# Chapter 3

# Traditional Symmetric-Key Ciphers

# Chapter 3

## Objectives

- To define the terms and the concepts of symmetric key ciphers
- To emphasize the two categories of traditional ciphers: substitution and transposition ciphers
- To describe the categories of cryptanalysis used to break the symmetric ciphers
- To introduce the concepts of the stream ciphers and block ciphers
- To discuss some very dominant ciphers used in the past, such as the Enigma machine

# 3-1 INTRODUCTION

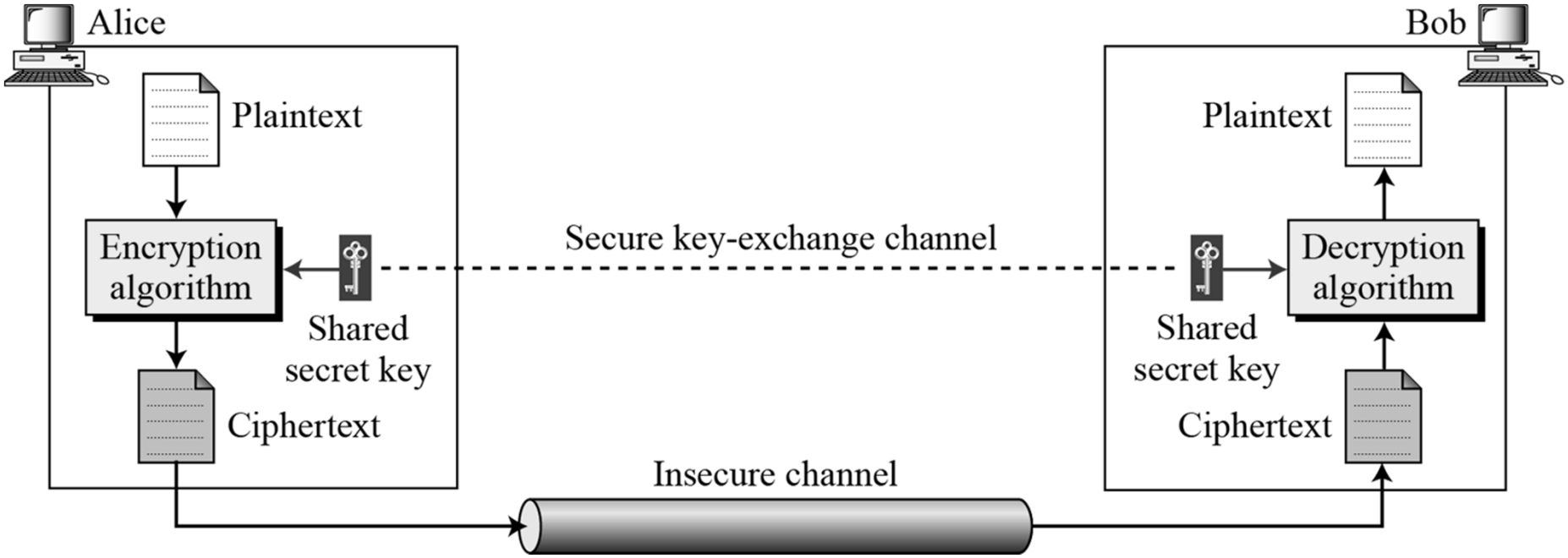
***Figure 3.1 shows the general idea behind a symmetric-key cipher.*** **The original message from Alice to Bob is called plaintext; the message that is sent through the channel is called the ciphertext.** ***To create the ciphertext from the plaintext, Alice uses an encryption algorithm and a shared secret key. To create the plaintext from ciphertext, Bob uses a decryption algorithm and the same secret key.***

***Topics discussed in this section:***

**3.1.1 Kerckhoff's Principle**
**3.1.2 Cryptanalysis**
**3.1.3 Categories of Traditional Ciphers**

# 3.1 Continued

**Figure 3.1** *General idea of symmetric-key cipher*



## *3.1 Continued*

**Traditional symmetric-key ciphers are not used today, but we study them for several reasons:**

**First, they are simpler than modern ciphers and easier to understand.**

**Second, they show the basic foundation of cryptography and encipherment: This foundation can be used to better understand modern ciphers.**

**Third, they provide the rationale for using modern ciphers, because the traditional ciphers can be easily attacked using a computer.**

# 3.1 Continued

***If P is the plaintext, C is the ciphertext, and K is the key,***

$$\text{Encryption: } C = E_k(P) \qquad \text{Decryption: } P = D_k(C)$$

$$\text{In which, } D_k(E_k(x)) = E_k(D_k(x)) = x$$

***We assume that Bob creates*** $\mathbf{P_1}$***; we prove that*** $\mathbf{P_1 = P}$**:**

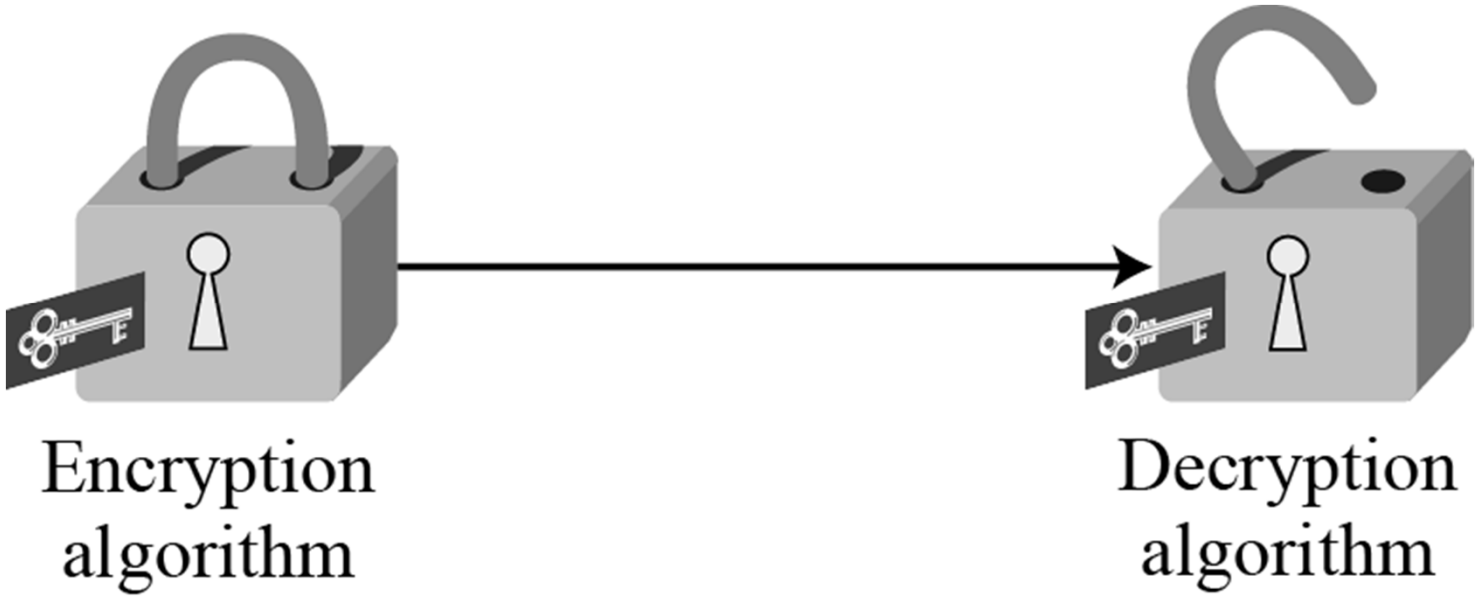
$$\textbf{Alice: } C = E_k(P)$$

$$\textbf{Bob: } P_1 = D_k(C) = D_k(E_k(P)) = P$$

- ***Encryption and decryption algorithms*** as **ciphers**
- A **key** is ***a set of values (numbers) that the cipher, as an algorithm, operates on***.

# 3.1 Continued

**Figure 3.2** *Locking and unlocking with the same key*



## *3.1.1 Kerckhoff's Principle*

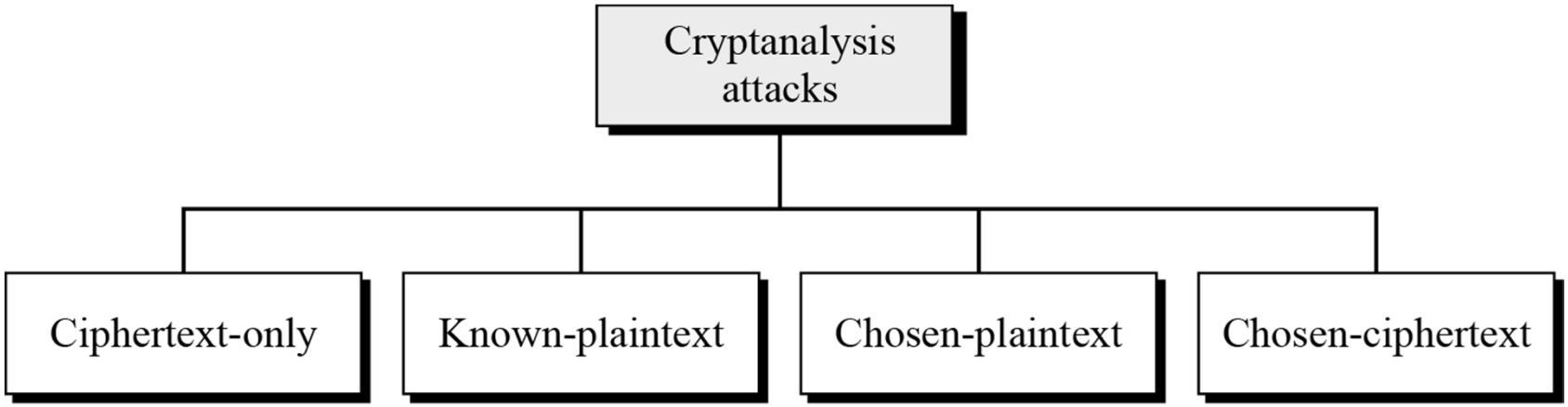
**Based on Kerckhoff's principle, one should always assume that the adversary, Eve, knows the encryption/decryption algorithm. The resistance of the cipher to attack must be based only on the secrecy of the key.**

اصل دوّم از اصول شش‌گانه کرکهف:
جزئیات الگوریتم‌های رمزنگاری بایستی آشکارا و در دید عموم باشد و فقط **کلیدهای رمز سرّی و محرمانه** باشد. به عبارت دیگر، **مقاومت الگوریتم‌های رمزنگاری و رمزگشایی در برابر حمله بایستی بر اساس سرّی و محرمانه بودن کلید باشد.**

## 3.1.2 Cryptanalysis

**As cryptography is the science and art of creating secret codes, cryptanalysis is the science and art of breaking those codes.**

**Figure 3.3** *Cryptanalysis attacks*

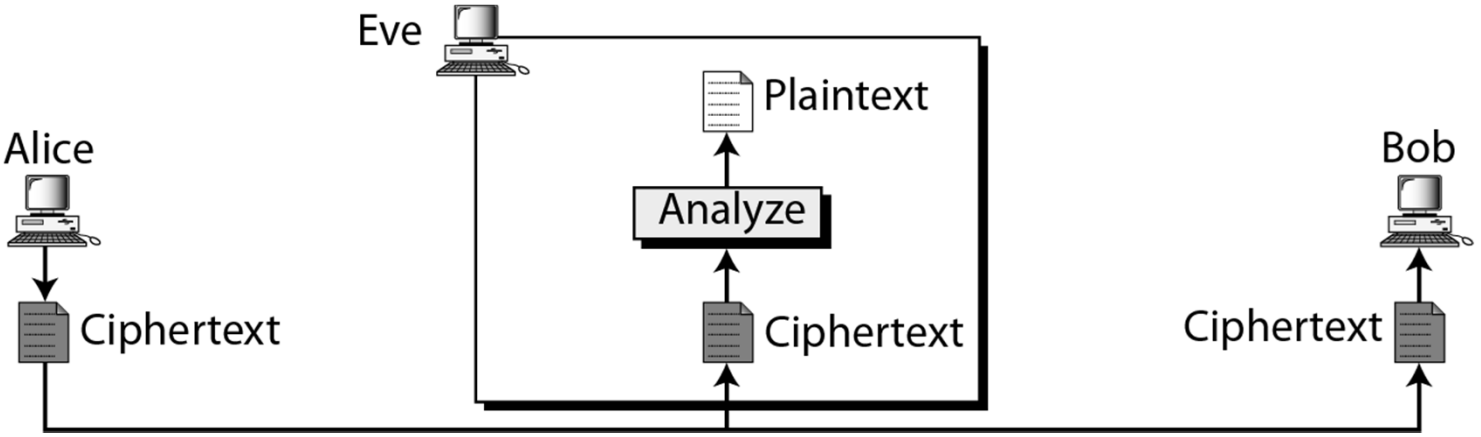


# 3.1.2 Continued

## Ciphertext-Only Attack

Eve has access to only some ciphertext. She tries to find the corresponding key and the plaintext.

**Figure 3.4** ***Ciphertext-only attack***



# 3.1.2 Continued

## Ciphertext-Only Attack

Various methods can be used in ciphertext-only attack. We mention some common ones here:

**Brute-Force Attack**

In the brute-force method or exhaustive-key-search method, Eve tries to use all possible keys.

Using brute-force attack was a difficult task in the past; it is easier today using a computer.

To **prevent** this type of attack, **the number of possible keys must be very large.**

# *3.1.2 Continued*

## Ciphertext-Only Attack

**Statistical Attack**

The cryptanalyst can benefit from some **inherent characteristics of the plaintext language** to launch a statistical attack. or example, we know that the letter E is the most frequently used letter in English text.

To **prevent** this type of attack, **the cipher should hide the characteristics of the language.**

# 3.1.2 Continued

## Ciphertext-Only Attack

### Pattern Attack

Some ciphers may hide the characteristics of the language, but may create some patterns in the ciphertext. A cryptanalyst may use a pattern attack to break the cipher. Therefore, it is **important to use ciphers that make the ciphertext look as random as possible.**

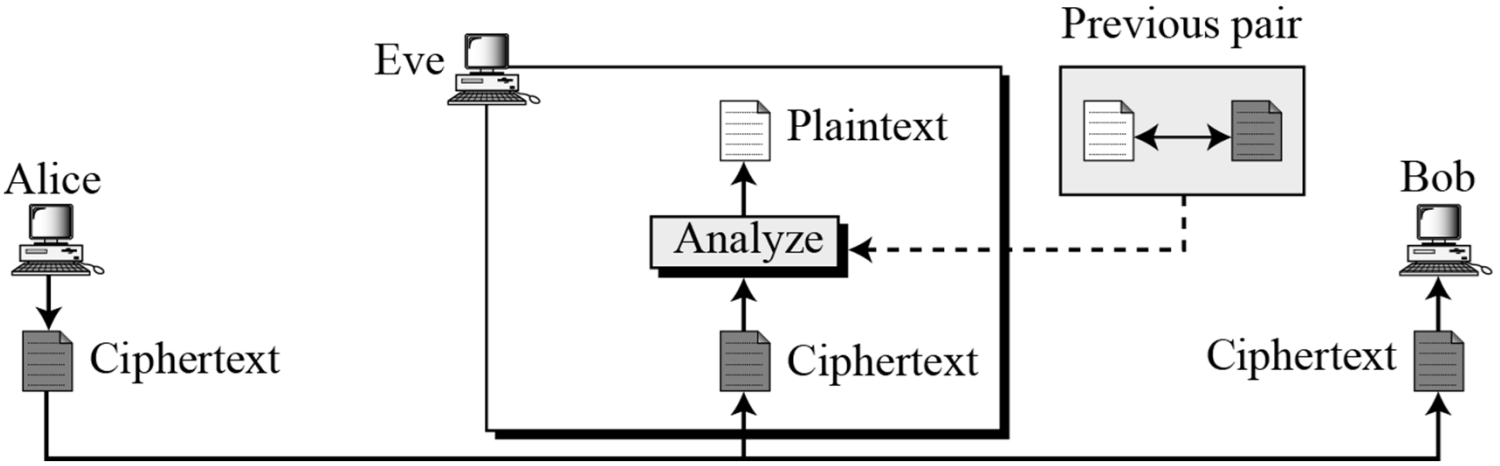
# 3.1.2 Continued

## Known-Plaintext Attack

The plaintext/ciphertext pairs have been collected earlier. For example, Alice has sent a secret message to Bob, but she has later made the contents of the message public.

Eve has kept both the ciphertext and the plaintext to use them to break the next secret message from Alice to Bob, assuming that **Alice has not changed her key**.

**Figure 3.5** ***Known-plaintext attack***



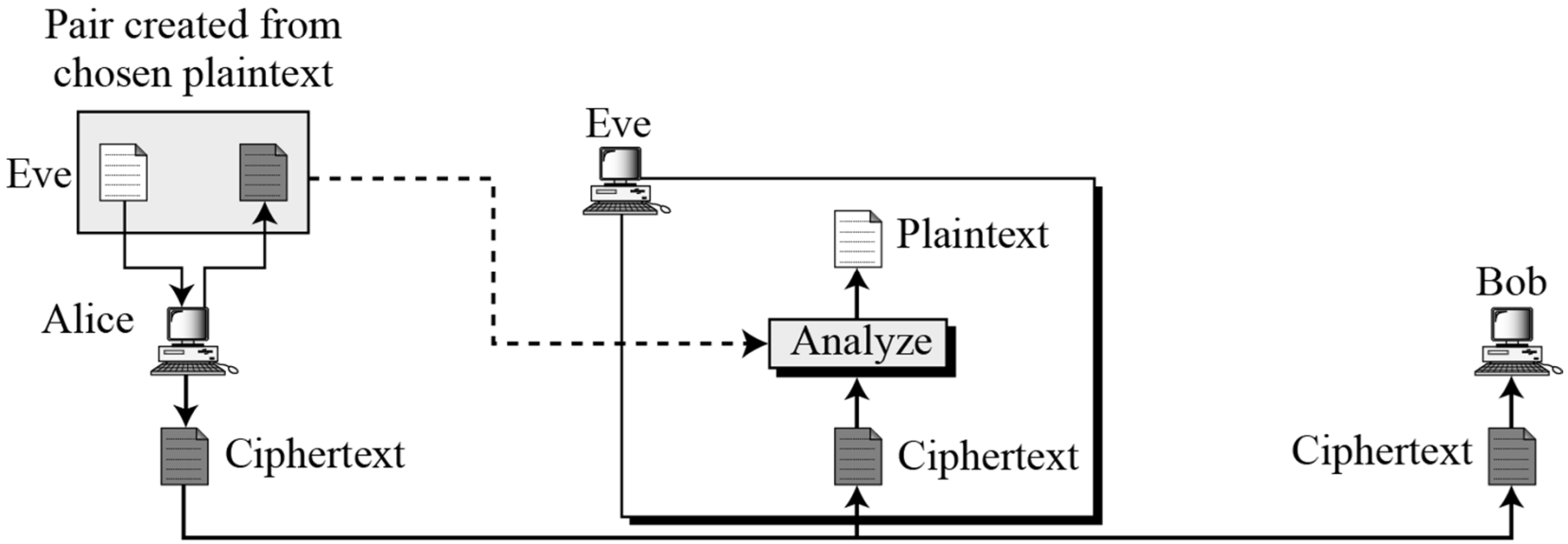
# 3.1.2 Continued

## Chosen-Plaintext Attack

The chosen-plaintext attack is similar to the known-plaintext attack, but the plaintext/ciphertext pairs have been chosen by the attacker herself.

This can happen, for example, if Eve has access to Alice's computer.

**Figure 3.6** ***Chosen-plaintext attack***

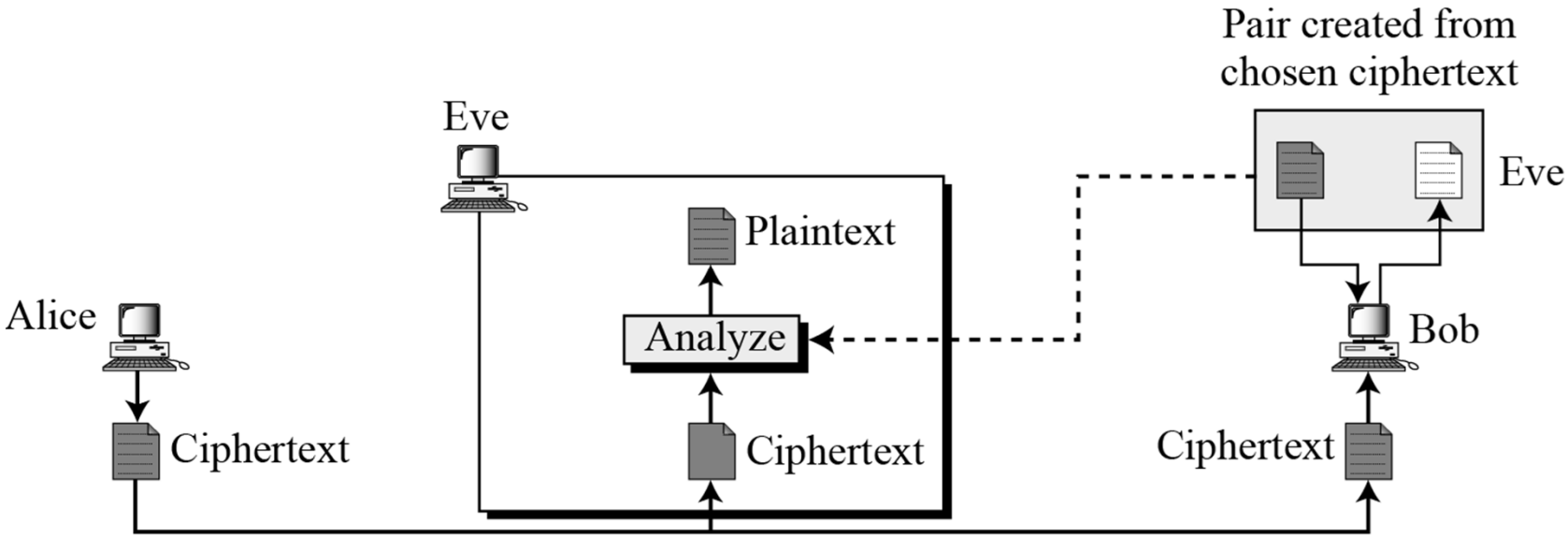


# 3.1.2 Continued

### Chosen-Ciphertext Attack

The chosen-ciphertext attack is similar to the chosen-plaintext attack, except that Eve chooses some ciphertext and decrypts it to form a ciphertext/plaintext pair.

**Figure 3.7** *Chosen-ciphertext attack*



# 3-2 SUBSTITUTION CIPHERS

***A substitution (جانشینی، جانشانی) cipher replaces one symbol with another. Substitution ciphers can be categorized as either monoalphabetic ciphers or polyalphabetic ciphers.***

***Note***

**A substitution cipher replaces one symbol with another.**

***Topics discussed in this section:***

**3.2.1 Monoalphabetic Ciphres**
**3.2.2 Polyalphabetic Ciphers**

# 3.2.1 Monoalphabetic Ciphers

*Note*

**In monoalphabetic substitution, a character (or a symbol) in the plaintext is always changed to the same character (or symbol) in the ciphertext regardless of its position in the text.**

**In monoalphabetic substitution, the relationship between a symbol in the plaintext to a symbol in the ciphertext is always one-to-one. For example, if the algorithm says that letter A in the plaintext is changed to letter D, every letter A is changed to letter D.**

# 3.2.1 Continued

**Example 3.1**

**The following shows a plaintext and its corresponding ciphertext. The cipher is probably monoalphabetic because both *l*'s (*els*) are encrypted as *O*'s.**

**Plaintext:** hello **Ciphertext:** KHOOR

**Example 3.2**

**The following shows a plaintext and its corresponding ciphertext. The cipher is not monoalphabetic because each *l* (*el*) is encrypted by a different character.**

**Plaintext:** hello **Ciphertext:** ABNZF

# 3.2.1 Continued

## Additive Cipher

**The *simplest* monoalphabetic cipher is the additive cipher. This cipher is sometimes called a shift cipher and sometimes a Caesar cipher, but the term additive cipher better reveals its mathematical nature.**

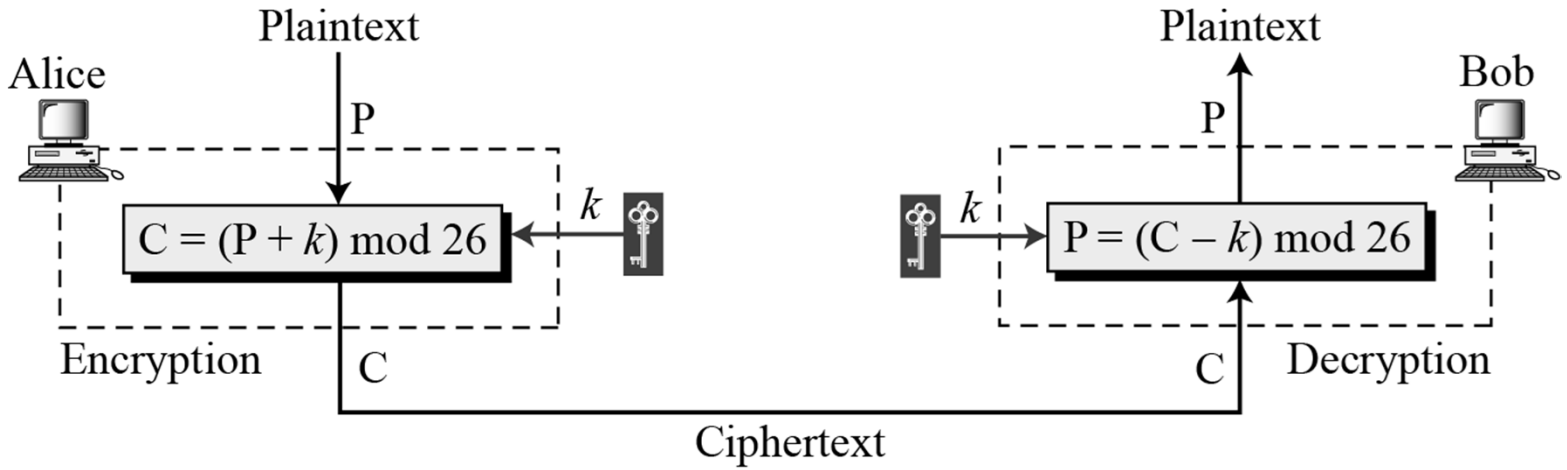
**Additive Cipher = Shift Cipher = Caesar Cipher**

**Figure 3.8** ***Plaintext and ciphertext in $Z_{26}$***

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Plaintext → | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| Ciphertext → | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| Value → | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

# 3.2.1 Continued

**Figure 3.9** *Additive cipher*




***Note***

**When the cipher is additive, the plaintext, ciphertext, and key are integers in $Z_{26}$.**

# 3.2.1 Continued

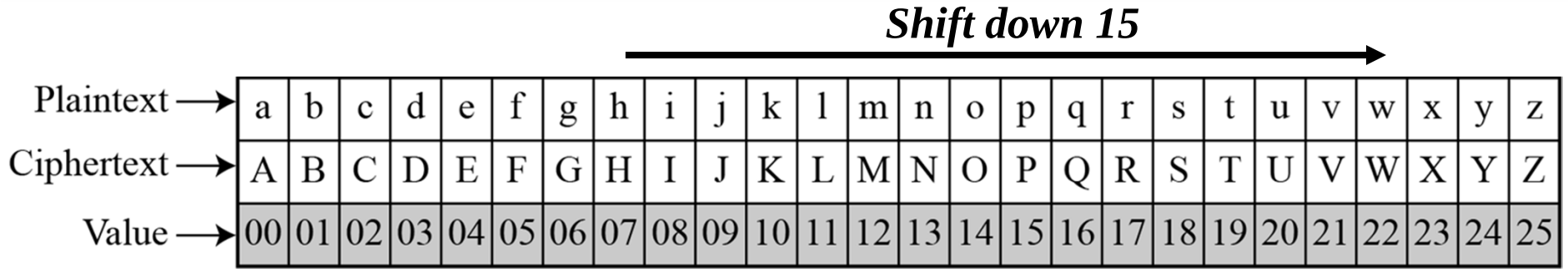
## Example 3.3

**Use the additive cipher with key = 15 to encrypt the message "hello".**

## Solution

**We apply the encryption algorithm to the plaintext, character by character:**

| | | |
|---|---|---|
| Plaintext: h → 07 | Encryption: (07 + 15) mod 26 | Ciphertext: 22 → W |
| Plaintext: e → 04 | Encryption: (04 + 15) mod 26 | Ciphertext: 19 → T |
| Plaintext: l → 11 | Encryption: (11 + 15) mod 26 | Ciphertext: 00 → A |
| Plaintext: l → 11 | Encryption: (11 + 15) mod 26 | Ciphertext: 00 → A |
| Plaintext: o → 14 | Encryption: (14 + 15) mod 26 | Ciphertext: 03 → D |



# 3.2.1 Continued

## Example 3.4

**Use the additive cipher with key = 15 to decrypt the message "WTAAD".**

## Solution

**We apply the decryption algorithm to the plaintext character by character:**

| | | |
|---|---|---|
| Ciphertext: W → 22 | Decryption: (22 – 15) mod 26 | Plaintext: 07 → h |
| Ciphertext: T → 19 | Decryption: (19 – 15) mod 26 | Plaintext: 04 → e |
| Ciphertext: A → 00 | Decryption: (00 – 15) mod 26 | Plaintext: 11 → l |
| Ciphertext: A → 00 | Decryption: (00 – 15) mod 26 | Plaintext: 11 → l |
| Ciphertext: D → 03 | Decryption: (03 – 15) mod 26 | Plaintext: 14 → o |

***Shift up 15***

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Plaintext → | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| Ciphertext → | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| Value → | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

## 3.2.1 Continued

### Shift Cipher and Caesar Cipher

**Historically, additive ciphers are called shift ciphers. Julius Caesar used an additive cipher to communicate with his officers. For this reason, additive ciphers are sometimes referred to as the Caesar cipher. Caesar used a key of 3 for his communications.**

*Note*

**Additive ciphers are sometimes referred to as shift ciphers or Caesar cipher.**

# 3.2.1 Continued

## Shift Cipher and Caesar Cipher

**The reason is that the encryption algorithm can be interpreted as "shift key characters down" and the decryption algorithm can be interpreted as "shift key character up". (Go To Slides 22 & 23) For example, if the key = 15, the encryption algorithm shifts 15 characters down (toward the end of the alphabet). The decryption algorithm shifts 15 characters up (toward the beginning of the alphabet). Of course, when we reach the end or the beginning of the alphabet, we wrap around (manifestation of modulo 26).**

**Additive ciphers are vulnerable to ciphertext-only attacks using exhaustive key searches method (brute-force attacks). The key domain of the additive cipher is very small; there are only 26 keys.**

# 3.2.1 Continued

## Example 3.5

**Eve has intercepted the ciphertext "UVACLYFZLJBYL". Show how she can use a brute-force attack to break the cipher.**

## Solution

**Eve tries keys from 1 to 7. With a key of 7, the plaintext is "not very secure", which makes sense.**

**Ciphertext:** UVACLYFZLJBYL

| | | |
|---|---|---|
| **K = 1** | → | **Plaintext:** tuzbkxeykiaxk |
| **K = 2** | → | **Plaintext:** styajwdxjhzwj |
| **K = 3** | → | **Plaintext:** rsxzivcwigyvi |
| **K = 4** | → | **Plaintext:** qrwyhubvhfxuh |
| **K = 5** | → | **Plaintext:** pqvxgtaugewtg |
| **K = 6** | → | **Plaintext:** opuwfsztfdvsf |
| **K = 7** | → | **Plaintext:** notverysecure |

# 3.2.1 Continued

## Shift Cipher and Caesar Cipher

**Additive ciphers are also subject to statistical attacks. This is especially true if the adversary has a long ciphertext. The adversary can use the frequency of occurrence of characters for a particular language.**

# 3.2.1 Continued

**Table 3.1** *Frequency of characters in English*

| *Letter* | *Frequency* | *Letter* | *Frequency* | *Letter* | *Frequency* | *Letter* | *Frequency* |
|---|---|---|---|---|---|---|---|
| E | 12.7 | H | 6.1 | W | 2.3 | K | 0.08 |
| T | 9.1 | R | 6.0 | F | 2.2 | J | 0.02 |
| A | 8.2 | D | 4.3 | G | 2.0 | Q | 0.01 |
| O | 7.5 | L | 4.0 | Y | 2.0 | X | 0.01 |
| I | 7.0 | C | 2.8 | P | 1.9 | Z | 0.01 |
| N | 6.7 | U | 2.8 | B | 1.5 | | |
| S | 6.3 | M | 2.4 | V | 1.0 | | |

**Table 3.2** *Frequency of diagrams and trigrams*

| | |
|---|---|
| Digram | TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, OF |
| Trigram | THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, DTH |

**http://practicalcryptography.com/**

# 3.2.1 Continued

## Example 3.6

**Eve has intercepted the following ciphertext. Using a statistical attack, find the plaintext.**

XLILSYWIMWRSAJSVWEPIJSVJSYVQMPPMSRHSPPEVWMXMWASVX-LQSVILY-
VVCFIJSVIXLIWIPPIVVIGIMZIWQSVISJJIVW

## Solution

**When Eve tabulates the frequency of letters in this ciphertext, she gets: I =14, V =13, S =12, and so on. The most common character is I with 14 occurrences. This means key = 4 (e → I).**

the house is now for sale for four million dollars it is worth more hurry before the seller receives more offers

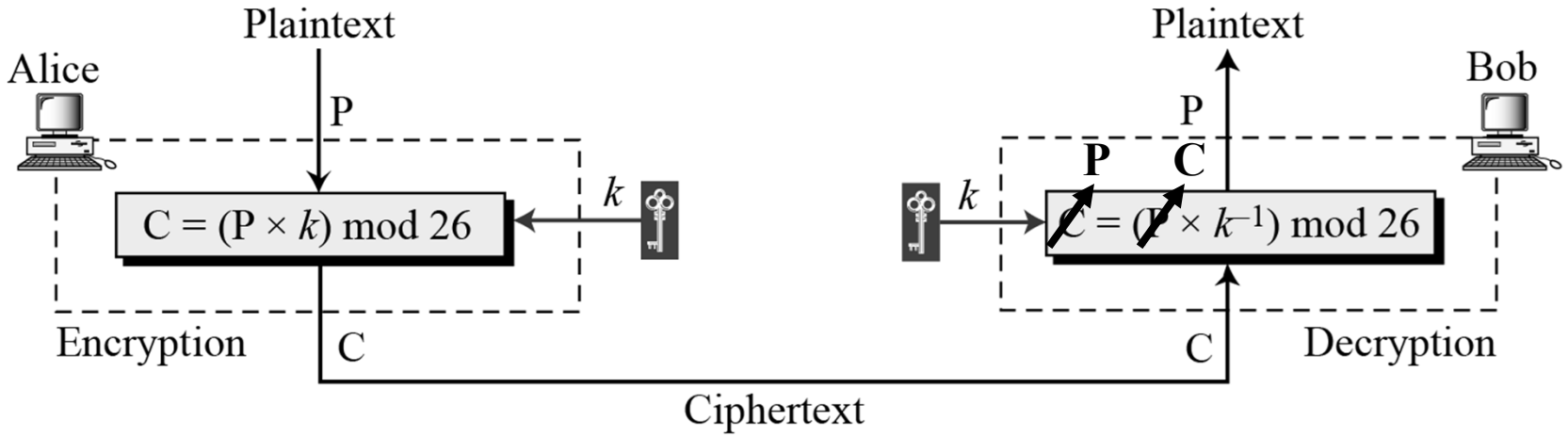
| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Plaintext → | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| Ciphertext → | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| Value → | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

**Key = 4**

# 3.2.1 Continued

## Multiplicative Ciphers

**Figure 3.10** *Multiplicative cipher*




*Note*

**In a multiplicative cipher, the plaintext and ciphertext are integers in $Z_{26}$; the key is an integer in $Z_{26}$*.**

# 3.2.1 Continued

**Example 3.7**

**What is the key domain for any multiplicative cipher?**

## Solution

**The key needs to be in $Z_{26}$*. This set has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.**

**Example 3.8**

**We use a multiplicative cipher to encrypt the message "hello" with a key of 7. The ciphertext is "XCZZU".**

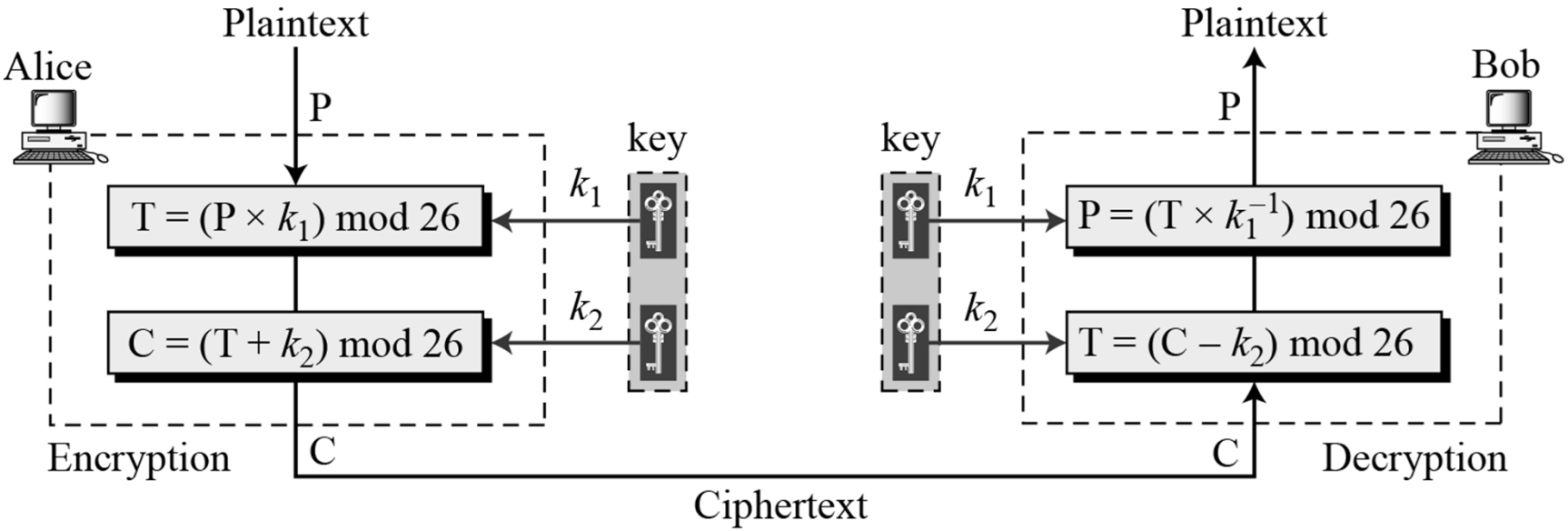
| | | |
|---|---|---|
| Plaintext: h $\rightarrow$ 07 | Encryption: $(07 \times 07) \bmod 26$ | ciphertext: 23 $\rightarrow$ X |
| Plaintext: e $\rightarrow$ 04 | Encryption: $(04 \times 07) \bmod 26$ | ciphertext: 02 $\rightarrow$ C |
| Plaintext: l $\rightarrow$ 11 | Encryption: $(11 \times 07) \bmod 26$ | ciphertext: 25 $\rightarrow$ Z |
| Plaintext: l $\rightarrow$ 11 | Encryption: $(11 \times 07) \bmod 26$ | ciphertext: 25 $\rightarrow$ Z |
| Plaintext: o $\rightarrow$ 14 | Encryption: $(14 \times 07) \bmod 26$ | ciphertext: 20 $\rightarrow$ U |

# 3.2.1 Continued

## Affine Ciphers

**We can combine the additive and multiplicative ciphers to get what is called the affine cipher -a combination of both ciphers with a pair of keys.**

**Figure 3.11** *Affine cipher*




$$C = (P \times k_1 + k_2) \bmod 26 \qquad P = ((C - k_2) \times k_1^{-1}) \bmod 26$$

where $k_1^{-1}$ is the multiplicative inverse of $k_1$ and $-k_2$ is the additive inverse of $k_2$

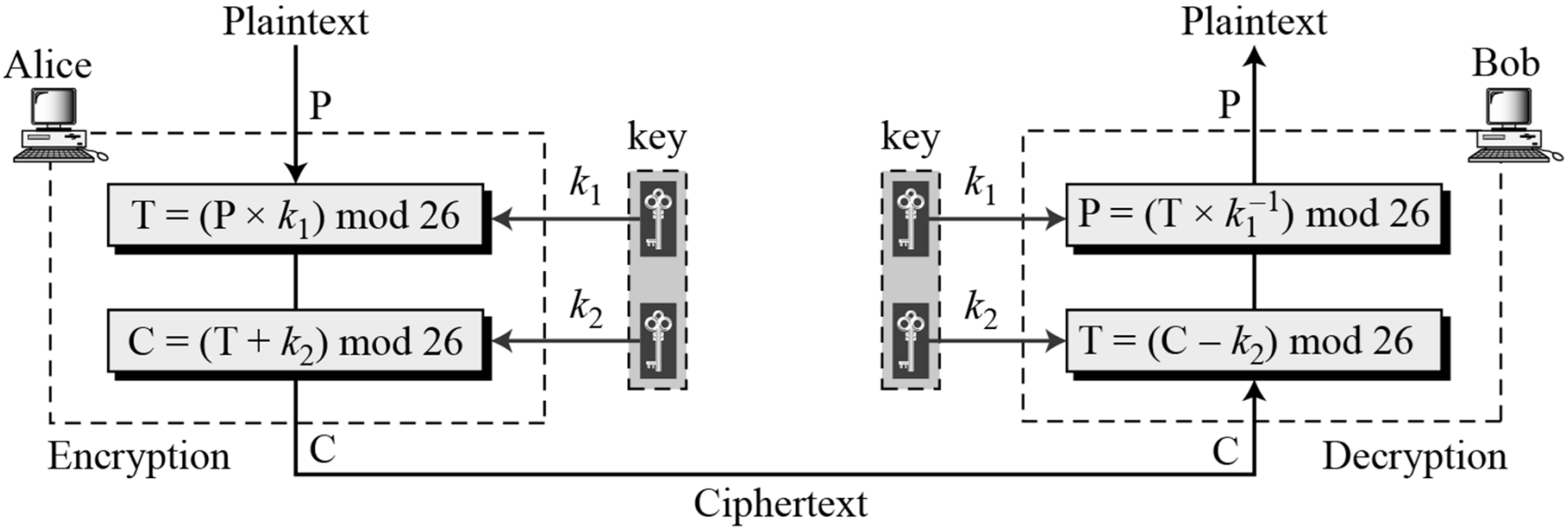
# 3.2.1 Continued

## Affine Ciphers

**Cryptanalysis**

**Ciphertext-only attack (brute-force and statistical method) and chosen-plaintext attack.**

**Figure 3.11** *Affine cipher*



# 3.2.1 Continued

**Example 3.09**

**The affine cipher uses a pair of keys in which the *first key* is from $Z_{26}^*$ and the *second key* is from $Z_{26}$. The size of the key domain is $26 \times 12 = 312$.**

**Example 3.10**

**Use an affine cipher to encrypt the message "hello" with the key pair (7, 2).**

| | | |
|---|---|---|
| P: h $\rightarrow$ 07 | Encryption: $(07 \times 7 + 2) \bmod 26$ | C: 25 $\rightarrow$ Z |
| P: e $\rightarrow$ 04 | Encryption: $(04 \times 7 + 2) \bmod 26$ | C: 04 $\rightarrow$ E |
| P: l $\rightarrow$ 11 | Encryption: $(11 \times 7 + 2) \bmod 26$ | C: 01 $\rightarrow$ B |
| P: l $\rightarrow$ 11 | Encryption: $(11 \times 7 + 2) \bmod 26$ | C: 01 $\rightarrow$ B |
| P: o $\rightarrow$ 14 | Encryption: $(14 \times 7 + 2) \bmod 26$ | C: 22 $\rightarrow$ W |

# 3.2.1 Continued

**Example 3.11**

**Use the affine cipher to decrypt the message "ZEBBW" with the key pair (7, 2) in modulus 26. ($7^{-1} = 15$)**

**Solution**

| | | |
|---|---|---|
| C: Z → 25 | Decryption: $((25 - 2) \times 7^{-1}) \bmod 26$ | P:07 → h |
| C: E → 04 | Decryption: $((04 - 2) \times 7^{-1}) \bmod 26$ | P:04 → e |
| C: B → 01 | Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$ | P:11 → l |
| C: B → 01 | Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$ | P:11 → l |
| C: W → 22 | Decryption: $((22 - 2) \times 7^{-1}) \bmod 26$ | P:14 → o |

**Example 3.12**

**The additive cipher is a special case of an affine cipher in which $k_1 = 1$. The multiplicative cipher is a special case of affine cipher in which $k_2 = 0$.**

# 3.2.1 Continued

## Monoalphabetic Substitution Cipher

**Because additive, multiplicative, and affine ciphers have small key domains, they are very vulnerable to brute-force attack.**

**A better solution is to create a mapping between each plaintext character and the corresponding ciphertext character. Alice and Bob can agree on a table showing the mapping for each character.**

**Figure 3.12** *An example key for monoalphabetic substitution cipher*

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Plaintext → | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| Ciphertext → | N | O | A | T | R | B | E | C | F | U | X | D | Q | G | Y | L | K | H | V | I | J | M | P | Z | S | W |

# 3.2.1 Continued

## Monoalphabetic Substitution Cipher

*Cryptanalysis*

**The size of the key space for the monoalphabetic substitution cipher is 26! (almost $4 \times 10^{26}$). This makes a brute-force attack extremely difficult for Eve even if she is using a powerful computer. However, she can use statistical attack based on the frequency of characters. *The cipher does not change the frequency of characters*.**

**Figure 3.12** *An example key for monoalphabetic substitution cipher*

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Plaintext → | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| Ciphertext → | N | O | A | T | R | B | E | C | F | U | X | D | Q | G | Y | L | K | H | V | I | J | M | P | Z | S | W |

# 3.2.1 Continued

**Example 3.13**

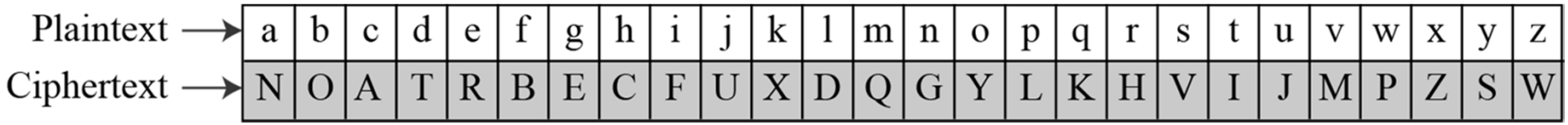
**We can use the key in Figure 3.12 to encrypt the message**

this message is easy to encrypt but hard to find the key

**The ciphertext is**

ICFVQRVVNEFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

**Figure 3.12** ***An example key for monoalphabetic substitution cipher***



| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Plaintext → | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| Ciphertext → | N | O | A | T | R | B | E | C | F | U | X | D | Q | G | Y | L | K | H | V | I | J | M | P | Z | S | W |

# 3.2.2 Polyalphabetic Ciphers

**In polyalphabetic substitution, each occurrence of a character may have a different substitute. The relationship between a character in the plaintext to a character in the ciphertext is one-to-many.**

**For example, "a" could be enciphered as "D" in the beginning of the text, but as "N" at the middle.**

**Polyalphabetic ciphers have the advantage of hiding the letter frequency of the underlying language.**

## Autokey Cipher

$$P = P_1P_2P_3 \ldots \qquad C = C_1C_2C_3\ldots \qquad k = (k_1, P_1, P_2, \ldots)$$

Encryption: $C_i = (P_i + k_i) \bmod 26$ Decryption: $P_i = (C_i - k_i) \bmod 26$

# 3.2.2 Polyalphabetic Ciphers

## Autokey Cipher

- **Autokey cipher is a simple polyalphabetic cipher.**
- **In this cipher, the key is a stream of subkeys, in which each subkey is used to encrypt the corresponding character in the plaintext.**
- **The first subkey is a predetermined value secretly agreed upon by Alice and Bob. The second subkey is the value of the first plaintext character (between 0 and 25). The third subkey is the value of the second plaintext. And so on**

### Autokey Cipher

$$P = P_1P_2P_3 \ldots \qquad C = C_1C_2C_3\ldots \qquad k = (k_1, P_1, P_2, \ldots)$$

Encryption: $C_i = (P_i + k_i) \bmod 26$ Decryption: $P_i = (C_i - k_i) \bmod 26$

# 3.2.2 Continued

## Example 3.14

**Assume that Alice and Bob agreed to use an autokey cipher with initial key value $k_1 = 12$. Now Alice wants to send Bob the message "Attack is today". Enciphering is done character by character.**

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Plaintext: | a | t | t | a | c | k | i | s | t | o | d | a | y |
| P's Values: | 00 | 19 | 19 | 00 | 02 | 10 | 08 | 18 | 19 | 14 | 03 | 00 | 24 |
| Key stream: | *12* | *00* | *19* | *19* | *00* | *02* | *10* | *08* | *18* | *19* | *14* | *03* | *00* |
| C's Values: | 12 | 19 | 12 | 19 | 02 | 12 | 18 | 00 | 11 | 7 | 17 | 03 | 24 |
| Ciphertext: | **M** | **T** | **M** | **T** | **C** | **M** | **S** | **A** | **L** | **H** | **R** | **D** | **Y** |

### Cryptanalysis

**The autokey cipher definitely hides the single-letter frequency statistics of the plaintext. However, it is still as vulnerable to the brute-force attack as the additive cipher. The first subkey can be only one of the 25 values (1 to 25).**

# 3.2.2 Continued

## Playfair Cipher

***Another example of a polyalphabetic cipher is the Playfair cipher used by the British army during World War I. The secret key in this cipher is made of 25 alphabet letters arranged in a 5 × 5 matrix (letters I and J are considered the same when encrypting). Different arrangements of the letters in the matrix can create many different secret keys. One of the possible arrangements is shown in Figure 3.13.***

**Figure 3.13** ***An example of a secret key in the Playfair cipher***

Secret Key =

| | | | | |
|---|---|---|---|---|
| L | G | D | B | A |
| Q | M | H | E | C |
| U | R | N | I/J | F |
| X | V | S | O | K |
| Z | Y | W | T | P |

$P = P_1P_2P_3 \ldots$ $C = C_1C_2C_3 \ldots$ $k = [(k_1, k_2), (k_3, k_4), \ldots]$

Encryption: $C_i = k_i$ Decryption: $P_i = k_i$

# 3.2.2 Continued

## Playfair Cipher

**Before encryption:**

**I. if the two letters in a pair are the same, a bogus letter is inserted to separate them.**

**II. After inserting bogus letters, if the number of characters in the plaintext is odd, one extra bogus character is added at the end to make the number of characters even.**

**The cipher uses three rules for encryption:**

**a. If the two letters in a pair are located in the same row of the secret key, the corresponding encrypted character for each letter is the next letter to the right in the same row (with wrapping to the beginning of the row if the plaintext letter is the last character in the row).**

**b. If the two letters in a pair are located in the same column of the secret key, the corresponding encrypted character for each letter is the letter beneath it in the same column (with wrapping to the beginning of the column if the plaintext letter is the last character in the column).**

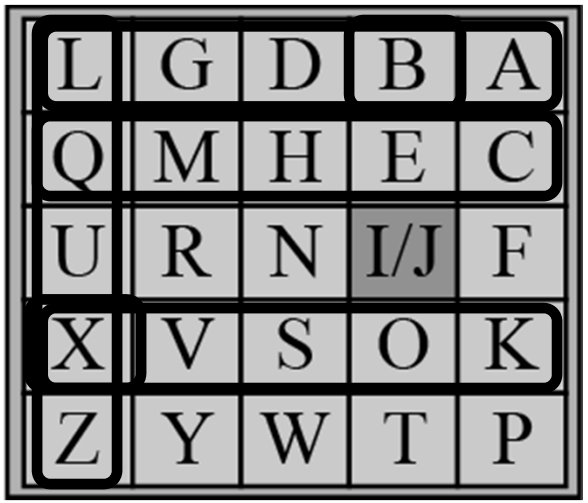
**c. If the two letters in a pair are not in the same row or column of the secret, the corresponding encrypted character for each letter is a letter that is in its own row but in the same column as the other letter.**

# 3.2.2 Continued

## Playfair Cipher

**Figure 3.13** *An example of a secret key in the Playfair cipher*

Secret Key =




**Example 3.15**

**Let us encrypt the plaintext "hello" using the key in Figure 3.13.**

he → EC lx → QZ lo → BX

Plaintext: hello Ciphertext: ECQZBX

# 3.2.2 Continued

## Playfair Cipher

**Cryptanalysis**

**Obviously a brute-force attack on a Playfair cipher is very difficult. The size of the key domain is 25! (factorial 25). In addition, the encipherment hides the single-letter frequency of the characters. However, the frequencies of diagrams are preserved (to some extent because of filler insertion), so a cryptanalyst can use a ciphertext-only attack based on the diagram frequency test to find the key.**

# 3.2.2 Continued

## Vigenere Cipher

**One interesting kind of polyalphabetic cipher was designed by Blaise de Vigenere, a sixteenth-century French mathematician. A Vigenere cipher uses a different strategy to create the key stream. The key stream is a repetition of an initial secret key stream of length m, where we have $1 \leq m \leq 26$.**

$$P = P_1P_2P_3 \ldots \qquad C = C_1C_2C_3 \ldots \qquad K = [(k_1, k_2, \ldots, k_m), (k_1, k_2, \ldots, k_m), \ldots]$$

$$\text{Encryption: } C_i = P_i + k_i \qquad \text{Decryption: } P_i = C_i - k_i$$

# 3.2.2 Continued

## Example 3.16

**Let us see how we can encrypt the message "She is listening" using the 6-character keyword "PASCAL". The initial key stream is (15, 0, 18, 2, 0, 11). The key stream is the repetition of this initial key stream (as many times as needed).**

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Plaintext:** | s | h | e | i | s | l | i | s | t | e | n | i | n | g |
| **P's values:** | 18 | 07 | 04 | 08 | 18 | 11 | 08 | 18 | 19 | 04 | 13 | 08 | 13 | 06 |
| **Key stream:** | ***15*** | ***00*** | ***18*** | ***02*** | ***00*** | ***11*** | ***15*** | ***00*** | ***18*** | ***02*** | ***00*** | ***11*** | ***15*** | ***00*** |
| **C's values:** | **07** | **07** | **22** | **10** | **18** | **22** | **23** | **18** | **11** | **6** | **13** | **19** | **02** | **06** |
| **Ciphertext:** | **H** | **H** | **W** | **K** | **S** | **W** | **X** | **S** | **L** | **G** | **N** | **T** | **C** | **G** |

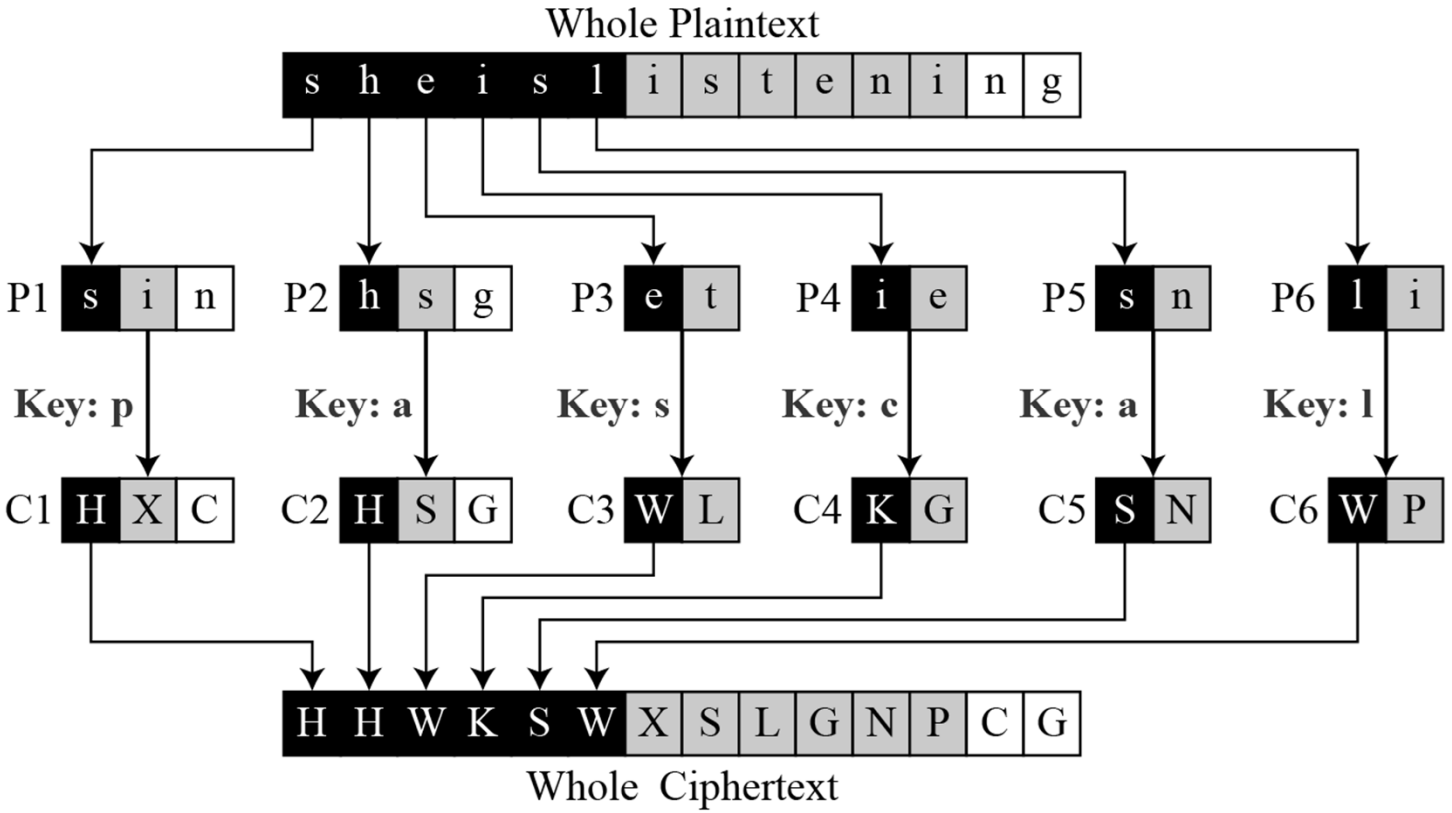
| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Plaintext → | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| Ciphertext → | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| Value → | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

# 3.2.2 Continued

## Example 3.17

**Vigenere cipher can be seen as combinations of m additive ciphers.**

**Figure 3.14** *A Vigenere cipher as a combination of m additive ciphers*



# 3.2.2 Continued

## Example 3.18

**Using Example 3.18, we can say that the additive cipher is a special case of Vigenere cipher in which $m = 1$.**

**Table 3.3**
*A Vigenere Tableau*

| | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | v | v | w | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *A* | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| *B* | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| *C* | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| *D* | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| *E* | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| *F* | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| *G* | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| *H* | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| *I* | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| *J* | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| *K* | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| *L* | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| *M* | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| *N* | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| *O* | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| *P* | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| *Q* | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| *R* | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| *S* | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| *T* | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| *U* | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| *V* | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| *W* | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| *X* | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| *Y* | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| *Z* | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |

# 3.2.2 Continued

## Vigenere Cipher (Crypanalysis)

**Example 3.19**

**Let us assume we have intercepted the following ciphertext:**

```
LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-
VTSGXQOVGCSVETQLTJSUMVWVEUVLXEWSLGFZMVVWLGYHCUSWXQH-
KVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWVWJWIXGFWLTSHGJOUEEHH-
VUCFVGOWICQLTJSUXGLW
```

**The Kasiski test for repetition of three-character segments yields the results shown in Table 3.4.**

| *String* | *First Index* | *Second Index* | *Difference* |
|---|---|---|---|
| JSU | 68 | 168 | 100 |
| SUM | 69 | 117 | 48 |
| VWV | 72 | 132 | 60 |
| MPH | 119 | 127 | 8 |

# 3.2.2 Continued

## Example 3.19

**Let us assume we have intercepted the following ciphertext:**

```
LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-
VTSGXQOVGCSVETQLTJSUMVWVEUVLXEWSLGFZMVVWLGYHCUSWXQH-
KVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWVWJWIXGFWLTSHGJOUEEHH-
VUCFVGOWICQLTJSUXGLW
```

**The Kasiski test for repetition of three-character segments yields the results shown in Table 3.4.**

| *String* | *First Index* | *Second Index* | *Difference* |
|---|---|---|---|
| JSU | 68 | 168 | 100 |
| SUM | 69 | 117 | 48 |
| VWV | 72 | 132 | 60 |
| MPH | 119 | 127 | 8 |

# 3.2.2 Continued

## Example 3.19 (Continued)

**The greatest common divisor of differences is 4, which means that the key length is multiple of 4. First try $m = 4$.**

```
C1: LWGWCRAOKTEPGTQCTJVUEGVGUQGECVPRPVJGTJEUGCJG
P1: jueuapymircneroarhtsthihytrahcieixsthcarrehe
C2: IGGGQHGWGKVCTSOSQSWVWFVYSHSVFSHZHWWFSOHCOQSL
P2: usssctsiswhofeaeceihcetesoecatnpntherhctecex
C3: OFDHURWQZKLZHGVVLUVLSZWHWKHFDUKDHVIWHUHFWLUW
P3: lcaerotnwhiwedssirsiirhketehretltiideatrairt
C4: MEVHCWILEMWVVXGETMEXLMLCXVELGMIMBWXLGEVVITX
P4: iardysehaisrrtcapiafpwtethecarhaesfterectpt
```

**In this case, the plaintext makes sense.**

---

**Julius Caesar used a cryptosystem in his wars, which is now referred to as Caesar cipher. It is an additive cipher with the key set to three. Each character in the plaintext is shifted three characters to create ciphertext.**

---

# 3.2.2 Continued

## Hill Cipher

**Another interesting example of a polyalphabetic cipher is the Hill cipher invented by Lester S. Hill. Unlike the other polyalphabetic ciphers we have already discussed, the plaintext is divided into equal-size blocks. The blocks are encrypted one at a time in such a way that each character in the block contributes to the encryption of other characters in the block.**

$$K = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1m} \\ k_{21} & k_{22} & \dots & k_{2m} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \dots & k_{mm} \end{bmatrix}$$

$$C_1 = P_1 k_{11} + P_2 k_{21} + \cdots + P_m k_{m1}$$
$$C_2 = P_1 k_{12} + P_2 k_{22} + \cdots + P_m k_{m2}$$
$$\cdots$$
$$C_m = P_1 k_{1m} + P_2 k_{2m} + \cdots + P_m k_{mm}$$

**The equations show that each ciphertext character such as C1 depends on all plaintext characters in the block (P1, P2, . . ., Pm). However, we should be aware that not all square matrices have multiplicative inverses in $Z_{26}$, so Alice and Bob should be careful in selecting the key.**

# 3.2.2 Continued

## Hill Cipher

**Figure 3.15** *Key in the Hill cipher*

$$K = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1m} \\ k_{21} & k_{22} & \cdots & k_{2m} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \cdots & k_{mm} \end{bmatrix}$$

$$C_1 = P_1 k_{11} + P_2 k_{21} + \cdots + P_m k_{m1}$$
$$C_2 = P_1 k_{12} + P_2 k_{22} + \cdots + P_m k_{m2}$$
$$\cdots$$
$$C_m = P_1 k_{1m} + P_2 k_{2m} + \cdots + P_m k_{mm}$$

***Note***

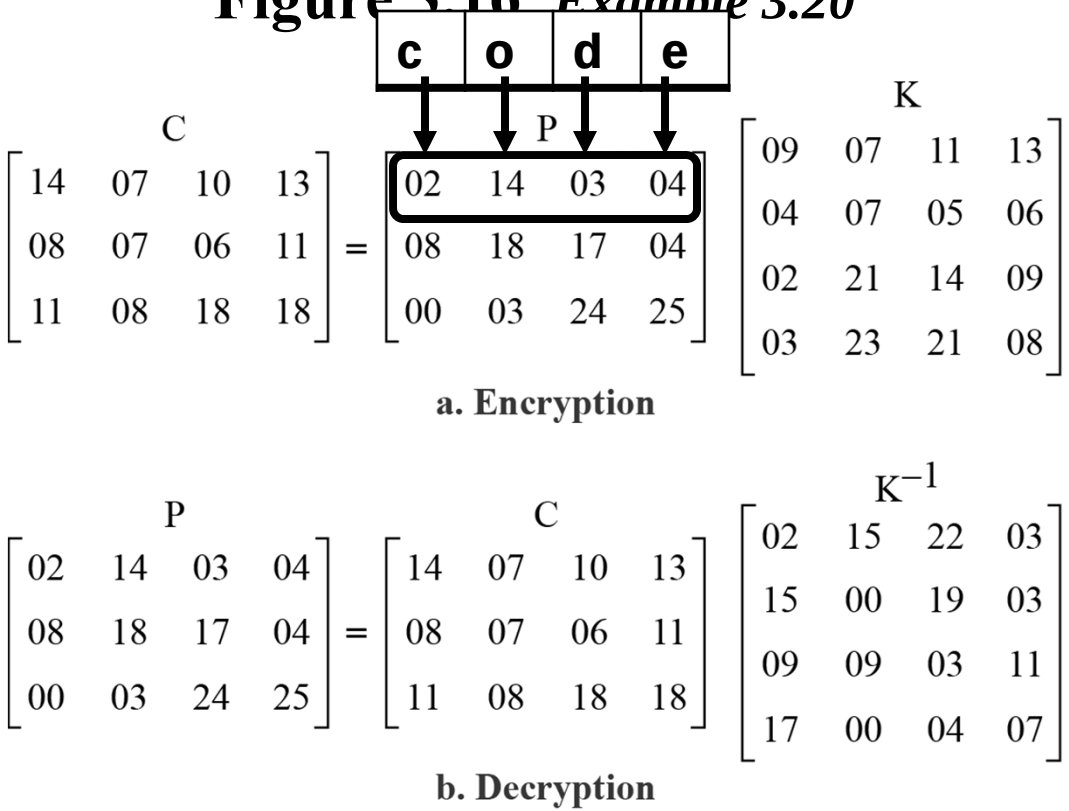
**The key matrix in the Hill cipher needs to have a multiplicative inverse.**

# 3.2.2 Continued

## Example 3.20

**For example, the plaintext "code is ready" can make a 3 × 4 matrix when adding extra bogus character "z" to the last block and removing the spaces. The ciphertext is "OHKNIHGKLISS".**

**Figure 3.16** *Example 3.20*

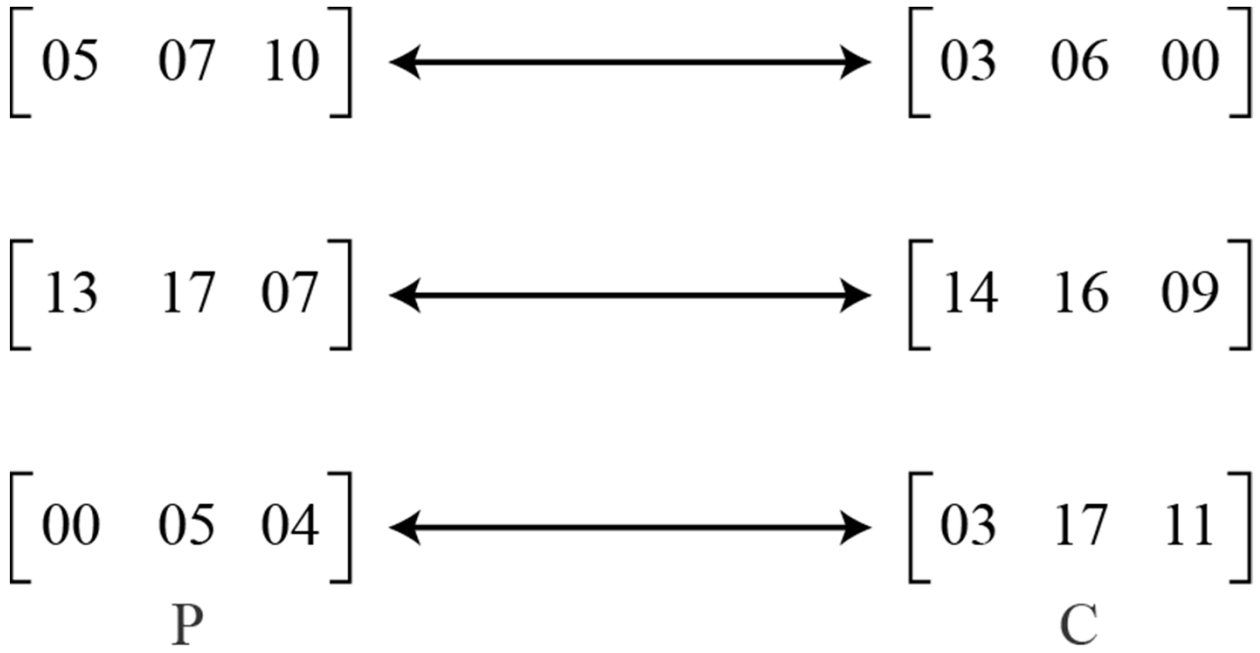


# 3.2.2 Continued

## Example 3.21

**Assume that Eve knows that $m = 3$. She has intercepted three plaintext/ciphertext pair blocks (not necessarily from the same message) as shown in Figure 3.17.**

**Figure 3.17** *Example 3.21*



# 3.2.2 Continued

**Example 3.21 (Continued)**

**She makes matrices P and C from these pairs. Because P is invertible, she inverts the P matrix and multiplies it by C to get the K matrix as shown in Figure 3.18.**

**Figure 3.18** *Example 3.21*

$$\begin{bmatrix} 02 & 03 & 07 \\ 05 & 07 & 09 \\ 01 & 02 & 11 \end{bmatrix} = \begin{bmatrix} 21 & 14 & 01 \\ 00 & 08 & 25 \\ 13 & 03 & 08 \end{bmatrix} \begin{bmatrix} 03 & 06 & 00 \\ 14 & 16 & 09 \\ 03 & 17 & 11 \end{bmatrix}$$

$$K \qquad P^{-1} \qquad C$$

**Now she has the key and can break any ciphertext encrypted with that key.**

# 3.2.2 Continued

## One-Time Pad

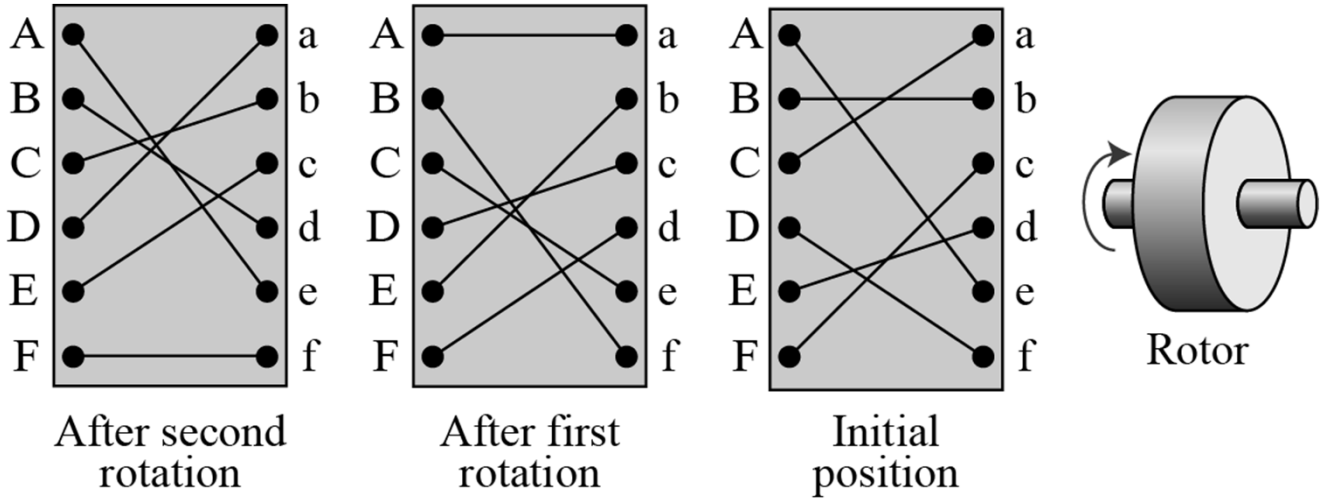
**One of the goals of cryptography is perfect secrecy. A study by Shannon has shown that perfect secrecy can be achieved if each plaintext symbol is encrypted with a key randomly chosen from a key domain. This idea is used in a cipher called one-time pad, invented by Vernam.**

**A one-time pad is a perfect cipher, but it is almost impossible to implement commercially. If the key must be newly generated each time, how can Alice tell Bob the new key each time she has a message to send?**

# 3.2.2 Continued

## Rotor Cipher

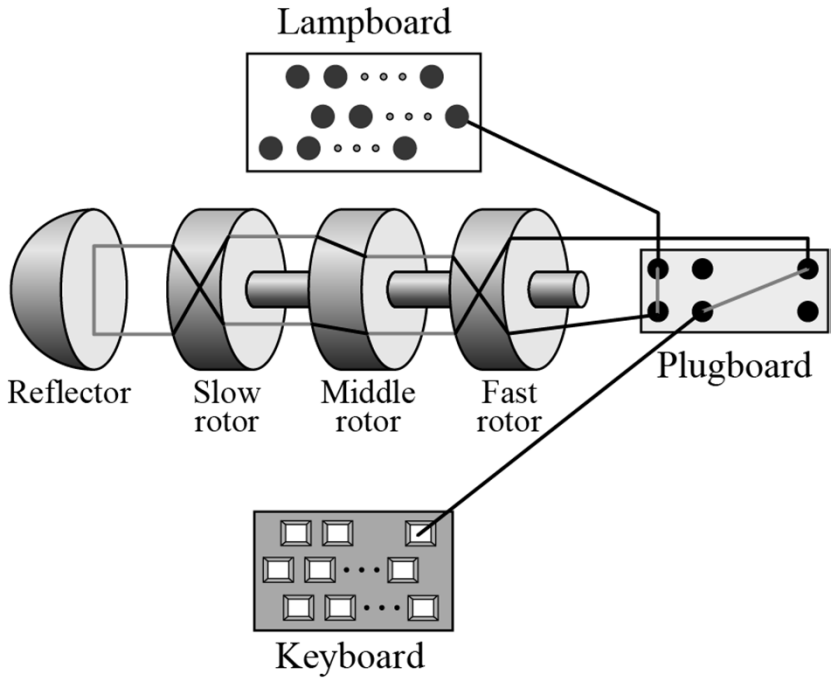
**Figure 3.19** *A rotor cipher*



# 3.2.2 Continued

## Enigma Machine

**Figure 3.20** *A schematic of the Enigma machine*



# 3-3 TRANSPOSITION CIPHERS

**A transposition (جابجایی) cipher does not substitute one symbol for another, instead it changes the location of the symbols.**

**A symbol in the first position of the plaintext may appear in the tenth position of the ciphertext.**

*Note*

**A transposition cipher reorders symbols.**

***Topics discussed in this section:***

**3.3.1 Keyless Transposition Ciphers**
**3.3.2 Keyed Transposition Ciphers**
**3.3.3 Combining Two Approaches**

## *3.3.1 Keyless Transposition Ciphers*

**Simple transposition ciphers, which were used in the past, are keyless.**

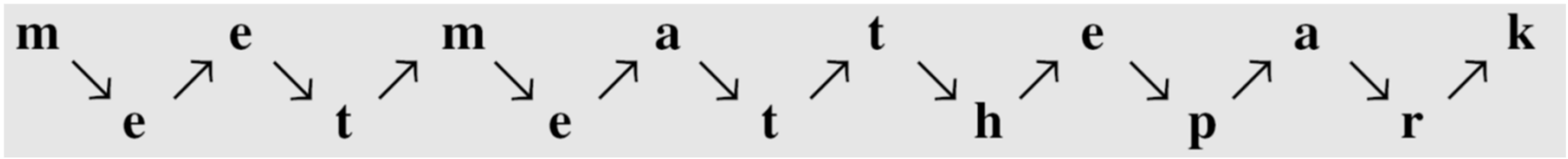
**There are two methods for permutation of characters:**

- ✓ In the **first method**, the text is written into a table column by column and then transmitted row by row.
- ✓ In the **second method**, the text is written into the table row by row and then transmitted column by column.

# 3.3.1 Keyless Transposition Ciphers

## Example 3.22

**A good example of a keyless cipher using the first method is the rail fence cipher. In this cipher, the plaintext is arranged in two lines as a zigzag pattern (which means column by column);The ciphertext is created reading the pattern row by row. For example, to send the message "Meet me at the park" to Bob, Alice writes**




**She then creates the ciphertext "MEMATEAKETETHPR".**

**ستون به ستون plaintext را می‌نویسیم و سپس سطر به سطر ارسال می‌کنیم.**

# 3.3.1 Continued

## Example 3.23

**Alice and Bob can agree on the number of columns and use the second method. Alice writes the same plaintext, row by row, in a table of four columns.**

| | | | |
|---|---|---|---|
| m | e | e | t |
| m | e | a | t |
| t | h | e | p |
| a | r | k | |

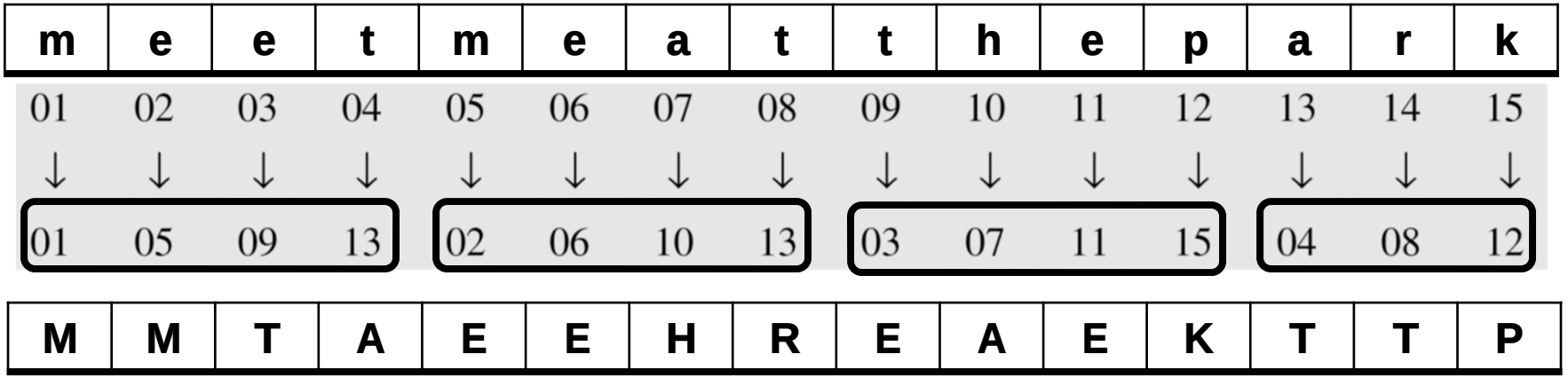
**She then creates the ciphertext "MMTAEEHREAEKTTP".**

**ابتدا تعداد ستون مورد توافق بین alice و bob قرار می‌گیرد سپس alice، plaintext را به صورت سطر به سطر نوشته و به صورت ستون به ستون ارسال می‌کند.**

# 3.3.1 Continued

## Example 3.24

**The cipher in Example 3.23 is actually a transposition cipher. The following shows the permutation of each character in the plaintext into the ciphertext based on the positions.**




**The second character in the plaintext has moved to the fifth position in the ciphertext; the third character has moved to the ninth position; and so on. Although the characters are permuted, there is a pattern in the permutation: (01, 05, 09, 13), (02, 06, 10, 14), (03, 07, 11, 15), and (04, 08, 12). In each section, the difference between the two adjacent numbers is 4.**

# 3.3.2 Keyed Transposition Ciphers

**The keyless ciphers permute the characters by using writing plaintext in one way and reading it in another way The permutation is done on the whole plaintext to create the whole ciphertext.**

**الگوریتم رمزنگاری بدون کلید کاراکترها را با استفاده از نوشتن plaintext با یک روش و خواندن آن با روش دیگر جابجا می‌کند. اِعمال جابجایی روی سرتاسر plaintext، ciphertext را ایجاد می‌کند.**

***Another method (keyed transposition ciphers)*** **is to divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately.**

**روش دیگر (الگوریتم رمزنگاری جابجایی همراه کلید) این است که plaintext را به گروه‌هایی با سایزهای از قبل تعیین شده که به آنها بلوک می‌گوییم، تقسیم کنیم و سپس از یک کلید برای جابجایی کاراکترها در هر بلوک استفاده کنیم.**

# 3.3.2 Continued

## Example 3.25

**Alice needs to send the message "Enemy attacks tonight" to Bob..**

e n e m y a t t a c k s t o n i g h t z

**The key used for encryption and decryption is a permutation key, which shows how the character are permuted.**

Encryption ↓

| 3 | 1 | 4 | 5 | 2 |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

↑ Decryption

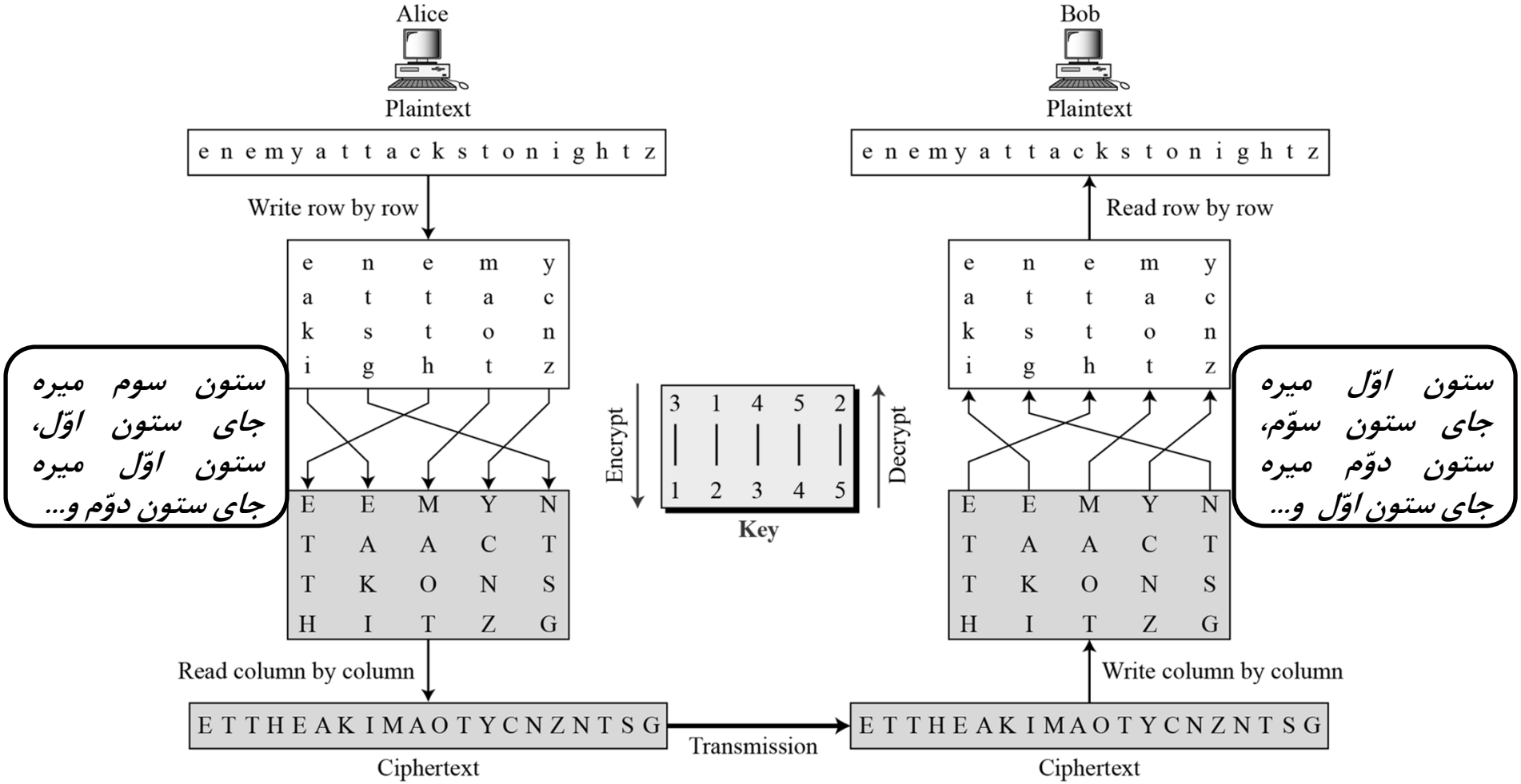
**The permutation yields**

E E M Y N T A A C T T K O N S H I T Z G

# 3.3.3 Combining Two Approaches

**Example 3.26**

**Figure 3.21**



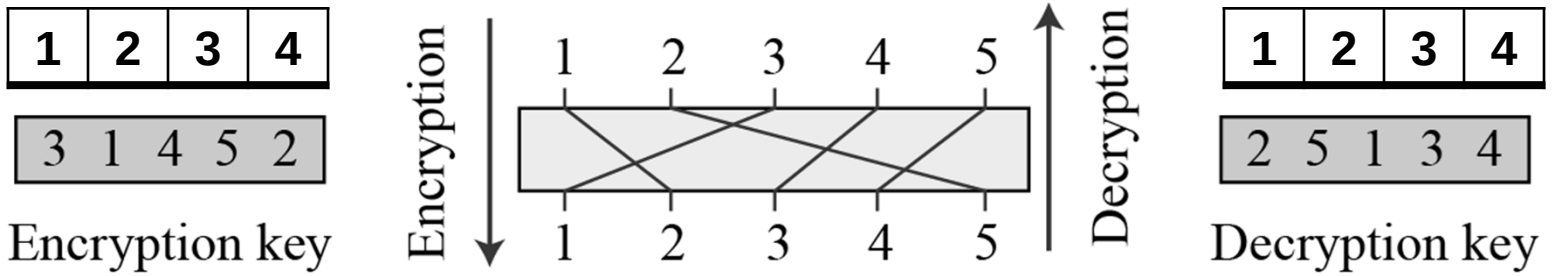
# 3.3.3 Continued

## Keys

**In Example 3.27, a single key was used in two directions for the column exchange: downward for encryption, upward for decryption. It is customary to create two keys.**

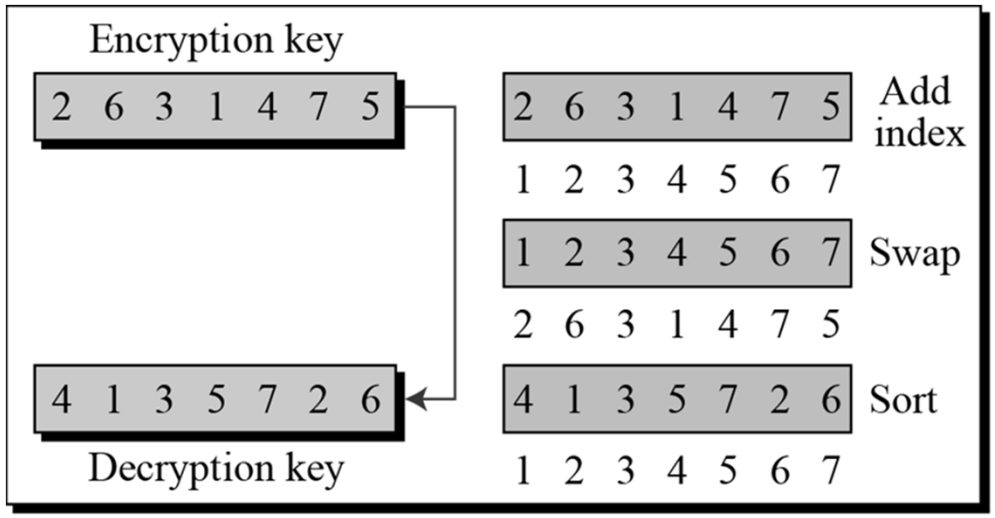
**در مثال قبل ما از یک کلید برای هر دو طرف استفاده می‌کردیم. فلش رو به پایین برای رمزنگاری، فلش رو به بالا برای رمزگشایی. اما روش مرسوم این است که از دو کلید استفاده کنیم یکی برای رمزنگاری و یکی برای رمزگشایی.**

**Figure 3.22** *Encryption/decryption keys in transpositional ciphers*



# 3.3.3 Continued

**Figure 3.23** *Key inversion in a transposition cipher*



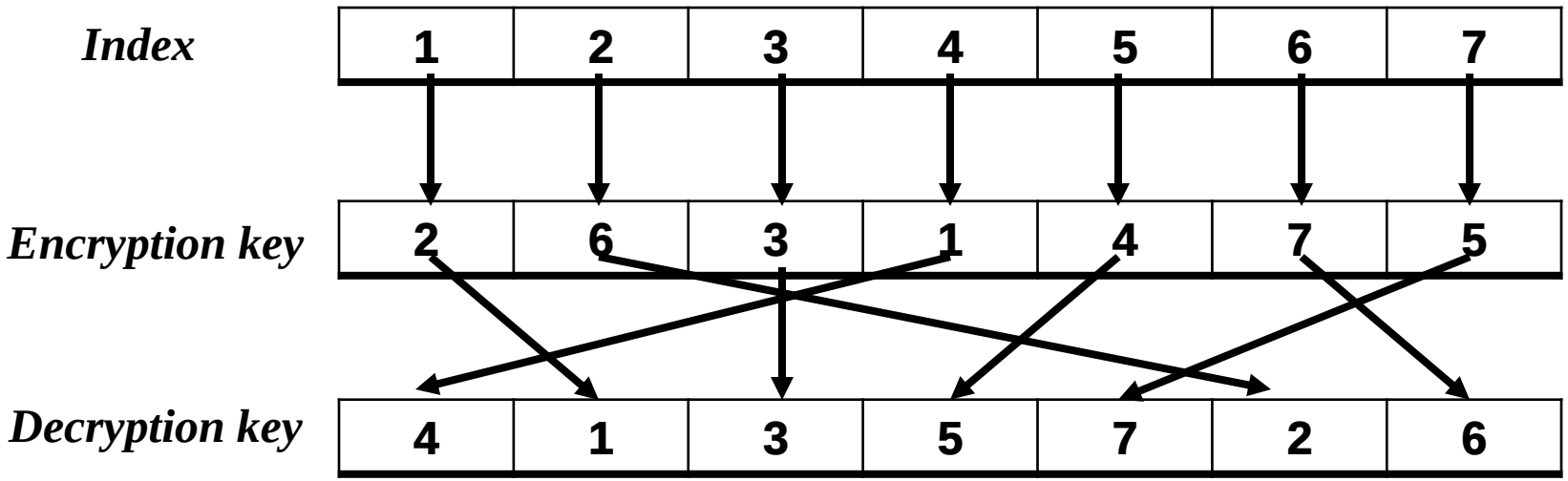

a. Manual process

```
Given: EncKey [index]
index ← 1
while (index ≤ Column)
  {
   DecKey[EncKey[index]] ← index
   index ← index + 1
  }
Return : DecKey [index]
```

b. Algorithm

# 3.3.3 Continued

**Figure 3.23** *Key inversion in a transposition cipher*




**عدد ۲ در index شماره‌ی ۱ است پس عدد ۱ میره به index شماره‌ی ۲.**

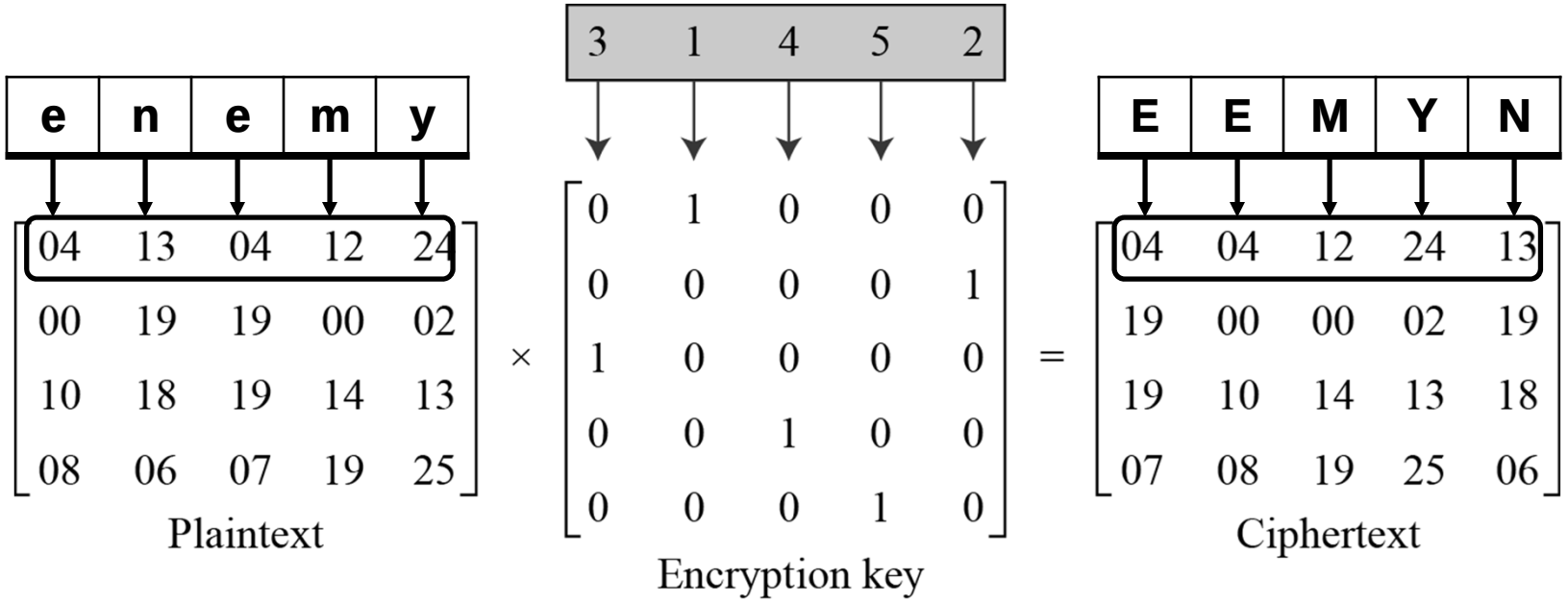
# 3.3.3 Continued

## Using Matrices

**We can use matrices to show the encryption/decryption process for a transposition cipher.**

**Example 3.27**

**Figure 3.24** *Representation of the key as a matrix in the transposition cipher*

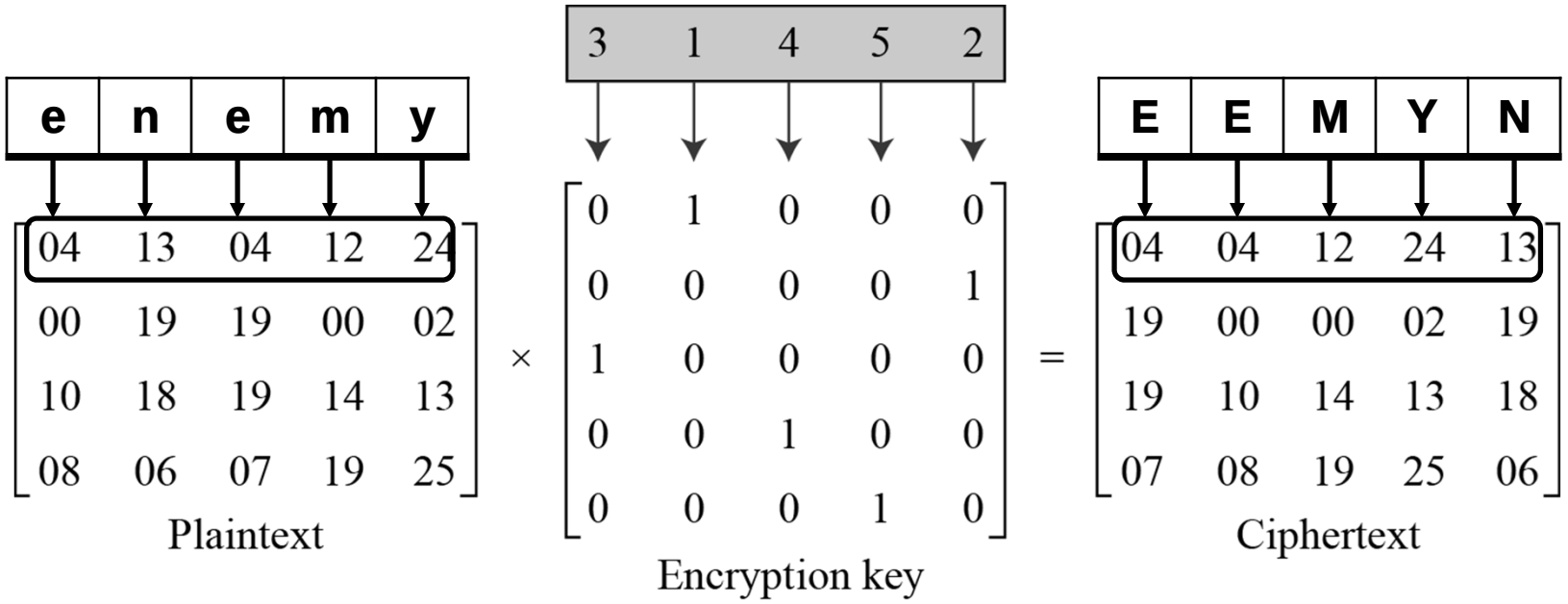


# 3.3.3 Continued

## Example 3.27

**Figure 3.24 shows the encryption process. Multiplying the 4 × 5 plaintext matrix by the 5 × 5 encryption key gives the 4 × 5 ciphertext matrix.**

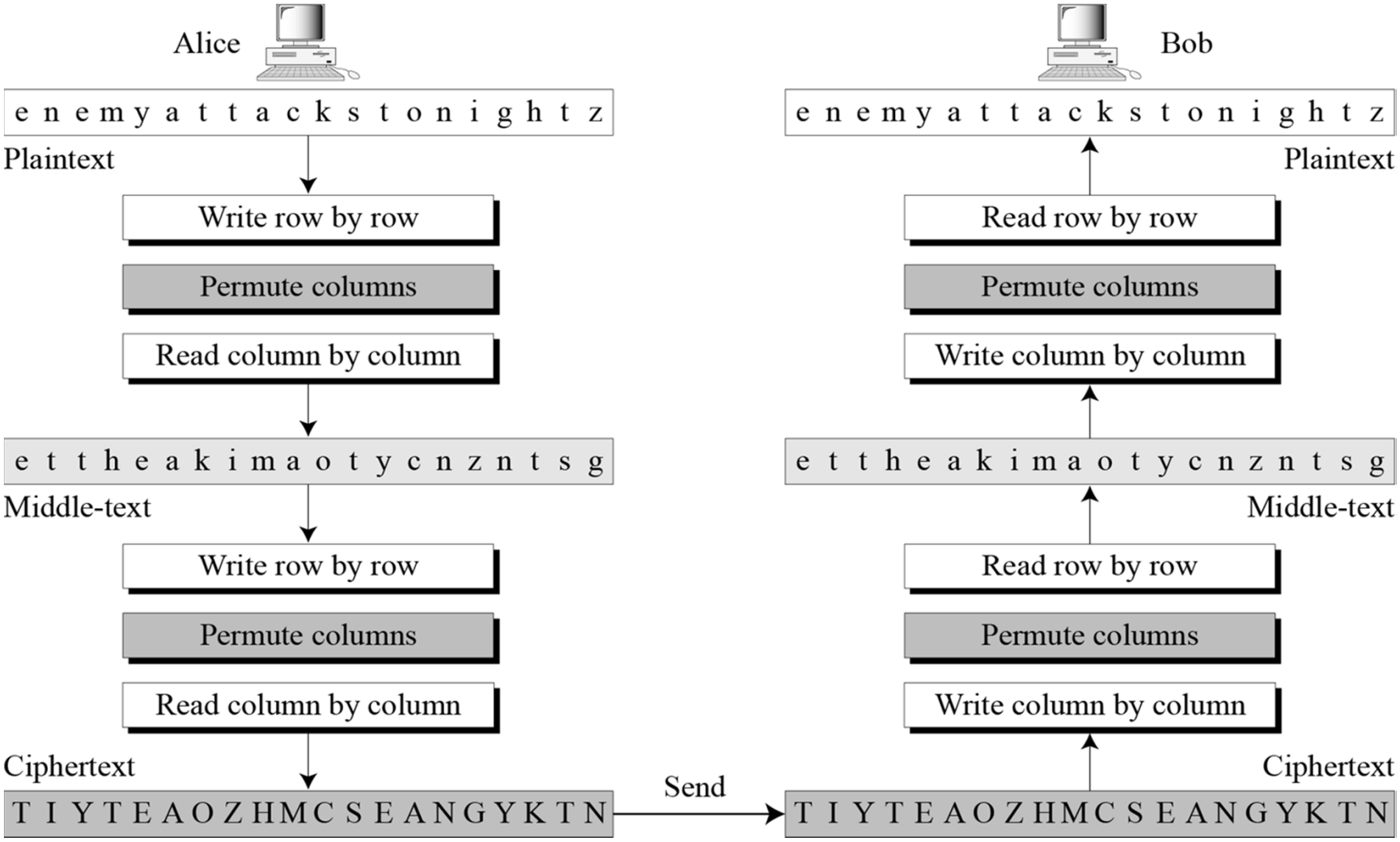
**Figure 3.24** *Representation of the key as a matrix in the transposition cipher*



# 3.3.3 Continued

## Double Transposition Ciphers

**Figure 3.25** *Double transposition cipher*



# 3-4 STREAM AND BLOCK CIPHERS

***The literature divides the symmetric ciphers into two broad categories: stream ciphers and block ciphers. Although the definitions are normally applied to modern ciphers, this categorization also applies to traditional ciphers.***

***Topics discussed in this section:***

**3.4.1 Stream Ciphers**
**3.4.2 Block Ciphers**
**3.4.3 Combination**

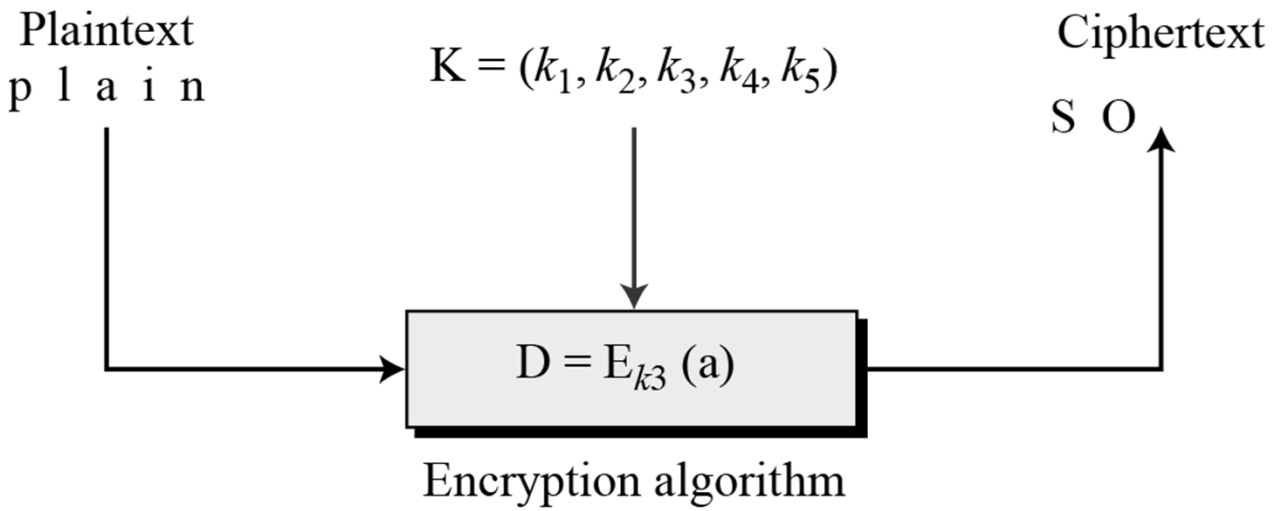
# 3.4.1 Stream Ciphers

**Call the plaintext stream P, the ciphertext stream C, and the key stream K.**

$$P = P_1P_2P_3, \ldots \qquad C = C_1C_2C_3, \ldots \qquad K = (k_1, k_2, k_3, \ldots)$$

$$C_1 = E_{k1}(P_1) \qquad C_2 = E_{k2}(P_2) \qquad C_3 = E_{k3}(P_3) \ldots$$

**Figure 3.26** ***Stream cipher***

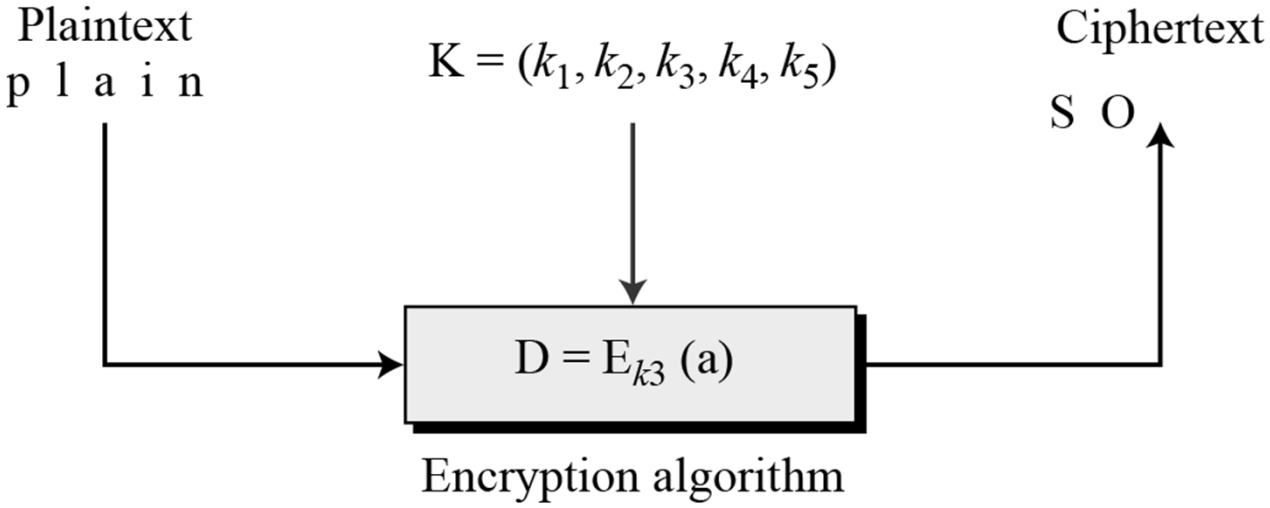


# 3.4.1 Continued

- ✓ **Characters in the plaintext are fed into the encryption algorithm, one at a time.**
- ✓ **The ciphertext characters are also created one at a time.**
- ✓ **The key stream, can be created in many ways:**
  - **It may be a stream of predetermined values.**
  - **It may be created one value at a time using an algorithm.**

**جریان کلید توسط روش‌های متفاوت ایجاد شود:**

- **جریان کلید ممکن است یک جریان از مقادیر از قبل تعیین شده باشد.**
- **یا ممکن است هر مقدار از جریان کلید به صورت یکی یکی و توسط الگوریتم ایجاد گردد.**



# 3.4.1 Continued

**Example 3.30**

**Additive ciphers can be categorized as stream ciphers in which the key stream is the repeated value of the key. In other words, the key stream is considered as a predetermined stream of keys or K = (k, k, …, k). In this cipher, however, each character in the ciphertext depends only on the corresponding character in the plaintext, because the key stream is generated independently.**

**Example 3.31**

**The monoalphabetic substitution ciphers discussed in this chapter are also stream ciphers. However, each value of the key stream in this case is the mapping of the current plaintext character to the corresponding ciphertext character in the mapping table.**

# 3.4.1 Continued

## Example 3.32

**Vigenere ciphers are also stream ciphers according to the definition. In this case, the key stream is a repetition of m values, where *m* is the size of the keyword. In other words,**

$$K = (k_1, k_2, \ldots k_m, k_1, k_2, \ldots k_m, \ldots)$$

## Example 3.33

**We can establish a criterion to divide stream ciphers based on their key streams. We can say that a stream cipher is a monoalphabetic cipher if the value of $k_i$ does not depend on the position of the plaintext character in the plaintext stream; otherwise, the cipher is polyalphabetic.**

**یک الگوریتم رمزنگاری جریانی، الگوریتم رمزنگاری تک حرفی است اگر مقدار $k_i$ به موقعیت کاراکتر plaintext در پیام plaintext وابسته نباشد در غیر اینصورت این اگوریتم رمزنگاری چند حرفی است.**

# 3.4.1 Continued

**Example 3.33 (Continued)**

- **Additive ciphers are definitely monoalphabetic because $k_i$ in the key stream is fixed; it does not depend on the position of the character in the plaintext.**

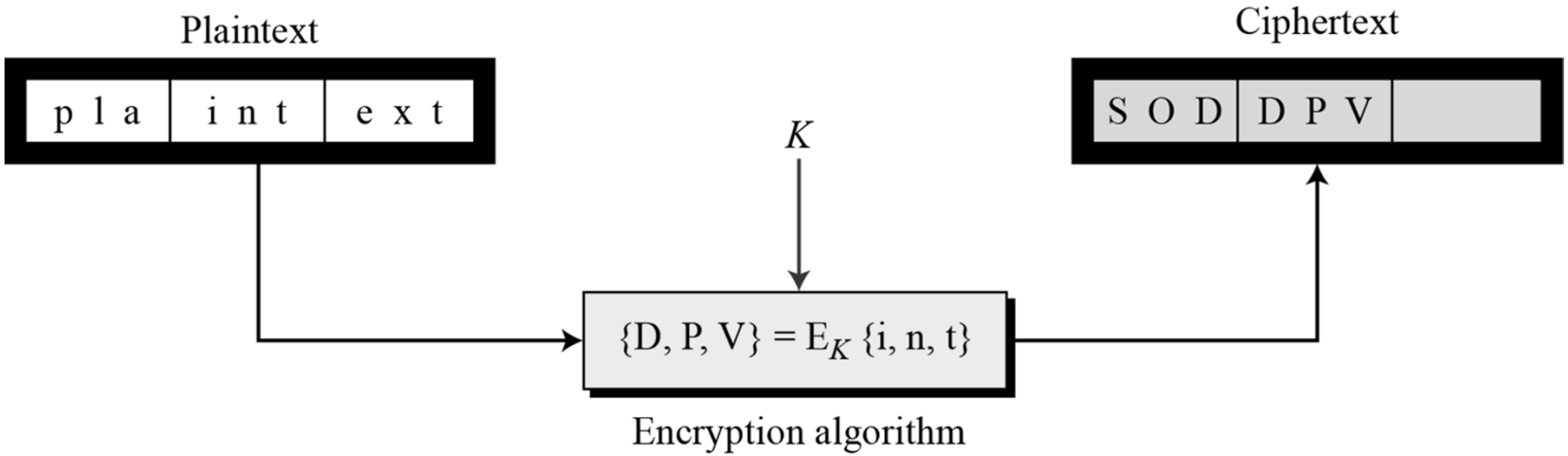
- **Monoalphabetic substitution ciphers are monoalphabetic because $k_i$ does not depend on the position of the corresponding character in the plaintext stream; it depends only on the value of the plaintext character.**

- **Vigenere ciphers are polyalphabetic ciphers because $k_i$ definitely depends on the position of the plaintext character. However, the dependency is cyclic. The key is the same for two characters *m* positions apart.**

# 3.4.2 Stream Ciphers

**In a block cipher, a group of plaintext symbols of size *m* (*m* > 1) are encrypted together creating a group of ciphertext of the same size. A single key is used to encrypt the whole block even if the key is made of multiple values. Figure 3.27 shows the concept of a block cipher.**

**Figure 3.27** ***Block cipher***



# 3.4.2 Continued

**Example 3.34**

**Playfair ciphers are block ciphers. The size of the block is $m = 2$. Two characters are encrypted together.**

**Example 3.35**

**Hill ciphers are block ciphers. A block of plaintext, of size 2 or more is encrypted together using a single key (a matrix). In these ciphers, the value of each character in the ciphertext depends on all the values of the characters in the plaintext. Although the key is made of $m \times m$ values, it is considered as a single key.**

**Example 3.36**

**From the definition of the block cipher, it is clear that every block cipher is a polyalphabetic cipher because each character in a ciphertext block depends on all characters in the plaintext block.**

## 3.4.3 Combination

**In practice, blocks of plaintext are encrypted individually, but they use a stream of keys to encrypt the whole message block by block. In other words, the cipher is a block cipher when looking at the individual blocks, but it is a stream cipher when looking at the whole message considering each block as a single unit.**

**در عمل، بلوک‌های plaintext به طور مجزا رمز می‌شوند امّا این بلوک‌ها از جریانی از کلیدها برای رمزکردن بلوک به بلوک پیام‌ها استفاده می‌کنند. به بیان دیگر، وقتی‌که به یک بلوک به صورت مجزا نگاه کنیم الگوریتم رمزنگاری بلوکی است امّا وقتی‌که به کل پیام نگاه کنیم (هر بلوک به عنوان یک واحد مستقل) نگاه کنیم الگوریتم رمزنگاری جریانی است.**