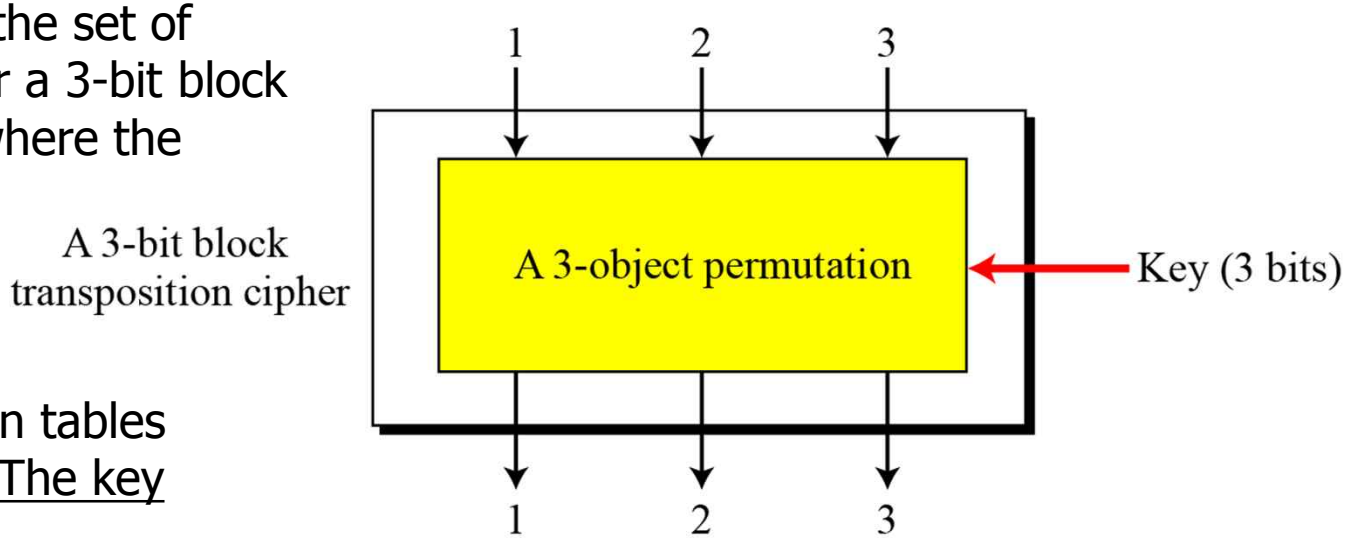
5.1.2 Block Ciphers as Permutation Groups

Full-Size Key Transposition Block Ciphers

In a full-size key transposition cipher, we need to have $n!$ possible keys, so the key should have $\lceil \log_2 n! \rceil$ bits.

Example 5.3

Show the model and the set of permutation tables for a 3-bit block transposition cipher where the block size is 3 bits.



A 3-bit block transposition cipher

Solution

The set of permutation tables has 3! = 6 elements. The key should be 3 bits long.

{[1 2 3], [1 3 2], [2 1 3], [2 3 1], [3 1 2], [3 2 1]}

The set of permutation tables with 3! = 6 elements

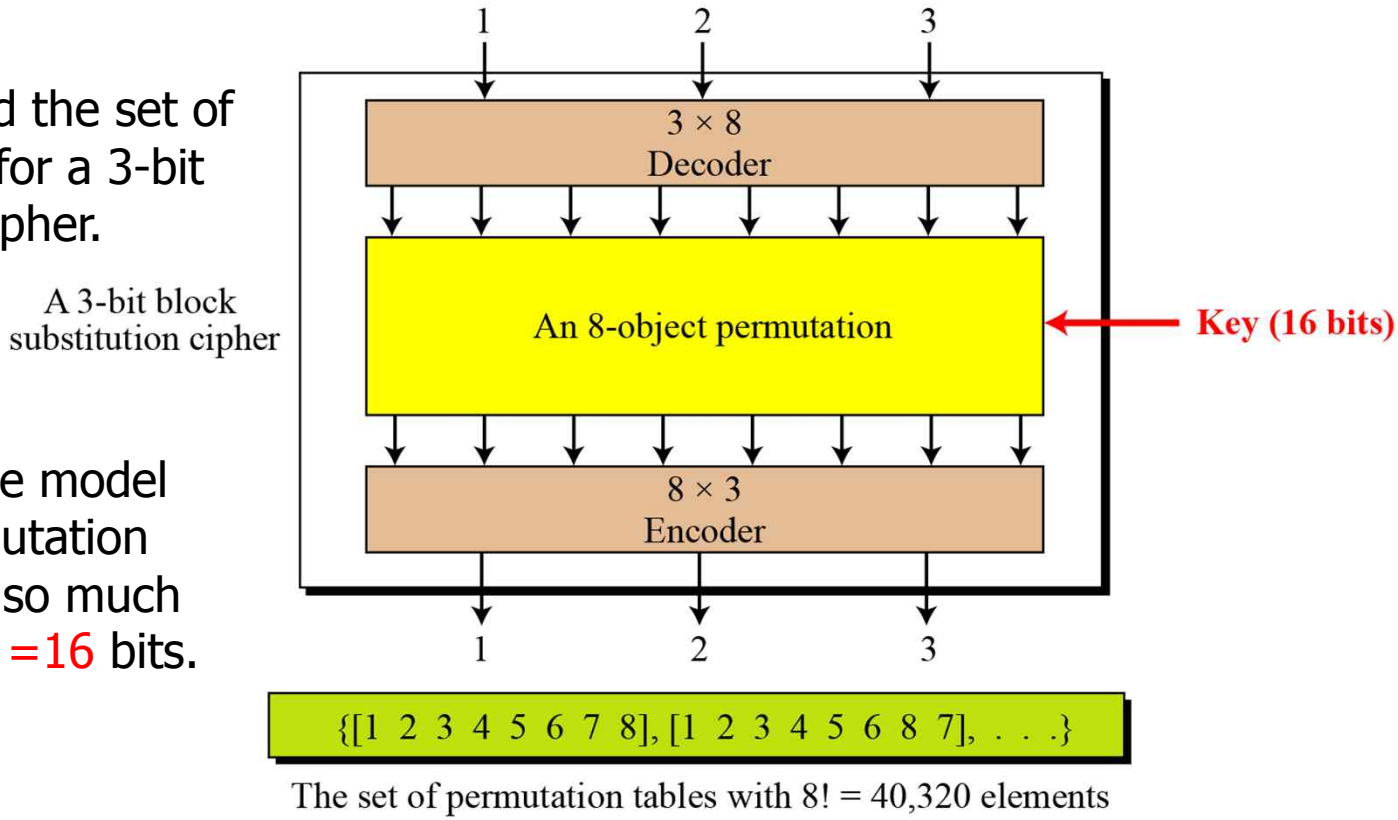
5.1.2 Continued

Full-Size Key Substitution Block Ciphers

A full-size key substitution cipher does not transpose bits; it substitutes bits. We can model the substitution cipher as a permutation if we can decode the input and encode the output.

Example 5.4

Show the model and the set of permutation tables for a 3-bit block substitution cipher.



A 3-bit block substitution cipher

The set of permutation tables with 8! = 40,320 elements

Solution

Figure 5.3 shows the model and the set of permutation tables. The key is also much longer, $\lceil \log_2 40{,}320 \rceil = 16$ bits.

5.1.2 Continued

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 $\lceil \log_2 (2^n)! \rceil$ bits long.

Actual ciphers use a partial-key cipher because the size of the full-size key becomes so large, especially for a substitution cipher.

A common substitution cipher called DES uses a 64-bit block cipher with a key of 56 bits.

5.1.3 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.

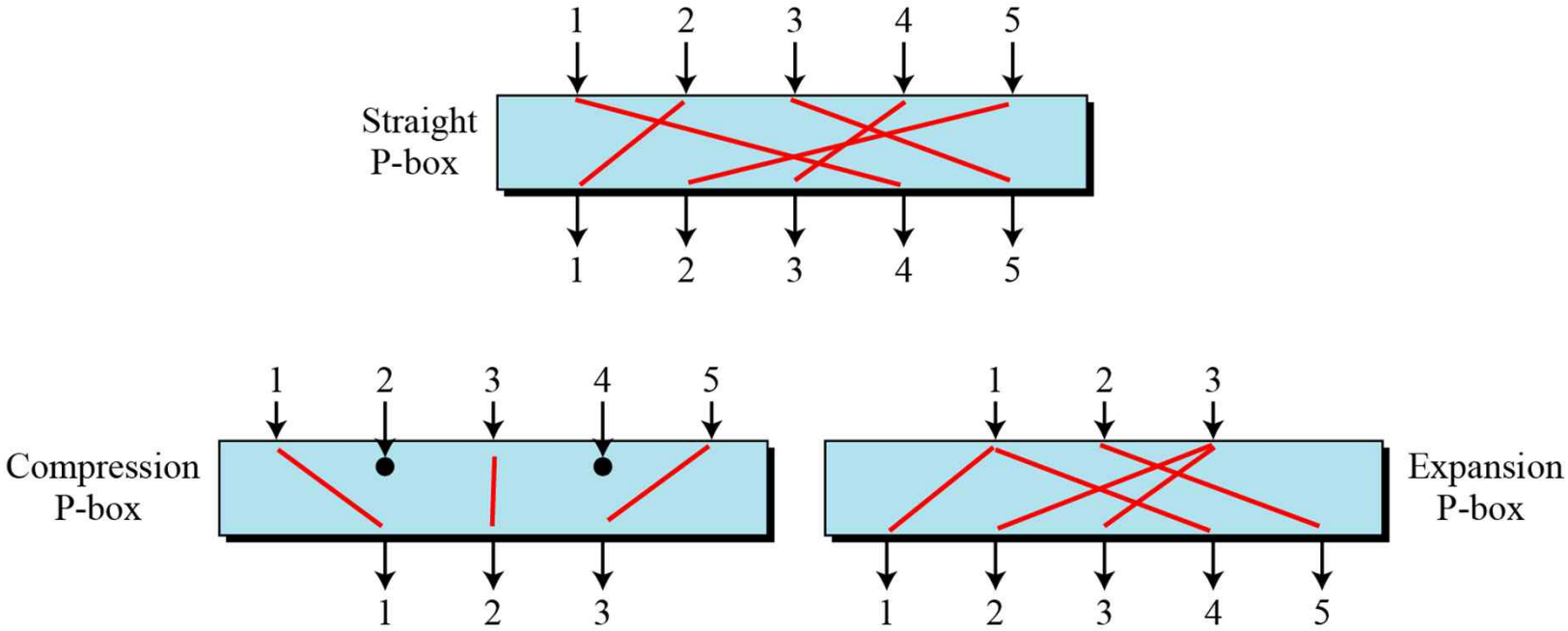
Although a keyless cipher is practically useless by itself, keyless ciphers are used as components of keyed ciphers.

Keyless transposition ciphers : **P-boxes**

Keyless substitution ciphers : **S-boxes**

5.1.3 Three Types of P-Boxes

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



5.1.3 Continued

A straight P-box : all 6 possible mappings of a 3 × 3 P-box.

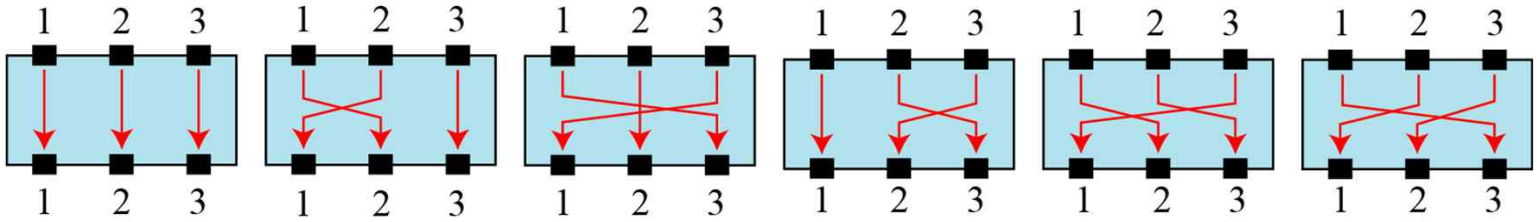


Table 5.1 Example of a 64 x 64 permutation table for a straight P-box.

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

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

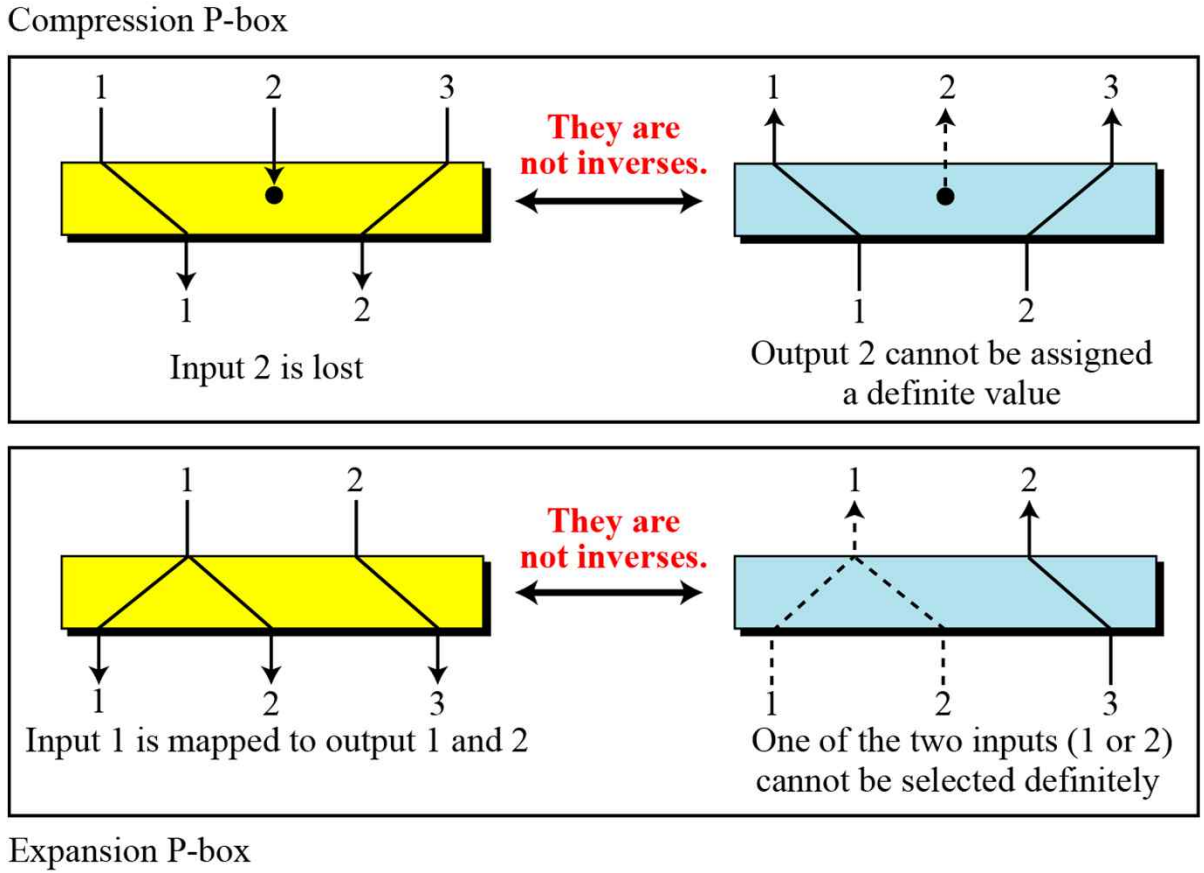
5.1.3 Continued

P-Boxes: Invertibility

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

5.1.3 Continued

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



5.1.3 Continued

S-Box

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

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

5.1.3 Continued

Example 5.8

In an S-box with three inputs and two outputs, we have

$$y_1 = x_1 \oplus x_2 \oplus x_3 \qquad y_2 = x_1$$

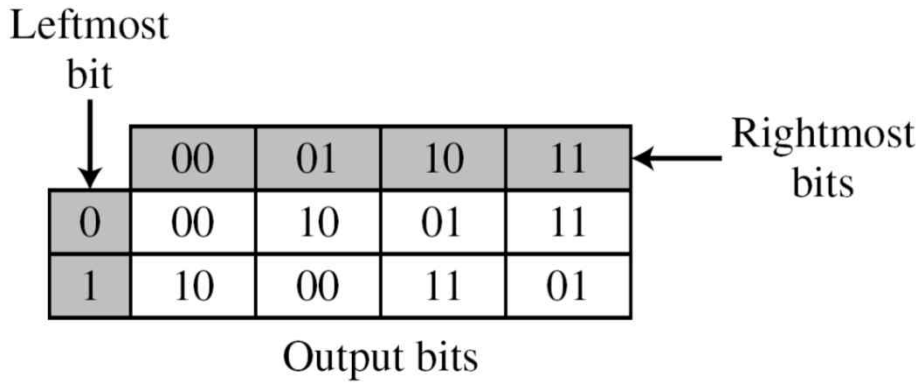
The S-box is linear because $a_{1,1} = a_{1,2} = a_{1,3} = a_{2,1} = 1$ and $a_{2,2} = a_{2,3} = 0$. The relationship can be represented by matrices, as shown below:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

5.1.3 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.

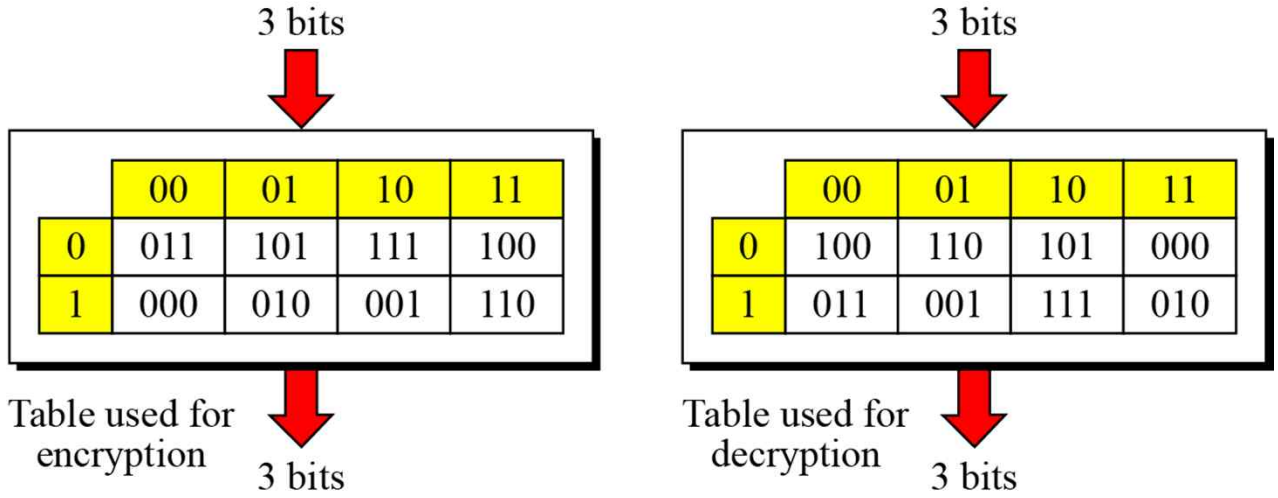
5.1.3 Continued

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.

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.

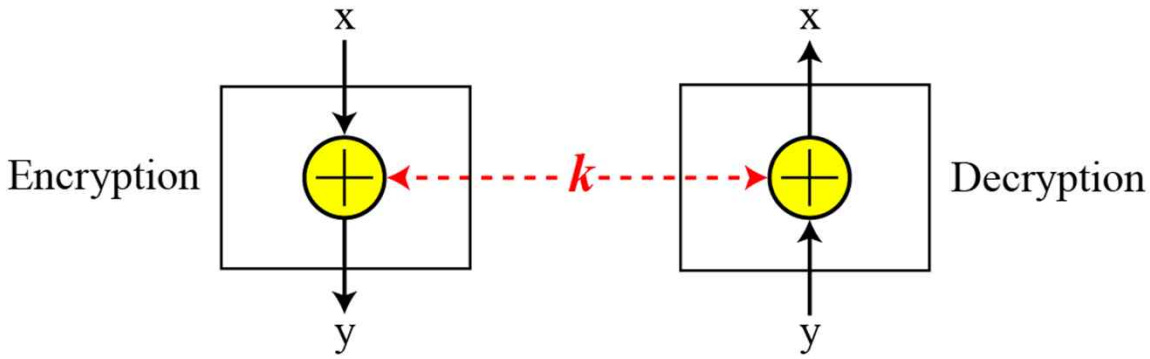


5.1.3 Continued

Exclusive-Or

An important component in most block ciphers is the exclusive-or operation. As we discussed in Chapter 4, addition and subtraction operations in the GF(2^n) field are performed by a single operation called the exclusive-or (XOR).

Figure 5.9 Invertibility of the exclusive-or operation

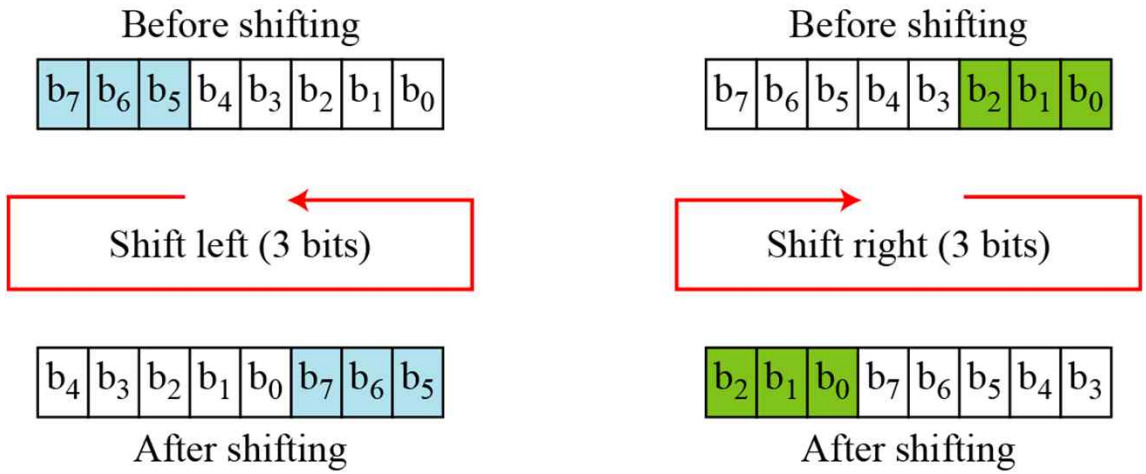


5.1.3 Continued

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

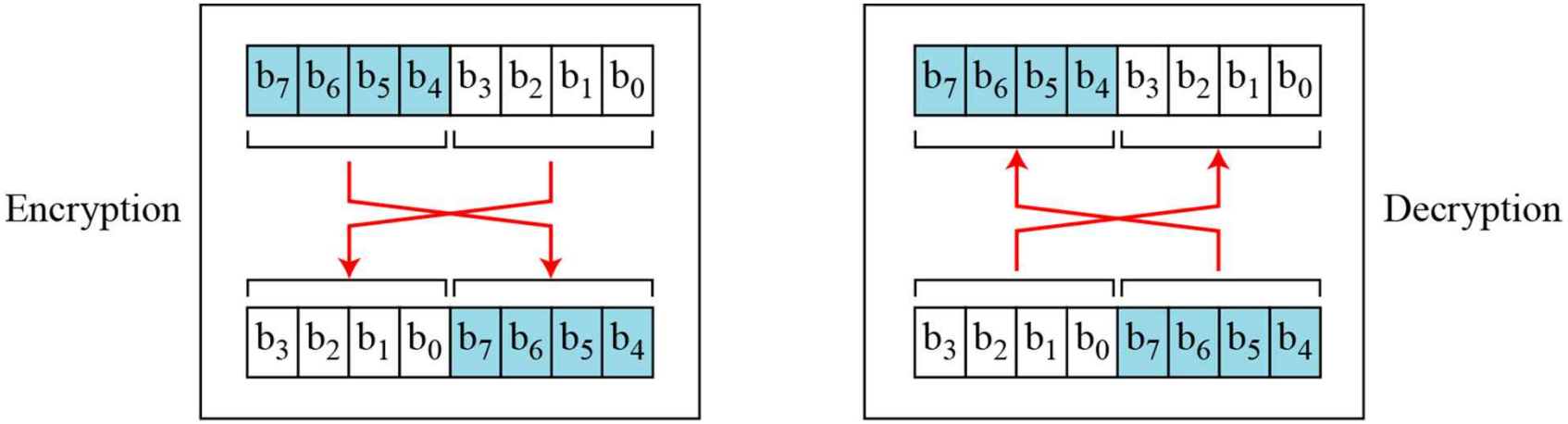


5.1.3 Continued

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

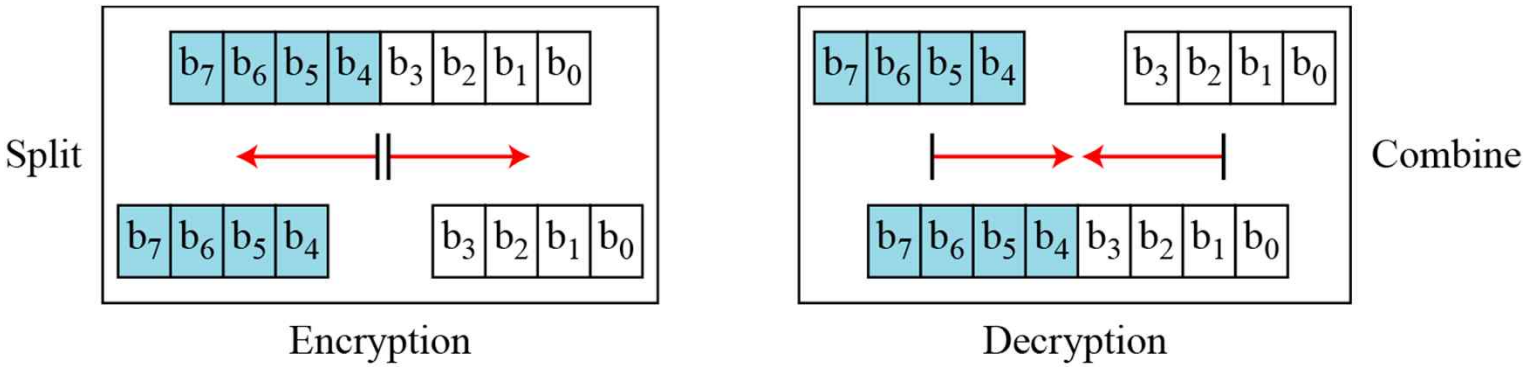


5.1.3 Continued

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



5.1.4 Product Ciphers

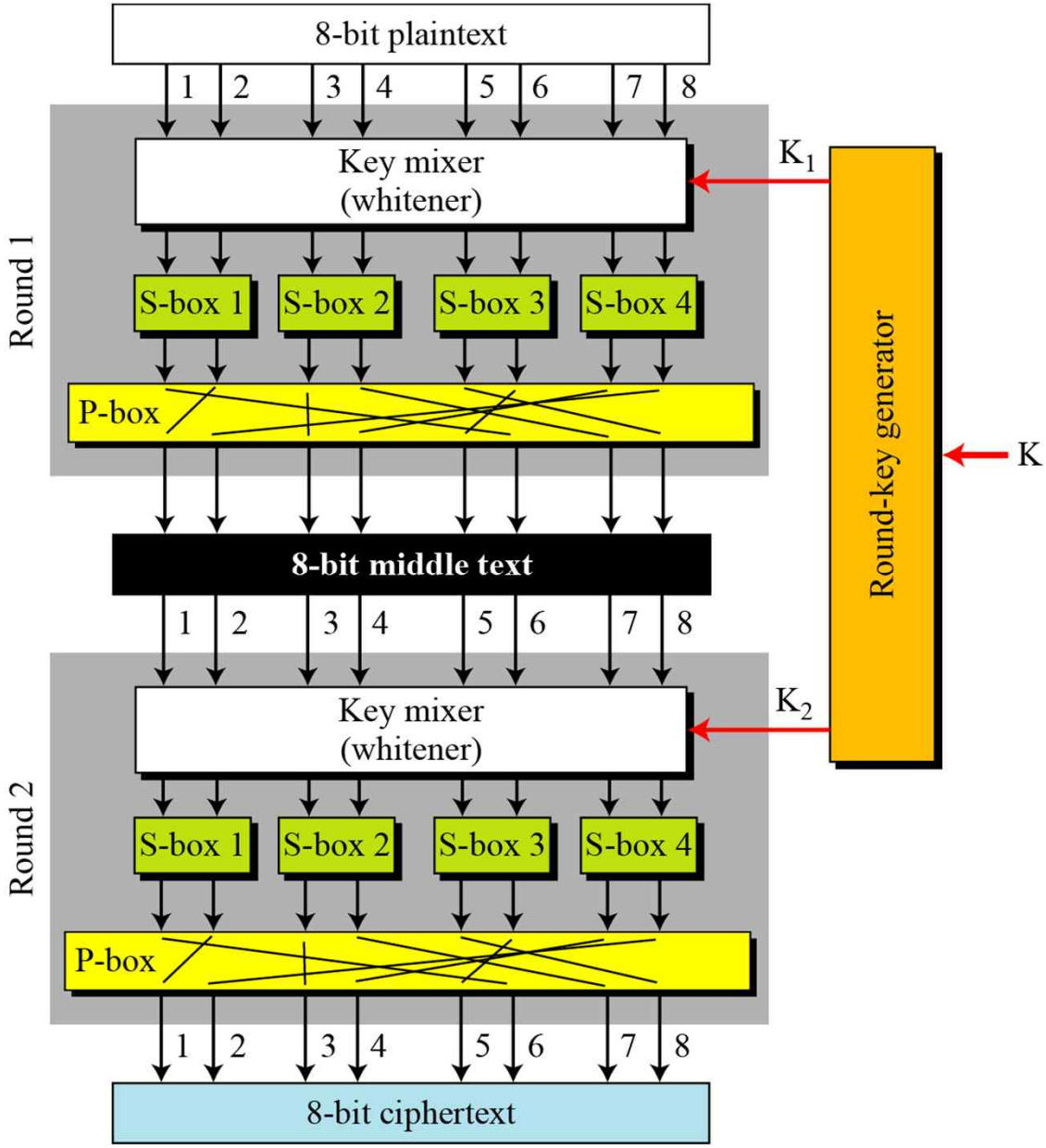
Shannon introduced the concept of a product cipher. A product cipher is a complex cipher combining substitution, permutation, and other components discussed in previous sections.

Diffusion The idea of diffusion is to hide the relationship between the ciphertext and the plaintext. If a single symbol in the plaintext is changed, several or all symbols in the ciphertext will also be changed.

Confusion The idea of confusion is to hide the relationship between the ciphertext and the key. If a single bit in the key is changed, most or all bits in the ciphertext will also be changed.

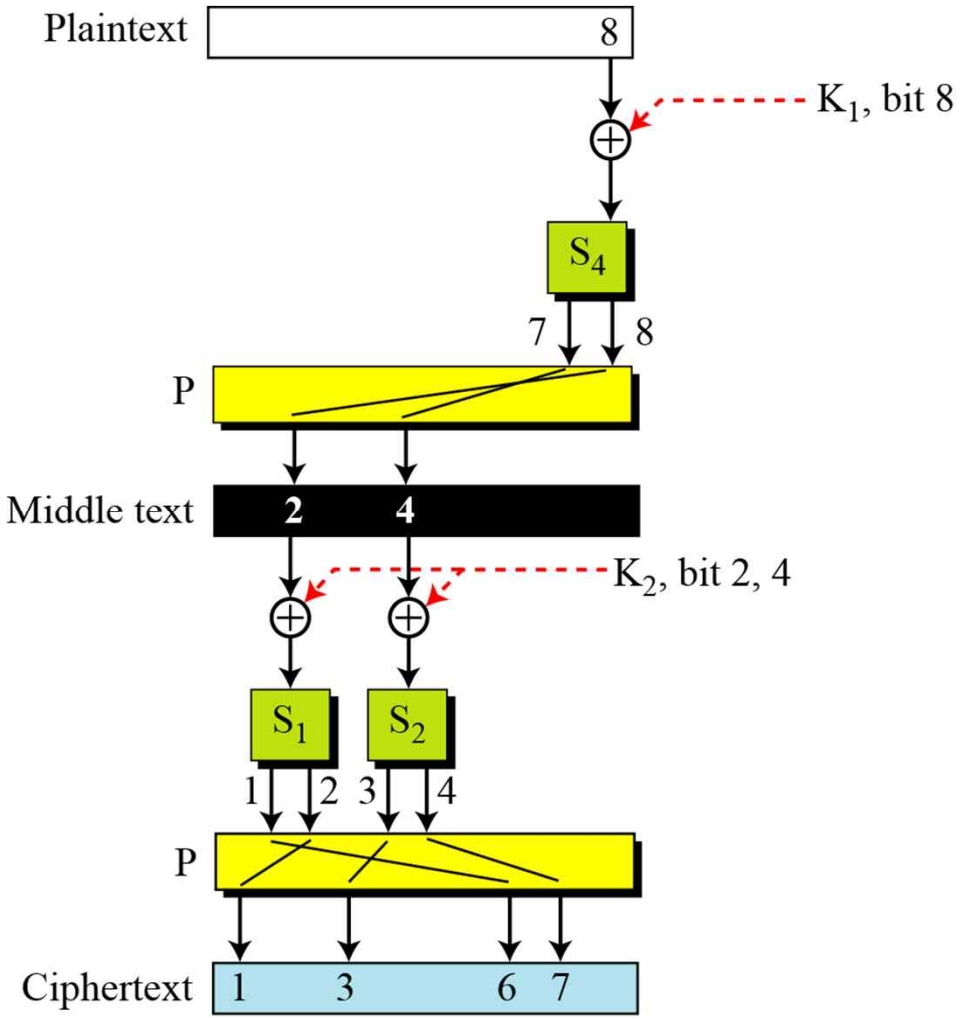
Rounds Diffusion and confusion can be achieved using iterated product ciphers where each iteration is a combination of S-boxes, P-boxes, and other components.

5.1.4 Continued A product cipher made of two rounds



5.1.4 Continued

Figure 5.14 Diffusion and confusion in a block cipher



5.1.5 Two Classes of Product Ciphers

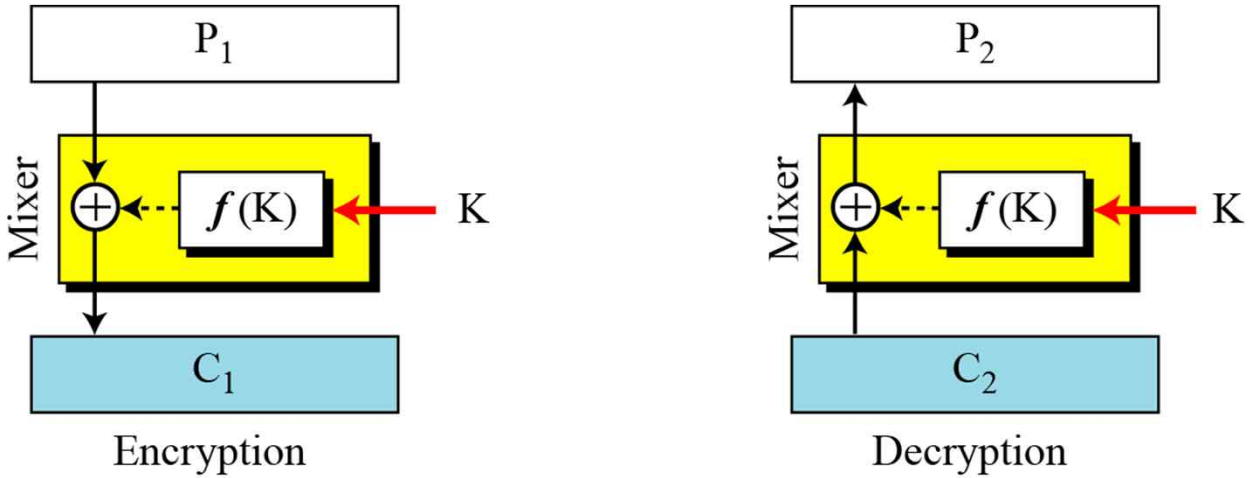
Modern block ciphers are all product ciphers, but they are divided into two classes.

1. Feistel ciphers : DES
   Has been used for decades.
   Can have three types of components :
   self-invertible, invertible, and noninvertible.

2. Non-Feistel ciphers : AES
   Uses only invertible components.
   A component in the encryption cipher has the corresponding component in the decryption cipher.

5.1.5 Continued

Figure 5.15 The first thought in Feistel cipher design



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

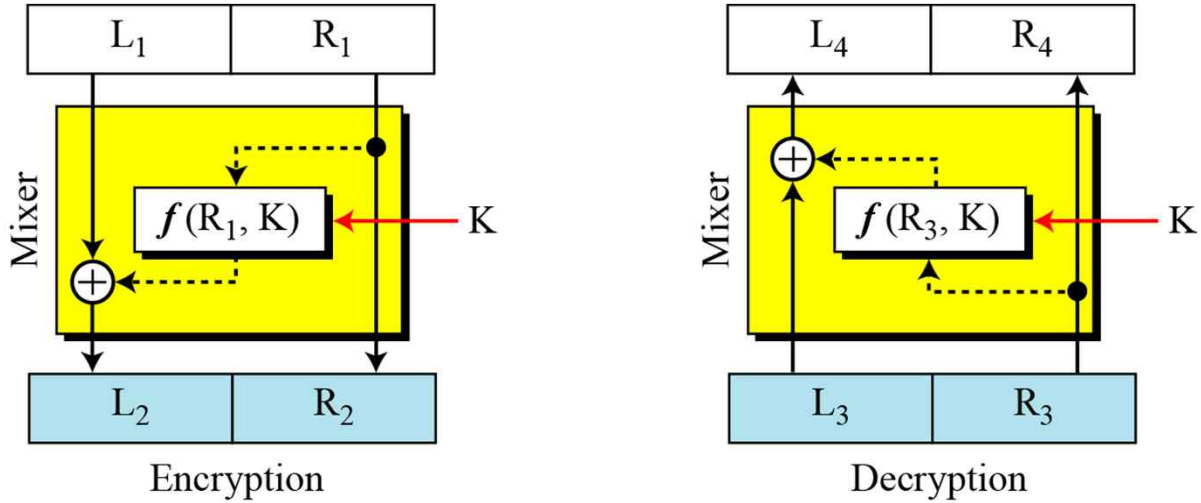
The function extracts the first and second bits to get 11 in binary or 3 in decimal. The result of squaring is 9, which is 1001 in binary.

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

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

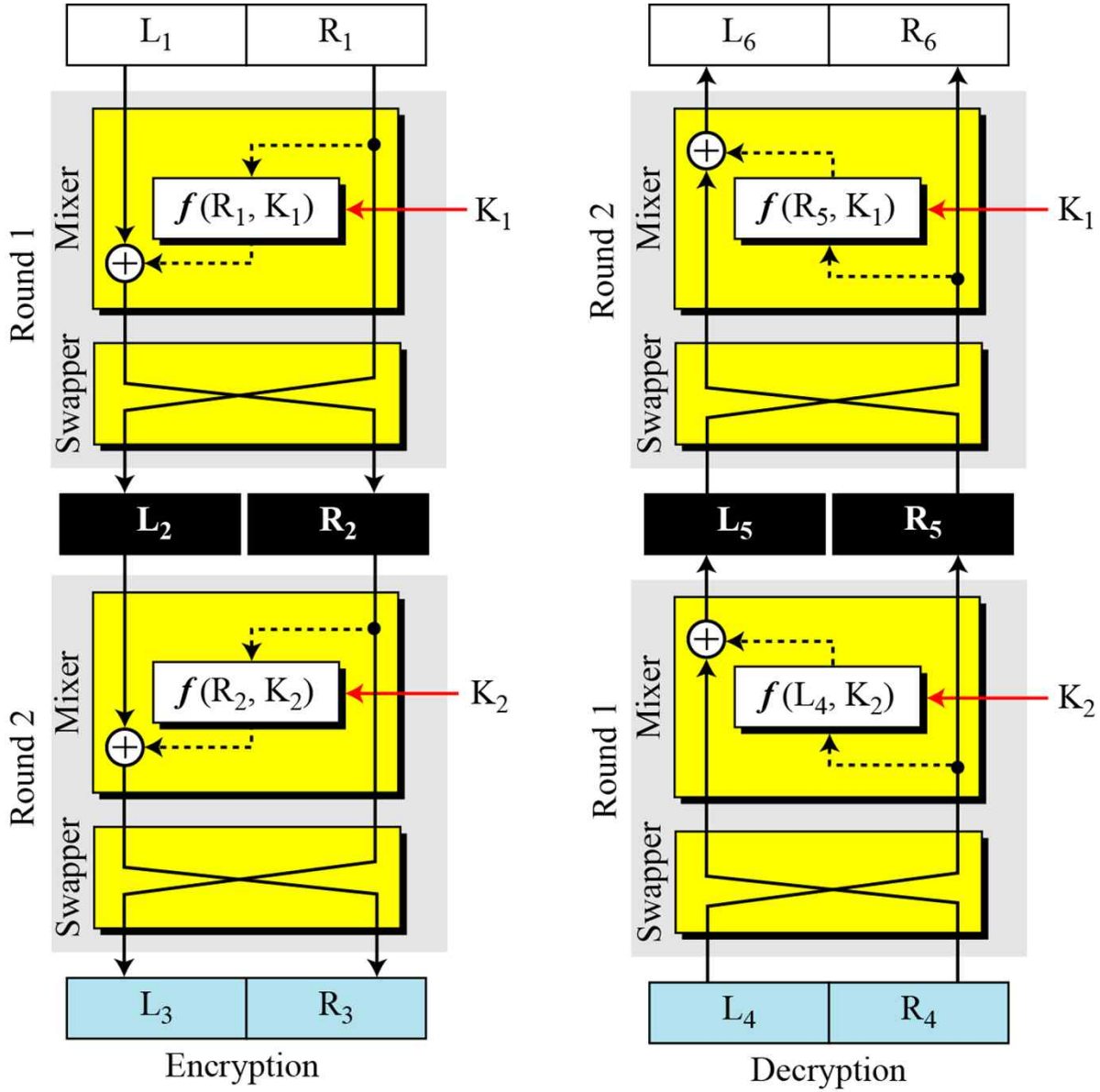
5.1.5 Continued

Figure 5.16 Improvement of the previous Feistel design



5.1.5 Continued

Figure 5.17 Final design of a Feistel cipher with two rounds



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.

Differential Cryptanalysis

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

Linear Cryptanalysis

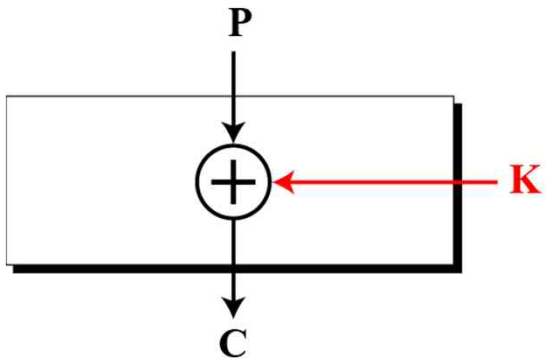
Linear cryptanalysis was presented by Mitsuru Matsui in 1993. The analysis uses known plaintext attacks.

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 $P_1 \oplus P_2$ and by ciphertext difference, we mean $C_1 \oplus C_2$. The following proves that $C_1 \oplus C_2 = P_1 \oplus P_2$:

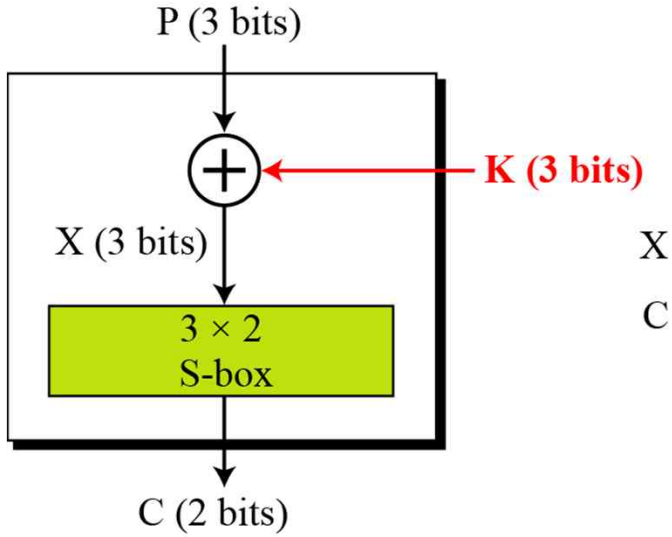
$$C_1 = P_1 \oplus K \quad C_2 = P_2 \oplus K \quad \rightarrow \quad C_1 \oplus C_2 = P_1 \oplus K \oplus P_2 \oplus K = P_1 \oplus P_2$$



5.1.6 Continued

Example 5.14

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



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 Tables 5.4 and 5.5.

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

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

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

S-box table

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

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

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

The two tests confirm that K = 011 or K =101. Although Eve is not sure what the exact value of the key is, she knows that the rightmost bit in the key is **1**.

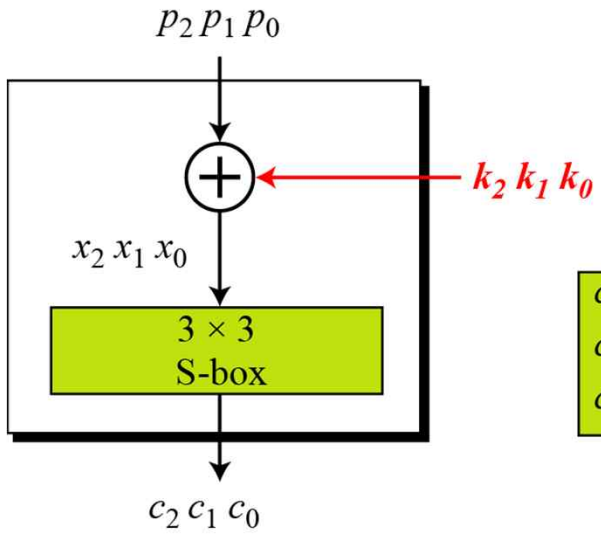
5.1.6 Continued

Linear Cryptanalysis

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

5.1.6 Continued

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



$$c_0 = x_0 \oplus x_1$$
$$c_1 = x_0 \oplus x_1 \oplus x_2$$
$$c_2 = x_1 \oplus x_2$$

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 .