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

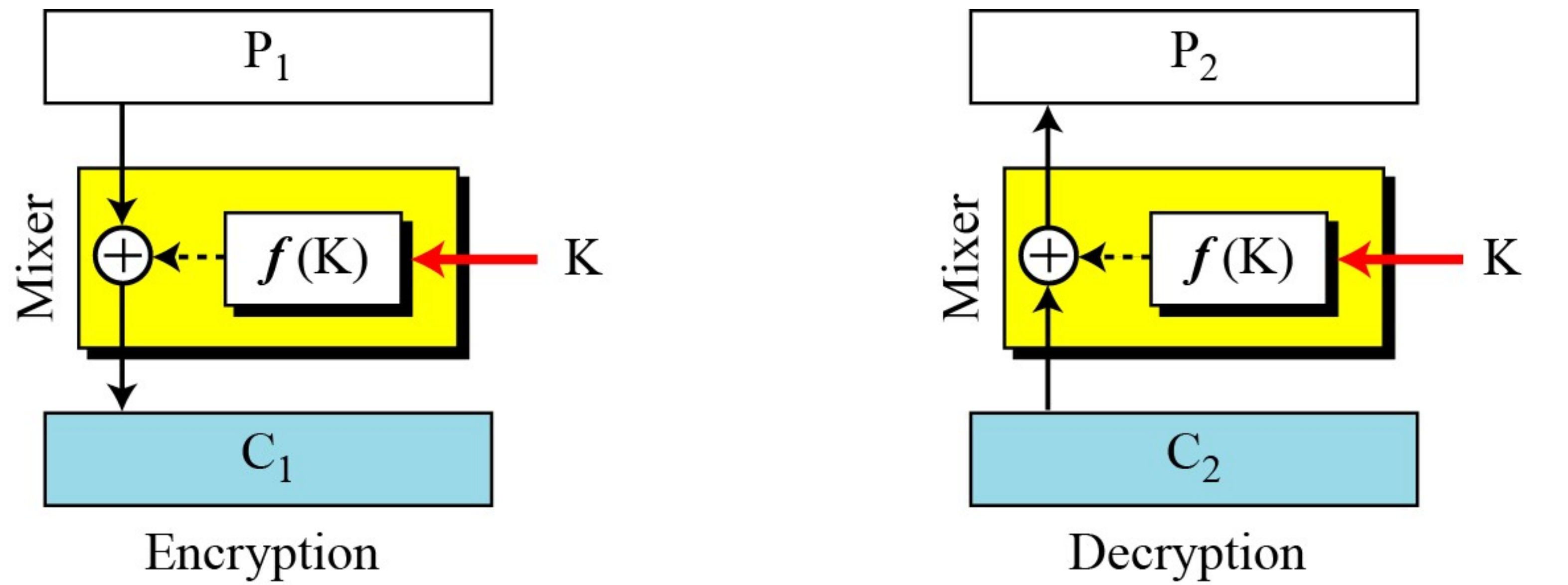
5.1.5 Continued

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.

5.1.5 Continued

Figure 5.15 *The first thought in Feistel cipher design*



Note

Diffusion hides the relationship between the ciphertext and the plaintext.

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

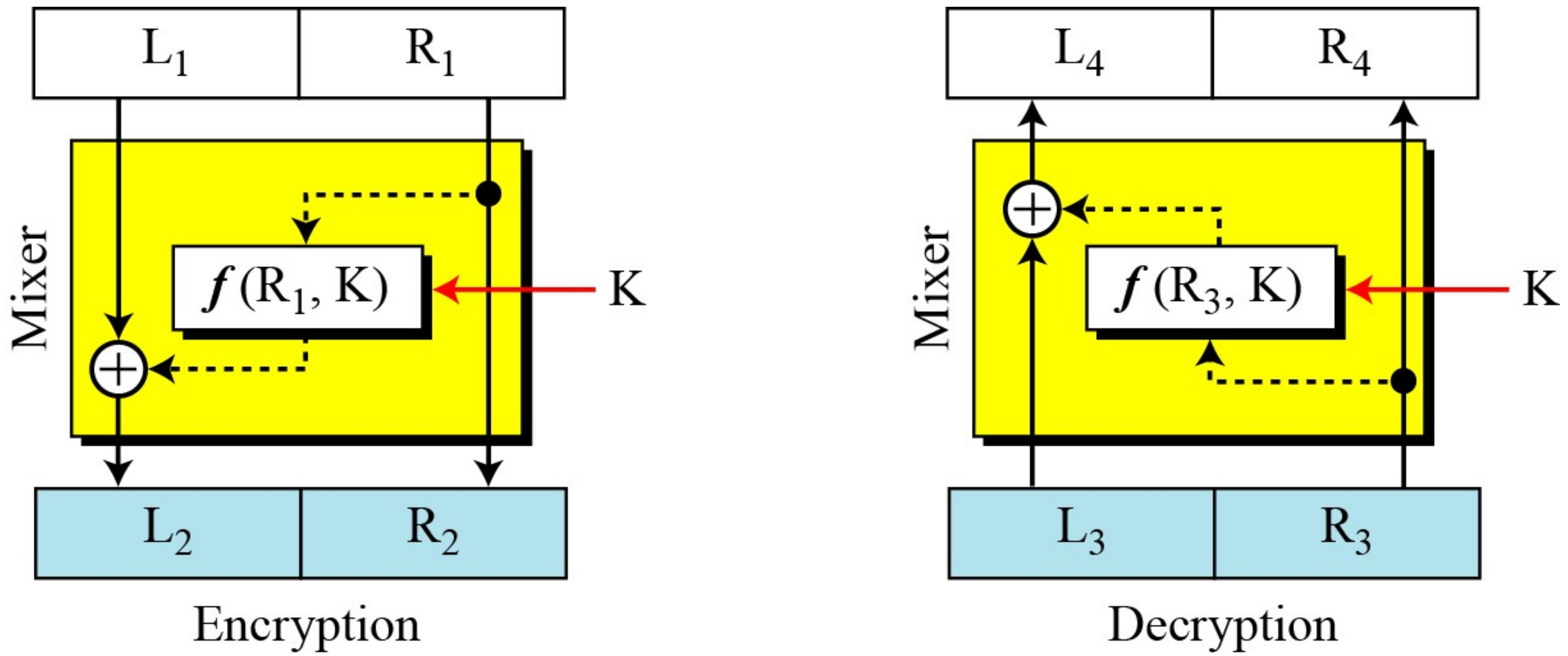
The function extracts the first and second bits to get 11 in binary or 3 in decimal. The result of squaring is 9, which is 1001 in binary.

Encryption: $C = P \oplus f\,(K) = 0111 \oplus 1001 = 1110$

Decryption: $P = C \oplus f\,(K) = 1110 \oplus 1001 = 0111$

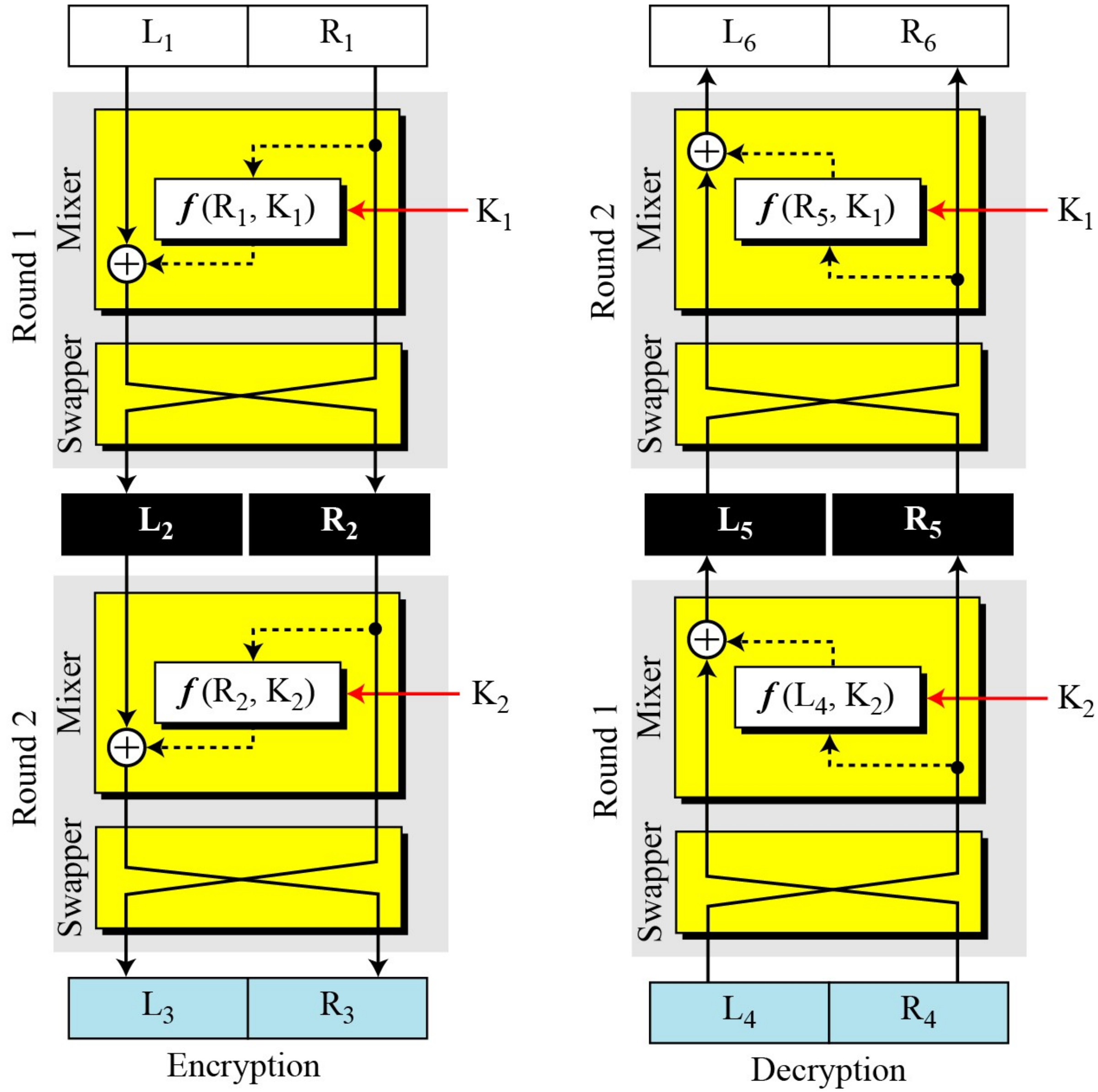
5.1.5 Continued

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***



5.1.5 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:

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

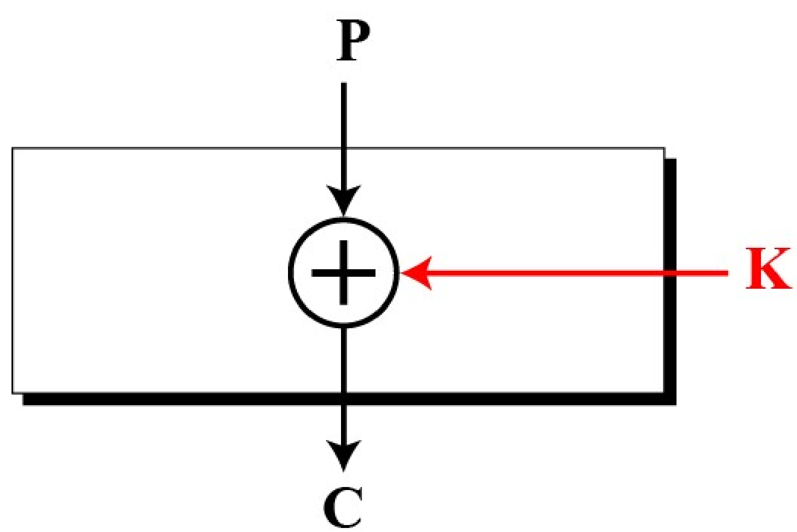


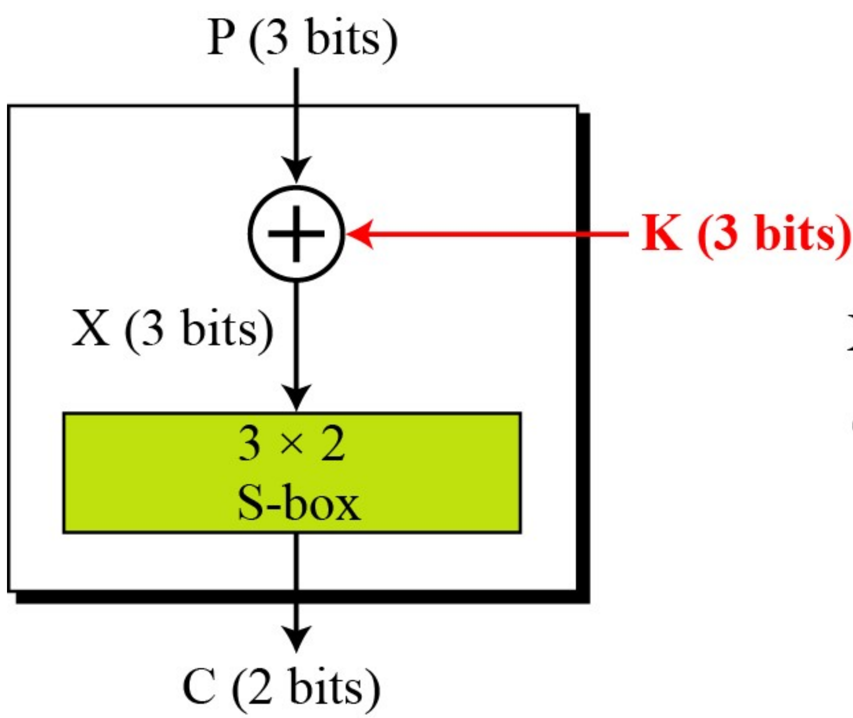
Figure 5.18 *Diagram for Example 5.13*

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

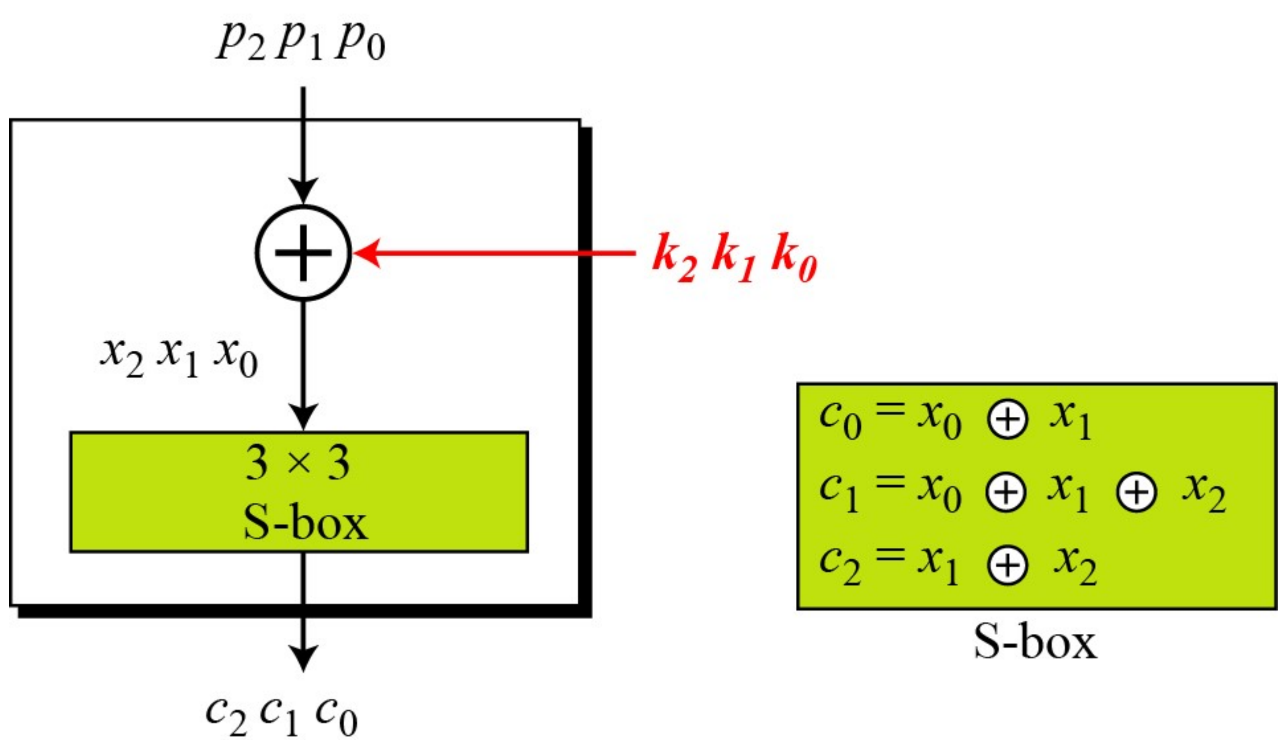
5.1.6 Continued

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*



5-2 MODERN STREAM CIPHERS

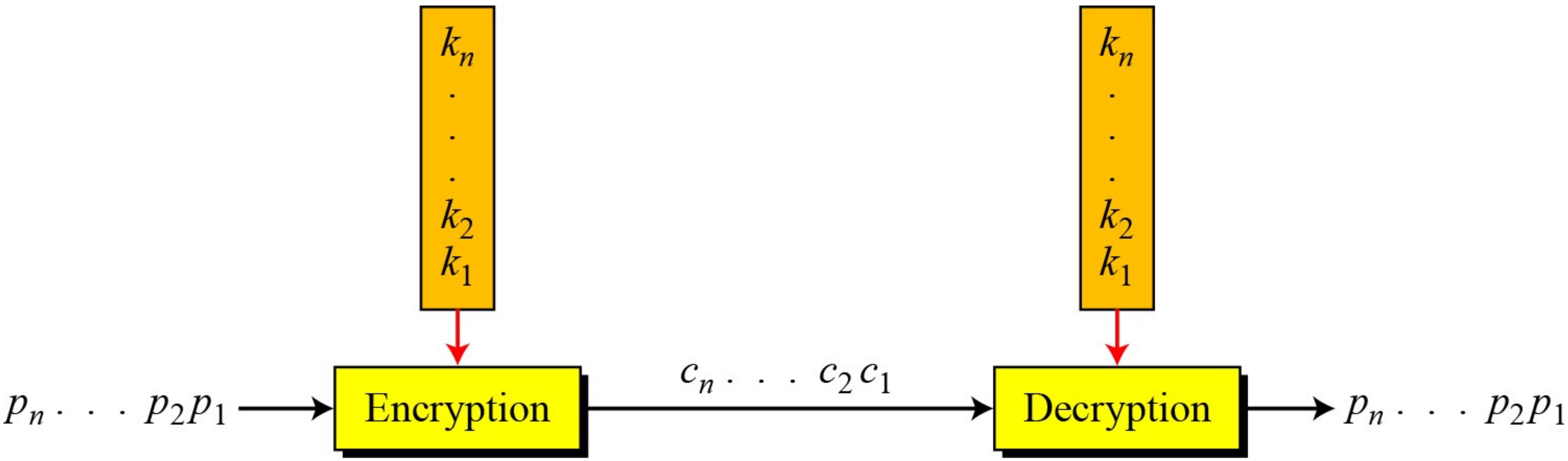
In a modern stream cipher, encryption and decryption are done r bits at a time. We have a plaintext bit stream $P = p_n \ldots p_2\ p_1$, ***a ciphertext bit stream*** $C = c_n \ldots c_2\ c_1$, ***and a key bit stream*** $K = k_n \ldots k_2\ k_1$, ***in which*** p_i, c_i, ***and*** k_i ***are r-bit words.***

Topics discussed in this section:

5.2.1 Synchronous Stream Ciphers
5.2.2 Nonsynchronous Stream Ciphers

5.2 Continued

Figure 5.20 ***Stream cipher***



Note

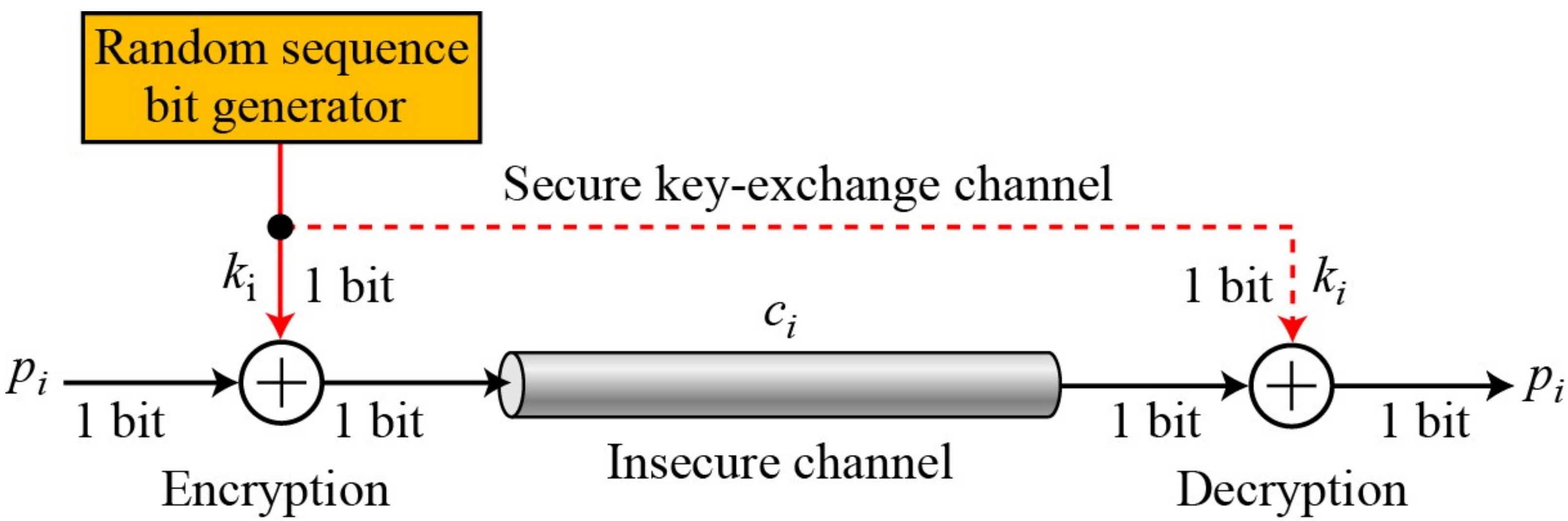
In a modern stream cipher, each *r*-bit word in the plaintext stream is enciphered using an *r*-bit word in the key stream to create the corresponding *r*-bit word in the ciphertext stream.

5.2.1 Synchronous Stream Ciphers

Note

In a synchronous stream cipher the key is independent of the plaintext or ciphertext.

Figure 5.22 ***One-time pad***



5.2.2 Nonsynchronous Stream Ciphers

In a nonsynchronous stream cipher, each key in the key stream depends on previous plaintext or ciphertext.

Note

In a nonsynchronous stream cipher, the key depends on either the plaintext or ciphertext.