Symmetric ciphers and the Data Encryption Standard (DES)

Killian O’Brien

6G6Z0024 Applied Cryptography 2024/25

Lecture Week 03 – Wed 16 October 2024

Manchester Metropolitan University

Introduction to the unit

- Teaching team: Dr Killian O'Brien  and Dr Safiullah Khan . See Moodle for contact details.
- 6G6Z0024 Applied Cryptography (15 credits)
- Assessment is 100% coursework. A portfolio of exercises.
- Timetable
- Let's look at the Moodle page for the unit.
- Reading for this topic
  - Stallings, Chapter 3, Just Section 3.1: Symmetric Cipher Model
  - Stallings, Chapter 4: Block Ciphers and the Data Encryption Standard (DES)

Some definitions, (Stallings, *Cryptography and Network Security*, Ch. 3)

- **Definition - Plaintext**
  The original intelligible message or data.
- **Definition - Encryption algorithm**
  The encryption algorithm performs various substitutions and transformations of the plaintext.
- **Definition - Secret key**
  The secret key K is input into the encryption algorithm along with the plaintext. The algorithm will produce different outputs depending on the specific value of K used for the same plaintext. The exact substitutions and transformations carried out by the algorithm depend on K.
- **Definition - Ciphertext**
  The scrambled message output by the encryption algorithm. It depends on the agorithm, plaintext and key K. The ciphertext should be an apparently unintelligible random stream of data.

Symmetric Ciphers

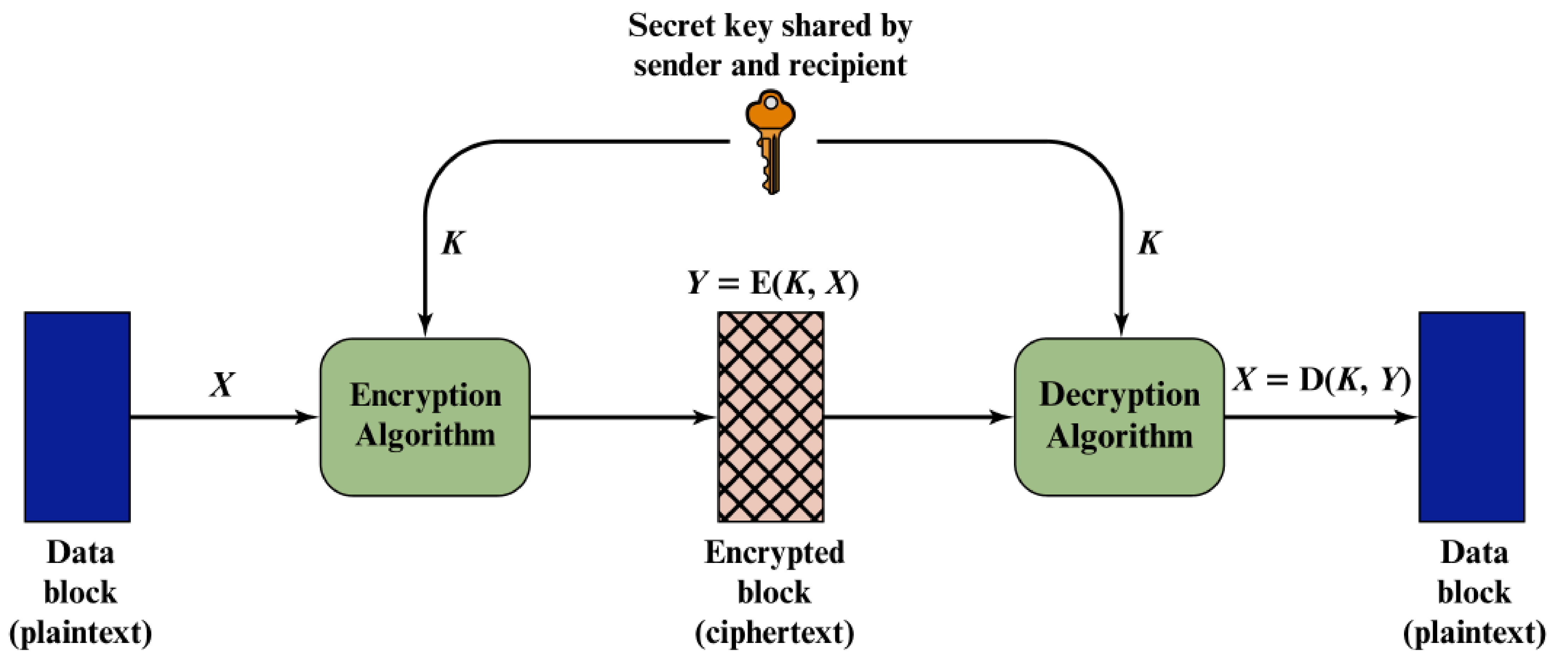


Figure 3.1 Simplified Model of Symmetric Encryption

Stallings *Cryptography and Network Security*, Sec. 3.1, pg 85

- Bob (message source) sends an encrypted message to Alice (destination)
- The cryptanalyst Eve, intercepts Y, has knowledge of the encryption and decryption algorithms, and seeks to develop estimates $\hat{X}$ and/or $\hat{K}$ of the plaintext X and key K.

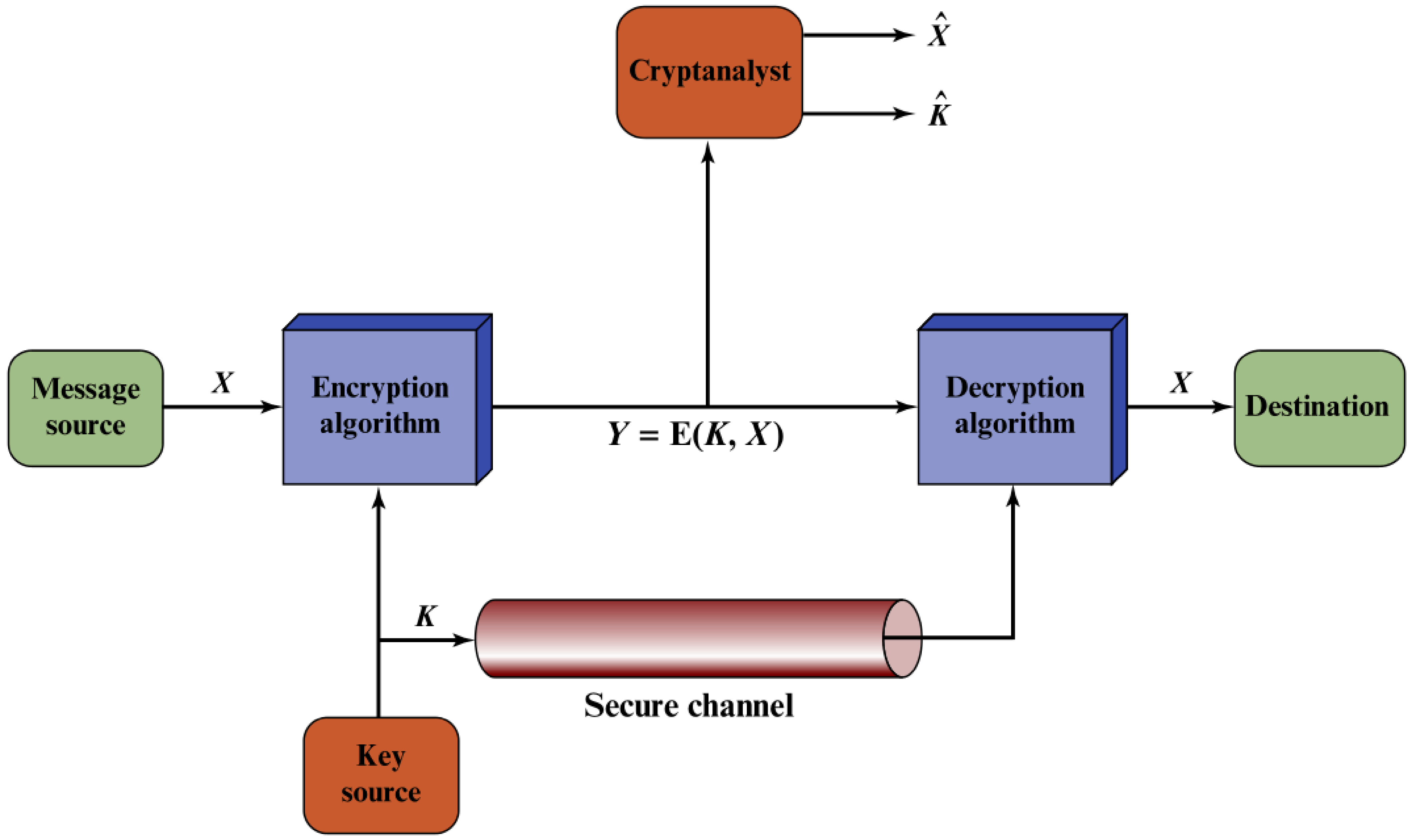


Figure 3.2 Model of Symmetric Cryptosystem

Stallings *Cryptography and Network Security*, Sec. 3.1, pg 86

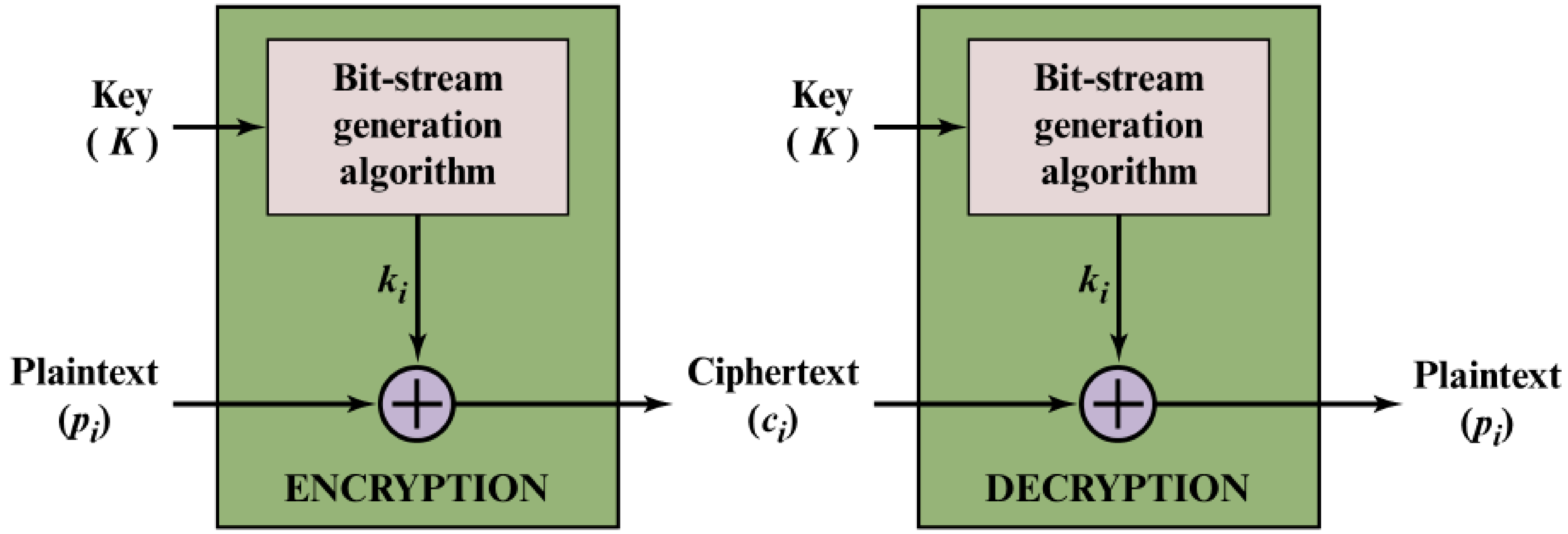
- The types of attacks carried out by Eve can be classified in various ways,

Table 3.1 Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst
Ciphertext Only	■ Encryption algorithm ■ Ciphertext
Known Plaintext	■ Encryption algorithm ■ Ciphertext ■ One or more plaintext–ciphertext pairs formed with the secret key
Chosen Plaintext	■ Encryption algorithm ■ Ciphertext ■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	■ Encryption algorithm ■ Ciphertext ■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	■ Encryption algorithm ■ Ciphertext ■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key ■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Stallings *Cryptography and Network Security*, Sec. 3.1, pg 88

- **Stream cipher**
- Considers plaintext P as a stream of individual bits, $P = (p_0, p_1, p_2, \ldots)$.
- Requires a key stream K of individual bits, $K = (k_0, k_1, k_2, \ldots)$, known only to sender and recipient.
- Encryption is by $\pmod 2$-addition-without-carry, also known as exclusive-or operation (XOR) $\oplus$.
- Ciphertext $C = (c_0, c_1, c_2, \ldots)$ computed as $c_i = p_i \oplus k_i$
    - $0 \oplus 0 = 0, 1 \oplus 1 = 0$
    - $0 \oplus 1 = 1, 1 \oplus 0 = 1$



(a) Stream cipher using algorithmic bit-stream generator

- Ideal K is so-called **one-time pad**, a random stream of bits known only to sender and recipient. But this is *impractical*.
- So some kind of keyed algorithm is used to produce the keystream K.
- More on stream ciphers later in the unit.
- Figure from Stallings, Ch 4, pg 114

- **Block cipher**
- Plaintext P divided into *blocks* of fixed bit-length b, typically 64 or 128 bits used.
- Encryption and decryption algortihms depend on same key K, known only to sender and recipient.
- More widely used design than stream ciphers.
- Provides a basic encryption/decryption component that can be used to build further ciphers, through so-called *modes of operation*. More on this later.
- Figure from Stallings, Ch 4, pg 114

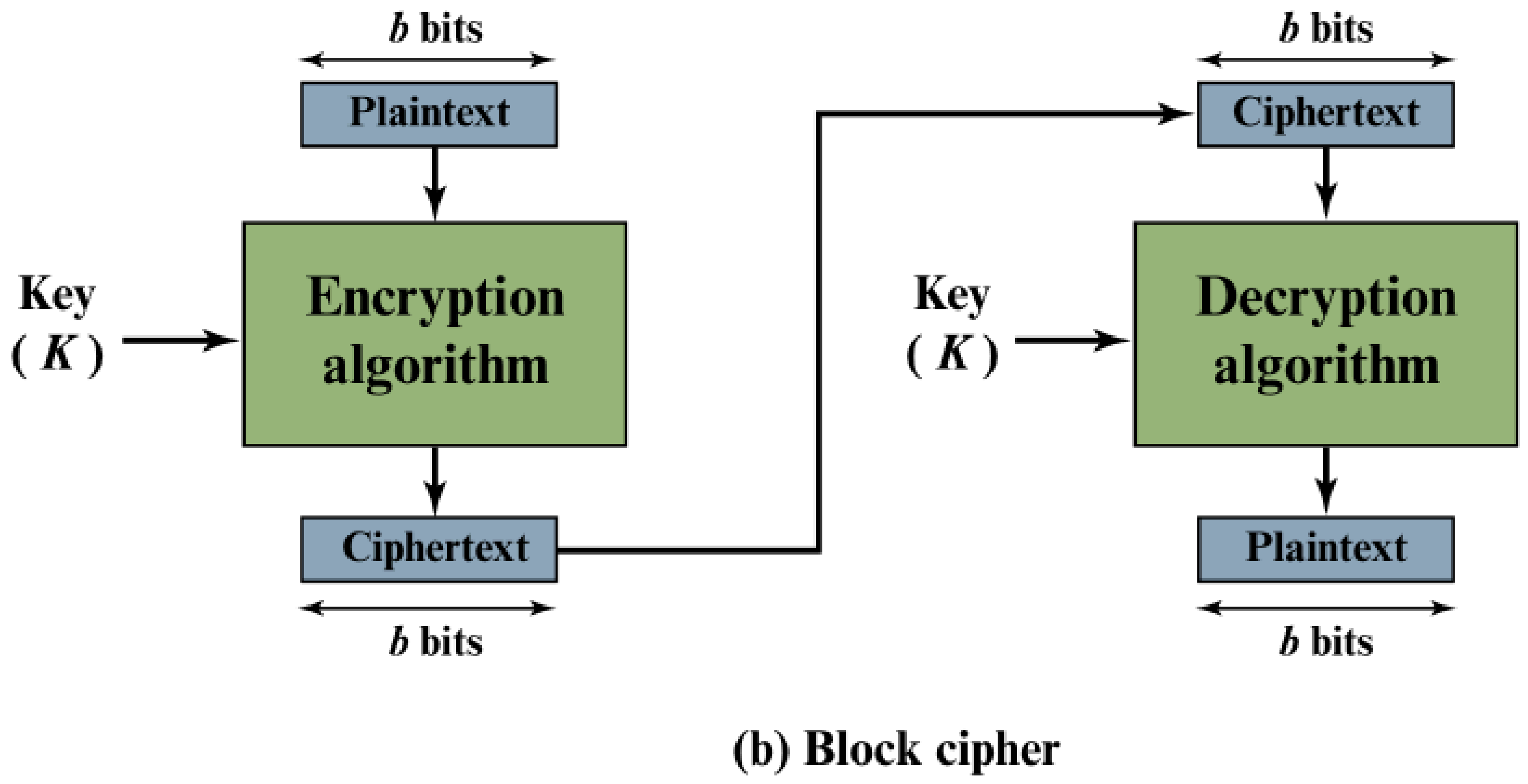


Figure 4.1 Stream Cipher and Block Cipher

- **Outlining the possiblities**
- The encryption algorithm needs to map blocks of bit-length n to blocks of bit-length n.
- There are 2^n possible blocks of length n.
- The mapping needs to be *reversible*, i.e. a so-called *permutation* or *non-singular transformation*.
- There are $(2^n)!$ such transformations to choose from.
- The factorial operator $!$ is defined as

$$N! = N \cdot (N-1) \cdot (N-2) \cdot \cdots \cdot 3 \cdot 2 \cdot 1.$$

- An example for $n = 4$ shown on the right.
- The key K is, in effect, the whole mapping table.
- However, with short block lengths, known statistical properties of the plaintexts would leak through to the ciphertexts and allow attacks, such as *frequency analysis*.
- So in practice the block bit-length needs to be large, eg. $n = 64$ or 128.
- But then the size of the mapping table is **very big** e.g. 2^{64} or 2^{128}, which makes it hard to manage K and keep it secure.
- So instead, require some way to base block ciphers on *smaller keys*.

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

Feistel ciphers

- A Feistel cipher uses a block length of n bits and a key of length k bits. So there are 2^k possible keys.
- It employs combinations of the two principles of **substitution** and **permutation** to achieve security.
    - **Definition - substitution**

      Each plaintext element is uniquely replaced by a corresponding ciphertext element.
    - **Definition - permutation**

      A sequence of plaintext elements is replaced by a permutation of that sequence. So no new elements are added or deleted, rather the order the elements appear in the sequence is changed.
- These correspond to the theoretical principles of **diffusion** and **confusion** developed by Claude Shannon. See Stallings chapter 4 for discussion.

Feistel cipher structure

- Feistel cipher diagram
- Encryption down the left hand side
- Plaintext of block length $2w$ divided into two halves, LE_0 and RE_0.
- Repeated rounds of processing applied.
- Round i takes inputs LE_{i-1}, LR_{i-1} and a subkey K_i, derived from the overall key K, and uses a **round function** F.
- A **substitution** applied to LE_{i-1} to define RE_i by by

$$RE_i = F(RE_{i-1}, K_i) \oplus LE_{i-1}.$$

- $\oplus$ is bit-wise XOR operation.
- A **permutation** is then applied for the round to output

$$LE_i := RE_{i-1} \quad \text{and } RE_i = F(RE_{i-1}, K_i) \oplus LE_{i-1}.$$

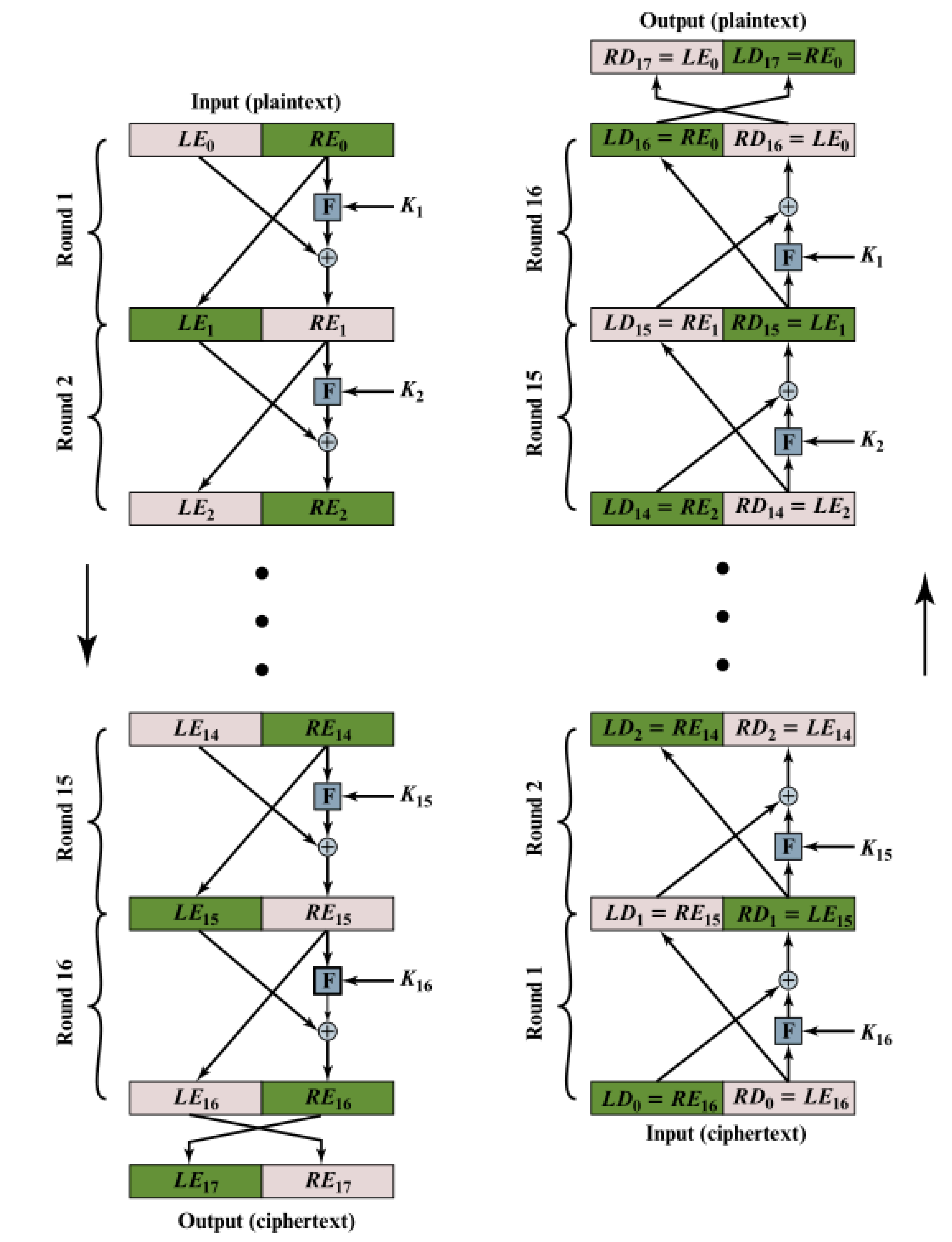


Figure 4.3 Feistel Encryption and Decryption (16 rounds)

- Decryption takes place up the right hand side.
- Ciphertext divided into two halves, $LD_0 = RE_{16}$ and $RD_0 = LE_{16}$.
- Round i will output

$$LD_i := RD_{i-1} \quad \text{and } RD_i = F(RD_{i-1}, K_{16-(i-1)}) \oplus LD_{i-1}.$$

- Note that the output of decryption round i will be the swap of the two halves of the input to encryption round $16-(i-1)$, for example

$$LD_1 = RD_0 = LE_{16} = RE_{15}, \text{ and}$$

$$RD_1 = LD_0 \oplus F(RD_0, K_{16}) = RE_{16} \oplus F(RE_{15}, K_{16})$$

- But notice that

$$RE_{16} \oplus F(RE_{15}, K_{16}) = \Big(LE_{15} \oplus F(RE_{15}, K_{16})\Big) \oplus F(RE_{15}, K_{16})$$

- But $\oplus$ satisfies $(x \oplus y) \oplus y = x \oplus (y \oplus y) = x \oplus 0 = x$.
- So in summary

$$LD_1 = RE_{15} \text{ and } RD_1 = LE_{15}.$$

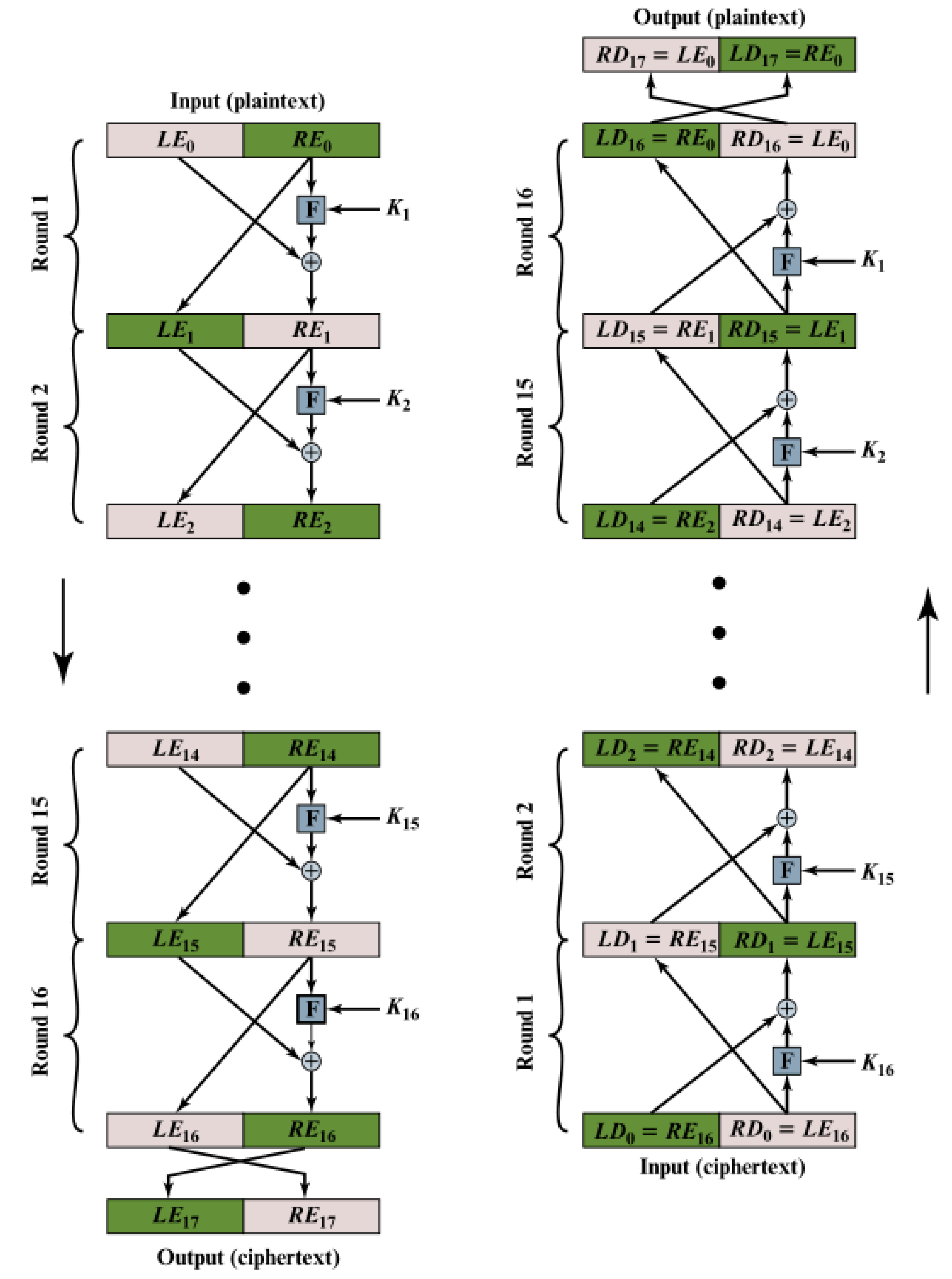


Figure 4.3 Feistel Encryption and Decryption (16 rounds)

Feistel cipher structure

- The repeated **substitutions** using F and **permutations** ensure that the original plaintext is strongly encrypted.
- Exact implementation of a Fesitel cipher will depend on:
  - **Block size**: Larger size means more security, but slower computation speed. A trade-off of 64 bits has traditionally been used. However, the newer scheme AES uses 128-bit blocks.
  - **Key size**: Again, larger means more secure but may decrease computation speed. Key sizes of less than 64 bits now considered inadequate and 128 bits or longer has become common.
  - **Number of rounds**: More is more secure, but longer computation times. Typical size is 16 rounds.
  - **Sub-key generation algorithm**: Greater complexity in this will enhance security.
  - **Round function** F: Greater complexity in this will enhance security.

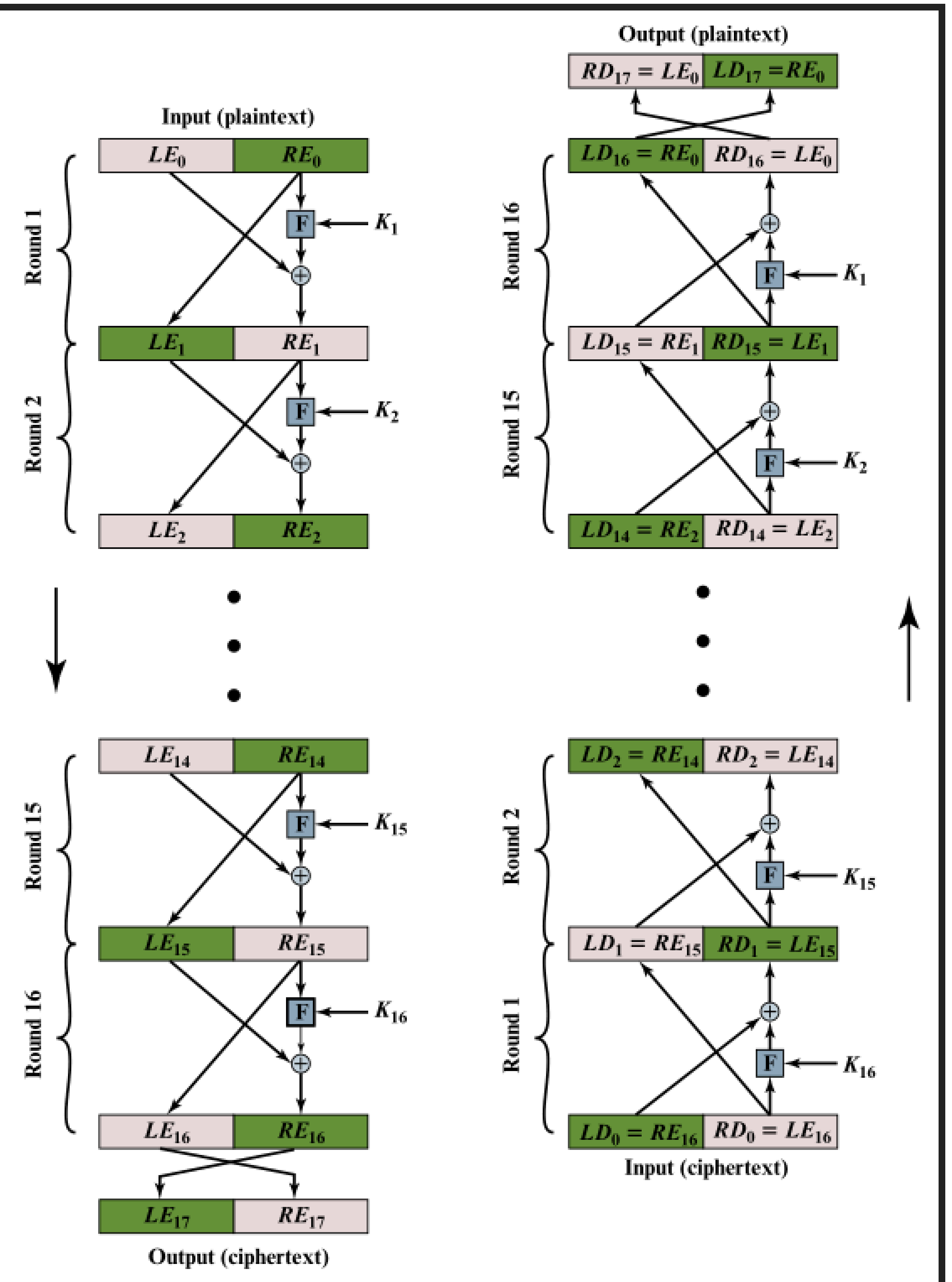


Figure 4.3 Feistel Encryption and Decryption (16 rounds)

Data Encryption Standard (DES)

- DES follows the Feistel cipher structure with added steps of an initial permutation of the plaintext and a corresponding final inverse initial permutation step.
- Precise details are involved. See Appendix C of Stallings for specifications of
    - initial permutation,
    - round permutations
    - round function F
    - sub key generation algorithm
- NIST = National Institute of Standards and Technology, a US government standards body.
- DES issued by NIST in 1977
- In 1999 advised to only use DES for legacy systems and instead advised triple-DES.
- Advanced Encryption Standard (AES) issued by NIST in 2001 and recommended over DES.

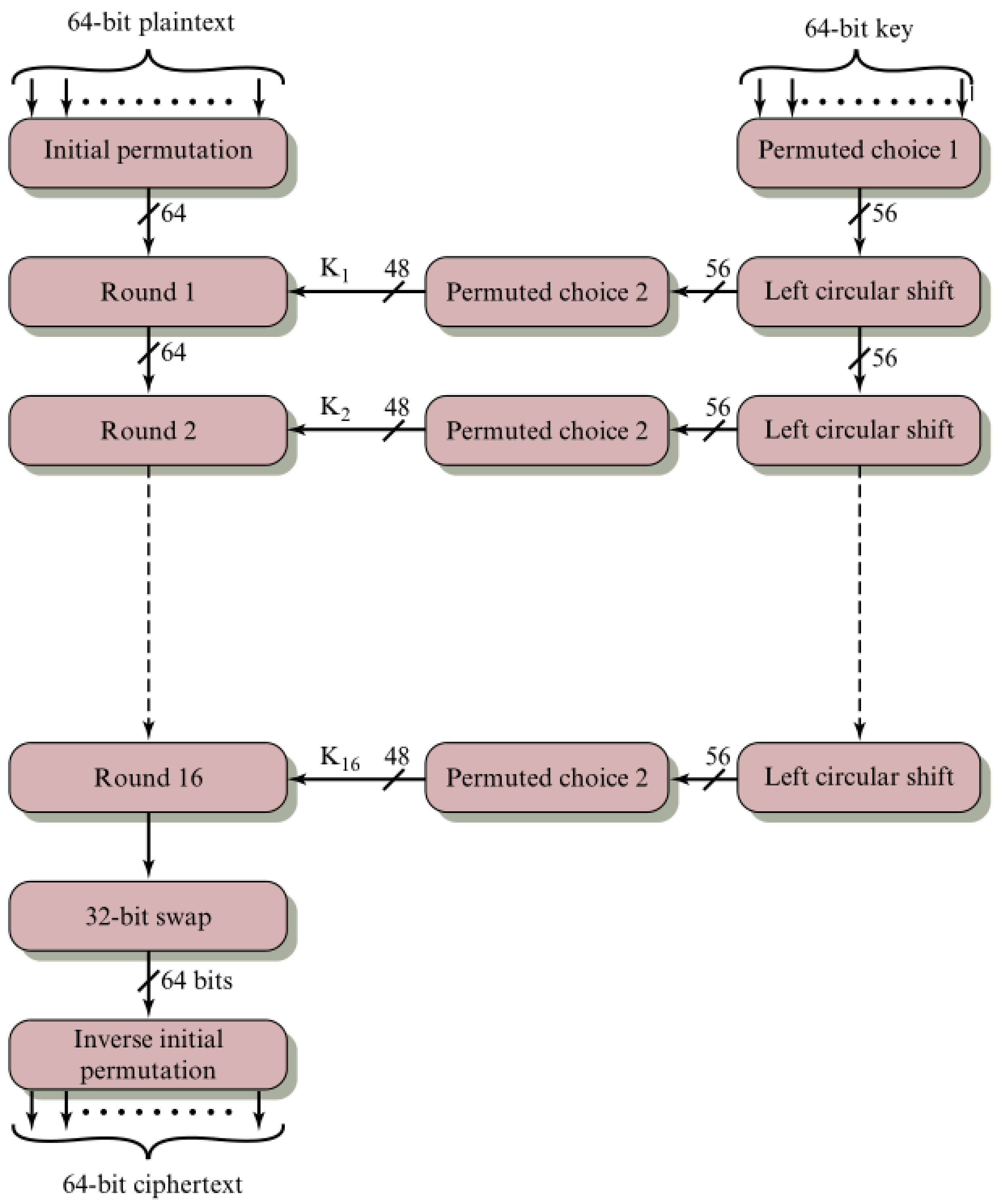


Figure 4.5 General Depiction of DES Encryption Algorithm

Avalanche effect in DES

For convenience, 64-bit blocks are presented as 16 digit hexadecimal values, where the digits

`0,1,2,...,8,9,a, ... ,f`

denote the 4-bit values

`0000,0001,0010,..., 1000,1001,1010,1111`

Avalanche effect in DES

- Table 4.3 from Stallings shows the effect of DES on plaintext blocks that differ only in a single bit, their fourth bit position
- Middle column shows intermediate states of the block.
- δ column counts the number of bit positions where the intermediate blocks differ.
- Note the way δ increases rapidly.
- By the end $\delta = 32$, which is the expected number of positions for two randomly selected 64-bit blocks to differ in.
- The small change in inputs has **avalanched** through DES and heavily affected the output. This is one source of security of DES and Feistel ciphers in general.

Table 4.3 Avalanche Effect in DES: Change in Plaintext

Round		δ	Round		δ
	02468aceeca86420 12468aceeca86420	1	9	c11bfc09887fbc6c 99f911532eed7d94	32
1	3cf03c0fbad22845 3cf03c0fbad32845	1	10	887fbc6c600f7e8b 2eed7d94d0f23094	34
2	bad2284599e9b723 bad3284539a9b7a3	5	11	600f7e8bf596506e d0f23094455da9c4	37
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18	12	f596506e738538b8 455da9c47f6e3cf3	31
4	0bae3b9e42415649 171cb8b3ccaca55e	34	13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
5	4241564918b3fa41 ccaca55ed16c3653	37	14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
6	18b3fa419616fe23 d16c3653cf402c68	33	15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
7	9616fe2367117cf2 cf402c682b2cefbc	32	16	75e8fd8f25896490 1ce2e6dc365e5f59	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33	IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

Avalanche effect in DES

- Table 4.4 from Stallings shows the effect of DES on the same plaintext block `02468aceeca86420` but where two differen keys have been used.
- The two keys are `0f1571c947d9e859` and `1f1571c947d9e859`, so again, differing only in their fourth bit position.
- Middle column shows intermediate states of the block.
- δ column counts the number of bit positions where the intermediate blocks differ.
- Note the way δ increases rapidly.
- By the end $\delta = 30$, which is near the expected number of positions for two randomly selected 64-bit blocks to differ in.
- The small change in keys has **avalanched** through DES and heavily affected the output. This avalanching effect due to small differences in keys is another source of security of DES and Feistel ciphers in general.

Table 4.4 Avalanche Effect in DES: Change in Key

Round		δ
	02468aceeca86420 02468aceeca86420	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	27
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP^{-1}	da02ce3a89ecac3b ee92b50606b62b0b	30