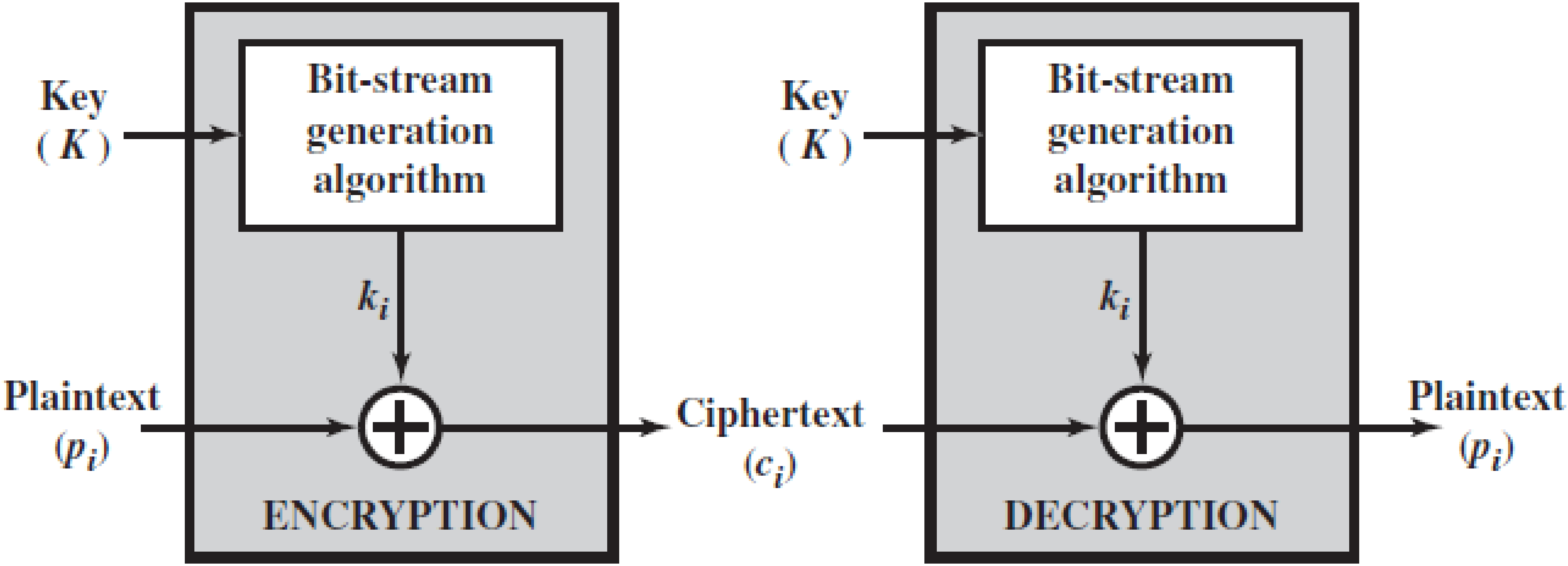
FEISTEL CIPHER

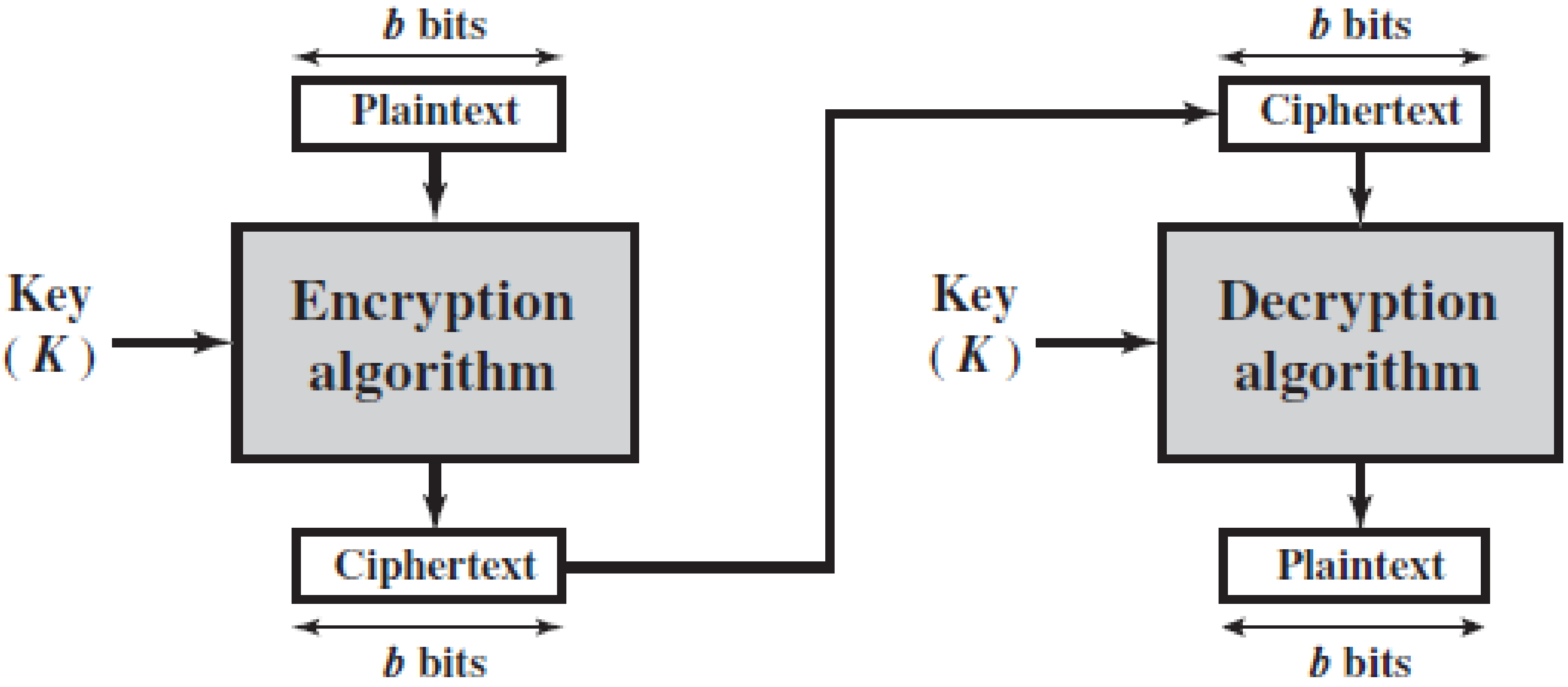
STREAM CIPHERS AND BLOCK CIPHERS

- A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. Examples of classical stream ciphers are the autokeyed Vigenère cipher and the Vernam cipher.
- The keystream must be provided to both users in advance via some independent and secure channel.
- In this approach, the bit-stream generator is a key-controlled algorithm and must produce a bit stream that is cryptographically strong.
- That is, 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 keystream.

- A **block cipher** is one in which 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.
- Using some of the modes of operation, a block cipher can be used to achieve the same effect as a stream cipher.
- In general, they seem applicable to a broader range of applications than stream ciphers. The vast majority of network-based symmetric cryptographic applications make use of block ciphers.



(a) Stream cipher using algorithmic bit-stream generator



(b) Block cipher

FEISTEL CIPHER STRUCTURE

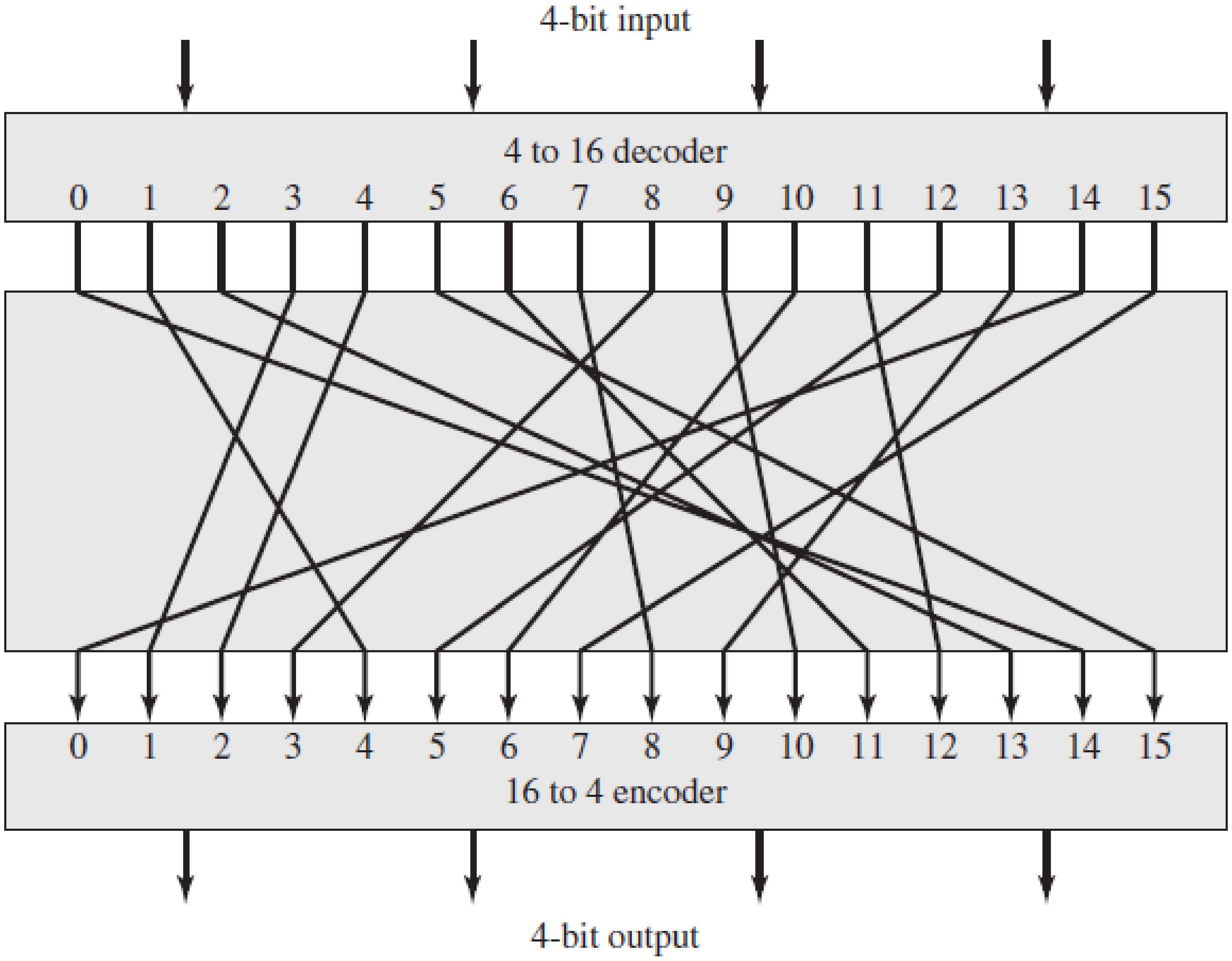
- A block cipher operates on a plaintext block of *n* bits to produce a ciphertext block of *n* bits.
- There are 2^n possible different plaintext blocks and, for the encryption to be reversible (i.e., for decryption to be possible), each must produce a unique ciphertext block. Such a transformation is called reversible, or nonsingular. The following examples illustrate nonsingular and singular transformations for $n = 2$.

Reversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	00
11	01

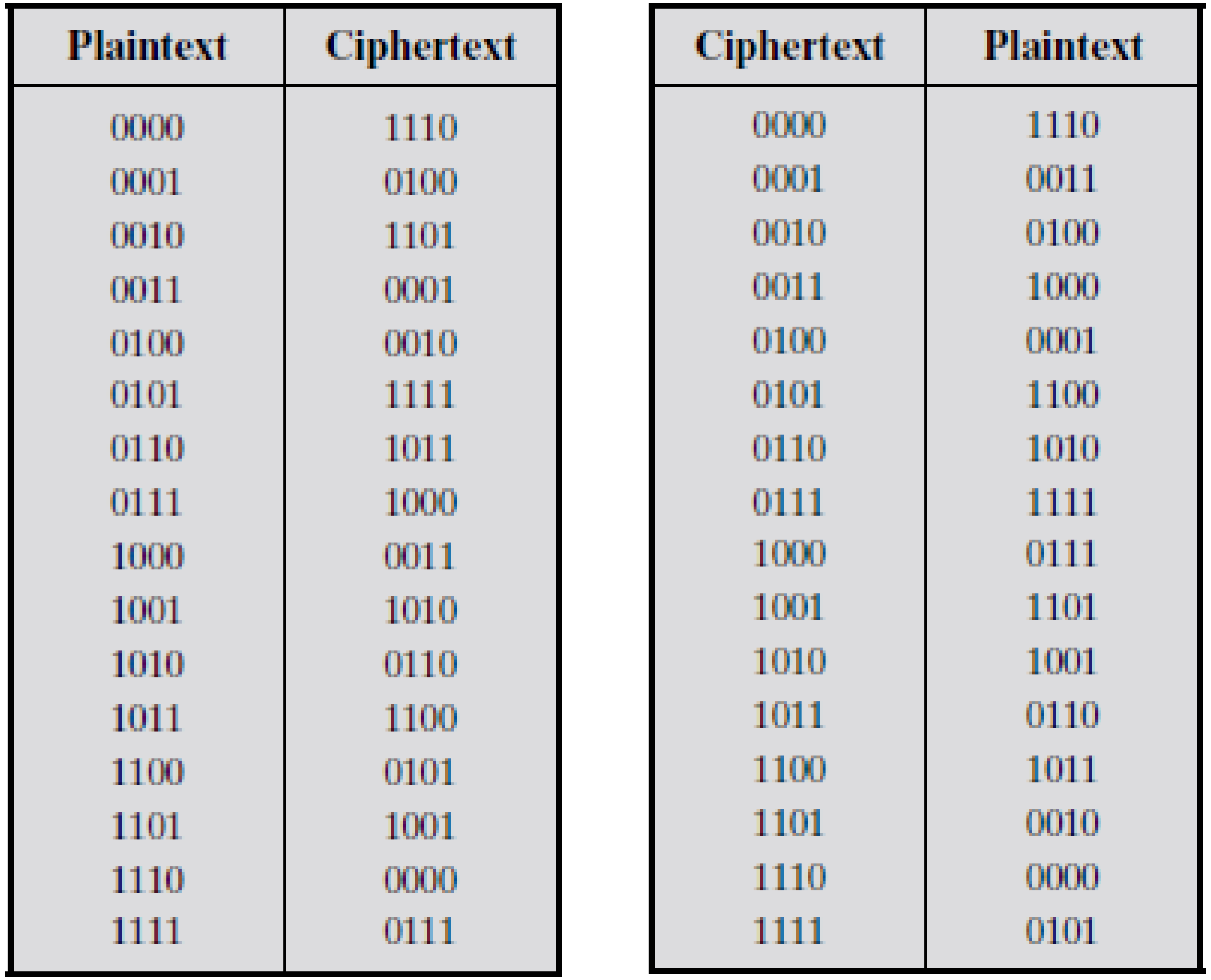
Irreversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	01
11	01



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

- Figure illustrates the logic of a general substitution cipher for $n = 4$.
- A 4-bit input produces one of 16 possible input states, which is mapped by the substitution cipher into a unique one of 16 possible output states, each of which is represented by 4 ciphertext bits.
- The encryption and decryption mappings can be defined by a tabulation, as shown in Table 3.1. This is the most general form of block cipher and can be used to define any reversible mapping between plaintext and ciphertext.



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

Table 3.1 Encryption and Decryption Tables for Substitution Cipher of Figure 3.2

- Feistel refers to this as the *ideal block cipher*, because it allows for the maximum number of possible encryption mappings from the plaintext block.
- But there is a practical problem with the ideal block cipher. If a small block size, such as *n* = 4, is used, then the system is equivalent to a classical substitution cipher. Such systems, are vulnerable to a statistical analysis of the plaintext.
- If *n* is sufficiently large and an arbitrary reversible substitution between plaintext and ciphertext is allowed, then the statistical characteristics of the source plaintext are masked to such an extent that this type of cryptanalysis is infeasible.

- An arbitrary reversible substitution cipher (the ideal block cipher) for a large block size is not practical.
- For such a transformation, the mapping itself constitutes the key. Consider again Table 3.1, which defines one particular reversible mapping from plaintext to ciphertext for n = 4.
- The mapping can be defined by the entries in the second column, which show the value of the ciphertext for each plaintext block. This, in essence, is the key that determines the specific mapping from among all possible mappings.
- In this case, using this straightforward method of defining the key, the required key length is (4 bits) * (16 rows) = 64 bits. In general, for an n-bit ideal block cipher, the length of the key defined in this fashion is $n * 2^n$ bits. For a 64-bit block, which is a desirable length to thwart statistical attacks, the required key length is $64 * 2^{64}$ = 270 1021 bits.

- Feistel proposed [FEIS73] that we can approximate the ideal block cipher by utilizing the concept of a product cipher, which is the execution of two or more simple ciphers in sequence in such a way that the final result or product is cryptographically stronger than any of the component ciphers. Feistel proposed the use of a cipher that alternates substitutions and permutations, where these terms are defined as follows:
  - **Substitution:** Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements.
  - **Permutation:** A sequence of plaintext elements is replaced by a permutation of that sequence. That is, no elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed.

- ***Diffusion and Confusion*** The terms *diffusion* and *confusion* were 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.
- The reasoning is as follows. Assume the attacker has some knowledge of the statistical characteristics of the plaintext. For example, in a human-readable message in some language, the frequency distribution of the various letters may be known. Or there may be words or phrases likely to appear in the message (probable words).
- If these statistics are in any way reflected in the ciphertext, the cryptanalyst may be able to deduce the encryption key, part of the key, or at least a set of keys likely to contain the exact key. In what Shannon refers to as a strongly ideal cipher, all statistics of the ciphertext are independent of the particular key used.

- In **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; generally, this is equivalent to having each ciphertext digit be affected by many plaintext digits. An example of diffusion is to encrypt a message $M = m1, m2, m3, \ldots$ of characters with an averaging operation:
- yn = aa
- k
- i=1
- $mn+i$b mod 26
- adding k successive letters to get a ciphertext letter yn. One can show that the statistical structure of the plaintext has been dissipated. Thus, the letter frequencies in the ciphertext will be more nearly equal than in the plaintext; the digram frequencies will also be more nearly equal, and so on. In a binary block cipher, diffusion can be achieved by repeatedly performing some permutation on the data followed by applying a function to that permutation; the effect is that bits from different positions in the original plaintext contribute to a single bit of ciphertext

- Every block cipher involves a transformation of a block of plaintext into a block of ciphertext, where the transformation depends on the key.
- The mechanism of diffusion seeks to make the statistical relationship between the plaintext and ciphertext as complex as possible in order to thwart attempts to deduce the key.
- On the other hand, **confusion** seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible, again to thwart attempts to discover the key.
- Thus, 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. This is achieved by the use of a complex substitution algorithm. In contrast, a simple linear substitution function would add little confusion.

The Feistel cipher structure, which dates back over a quarter century and which, in turn, is based on Shannon’s proposal of 1945, is the structure used by many significant symmetric block ciphers currently in use.

- ***Feistel Cipher Structure*** The inputs to the encryption algorithm are a plaintext block of length 2*w* bits and a key *K*.
- The plaintext block is divided into two halves, *L*0 and *R*0. The two halves of the data pass through *n* rounds of processing and then combine to produce the ciphertext block.
- Each round *i* has as inputs *Li*-1 and *Ri*-1 derived from the previous round, as well as a subkey *Ki* derived from the overall *K*. In general, the subkeys *Ki* are different from *K* and from each other. In Figure, 16 rounds are used, although any number of rounds could be implemented. 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 *Ki*. Another way to express this is to say that F is a function of right-half block of *w* bits and a subkey of *y* bits, which produces an output value of length w bits: *F*(*REi*, *Ki*+1).
- Following this substitution, a **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

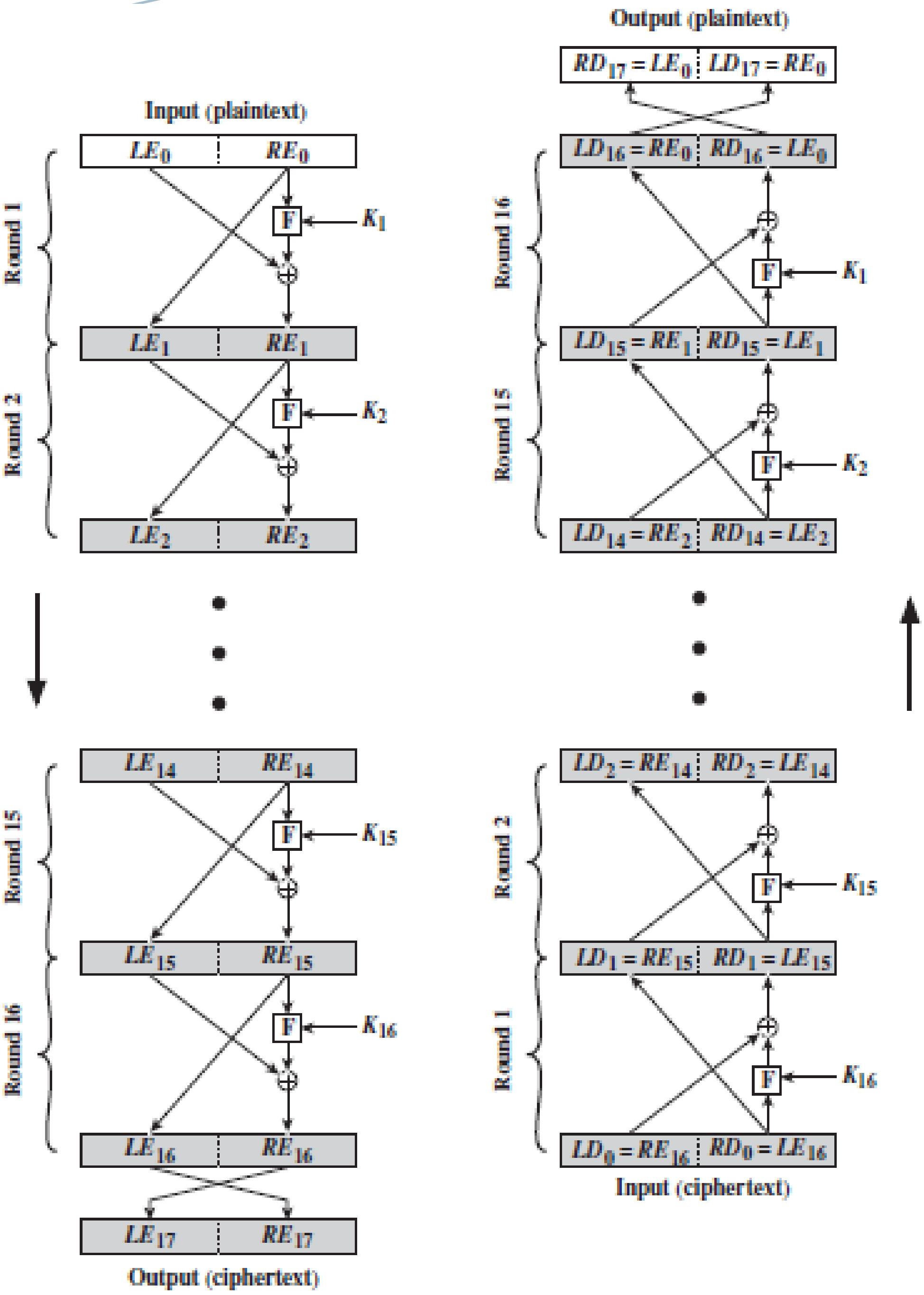


Figure 3.3 Feistel Encryption and Decryption (16 rounds)

The exact realization of a Feistel network depends on the choice of the following parameters and design features:

- **Block size:** Larger block sizes mean greater security (all other things being equal) but reduced encryption/decryption speed for a given algorithm. The greater security is achieved by greater diffusion. Traditionally, a block size of 64 bits has been considered a reasonable tradeoff and was nearly universal in block cipher design. However, the new AES uses a 128-bit block size.
- **Key size:** Larger key size means greater security but may decrease encryption/decryption speed. The greater security is achieved by greater resistance to brute-force attacks and greater confusion. Key sizes of 64 bits or less are now widely considered to be inadequate, and 128 bits has become a common size.
- **Number of rounds:** The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.
- **Subkey generation algorithm:** Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.
- **Round function F:** Again, greater complexity generally means greater resistance to cryptanalysis.

- ***Feistel Decryption Algorithm*** The process of decryption with a Feistel cipher is essentially the same as the encryption process.
- The rule is as follows: Use the ciphertext as input to the algorithm, but use the subkeys *Ki* in reverse order. That is, use *Kn* in the first round, *Kn*-1 in the second round, and so on, until *K*1 is used in the last round.
- This is a nice feature, because it means we need not implement two different algorithms; one for encryption and one for decryption. To see that the same algorithm with a reversed key order produces the correct result, Figure 3.3 shows the encryption process going down the left-hand side and the Decryption process going up the right-hand side for a 16-round algorithm.

- we use the notation *LEi* and *REi* for data traveling through the encryption algorithm and *LDi* and *RDi* for data traveling through the decryption algorithm. The diagram indicates that, at every round, the intermediate value of the decryption process is equal to the corresponding value of the encryption process with the two halves of the value swapped. To put this another way, let the output of the *i*th encryption round be *LEi* 7*REi* (*LEi* concatenated with *REi*). Then the corresponding output of the (16 – *i*) th decryption round is *REi* 7*LEi* or, equivalently, *LD*16-*i* & *RD*16-*i*.
- After the last iteration of the encryption process, the two halves of the output are swapped, so that the ciphertext is *RE*16 & *LE*16. The output of that round is the ciphertext. Now take that ciphertext and use it as input to the same algorithm.
- The input to the first round is *RE*16 & *LE*16, which is equal to the 32-bit swap of the output of the sixteenth round of the encryption process.