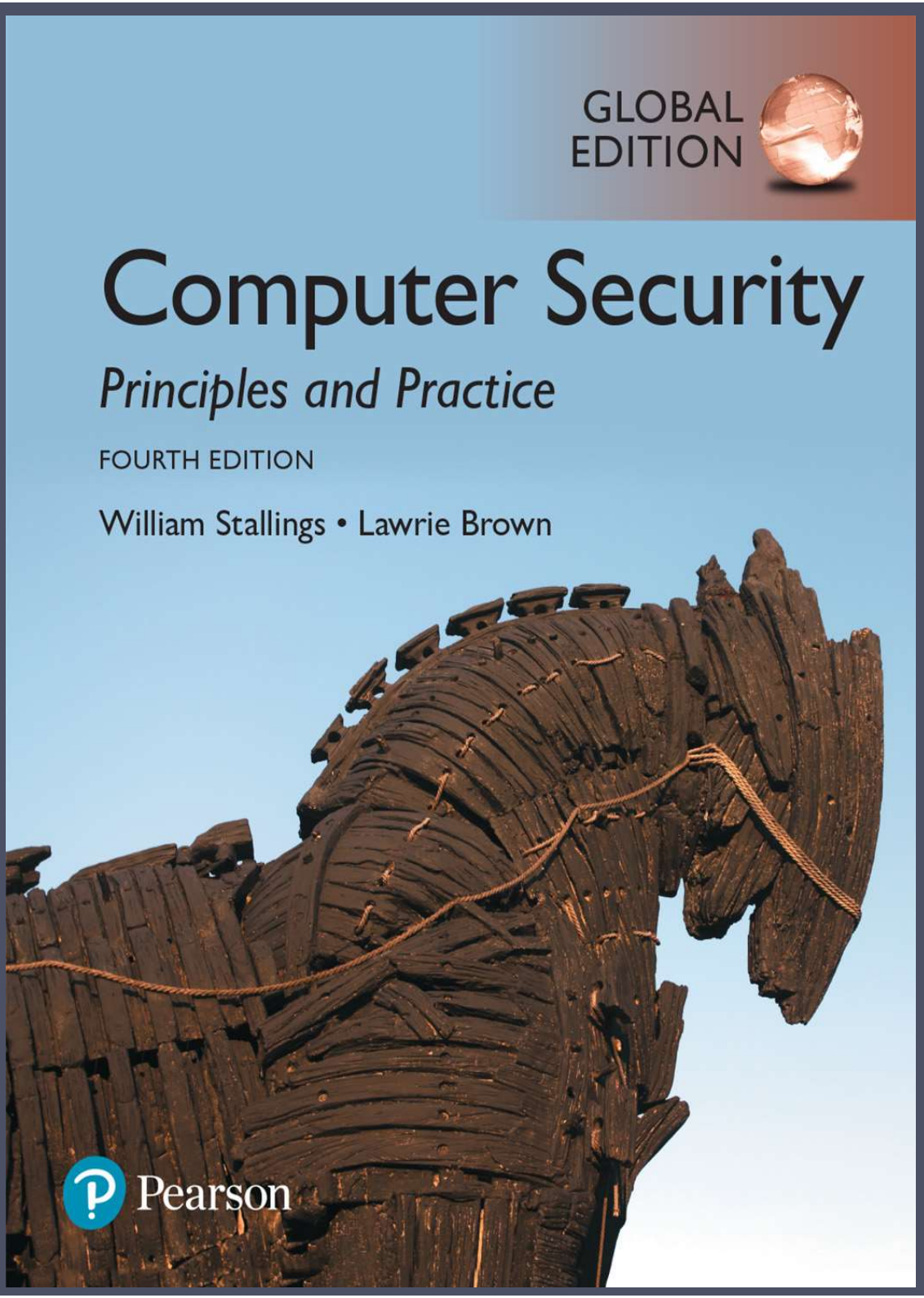

GLOBAL
EDITION
Computer Security
Principles and Practice
FOURTH EDITION
William Stallings • Lawrie Brown
Pearson

Chapter 2

Cryptographic Tools

Symmetric Encryption

Symmetric encryption, also referred to as conventional encryption or single-key encryption, was the only type of encryption in use prior to the introduction of public-key encryption in the late 1970s.

A symmetric encryption scheme has five ingredients (Figure 2.1):

- **Plaintext:** This is the original message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key:** The secret key is also input to the encryption algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.
- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts.
- **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

Symmetric Encryption

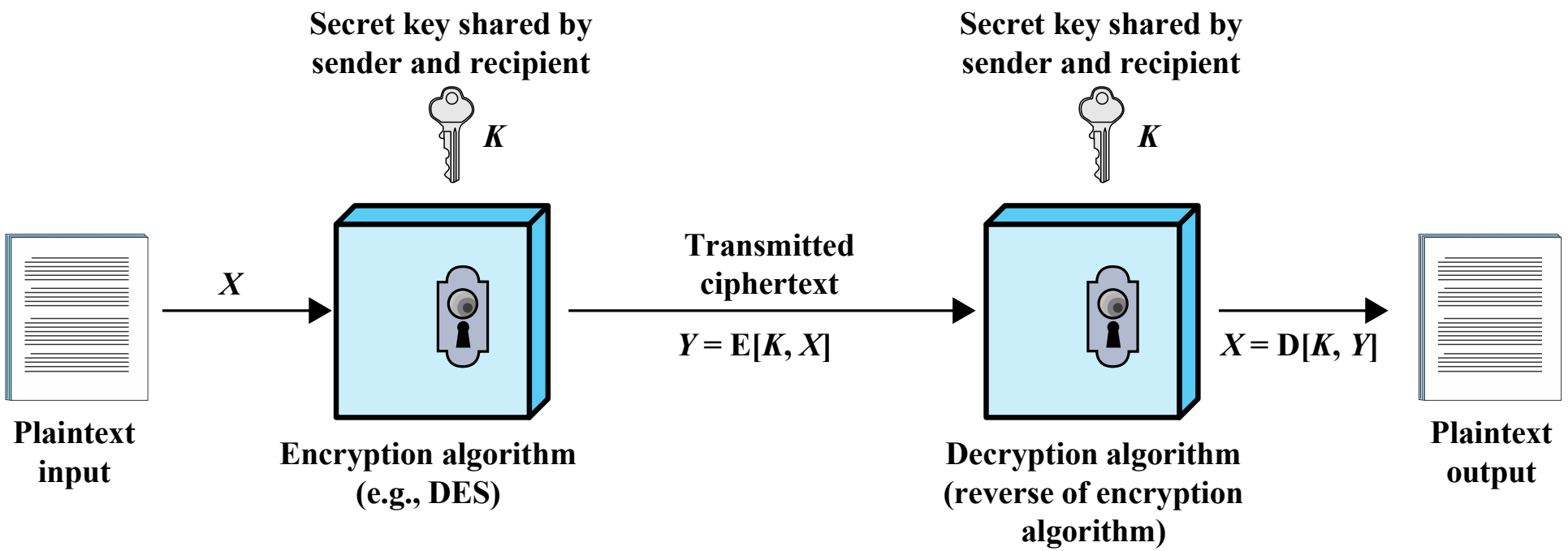


Figure 2.1 Simplified Model of Symmetric Encryption

Requirements for symmetric encryption

There are two requirements for secure use of symmetric encryption:

1. We need a strong encryption algorithm. The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext.
2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure.

Approaches to attack Symmetric Encryption

There are two general approaches to attacking a symmetric encryption scheme.

- The first attack is known as **cryptanalysis**. Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.
- The second method, known as the **brute-force attack**, is to try every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. That is, if there are x different keys, on average an attacker would discover the actual key after x/2 tries.

Types of Symmetric Encryption

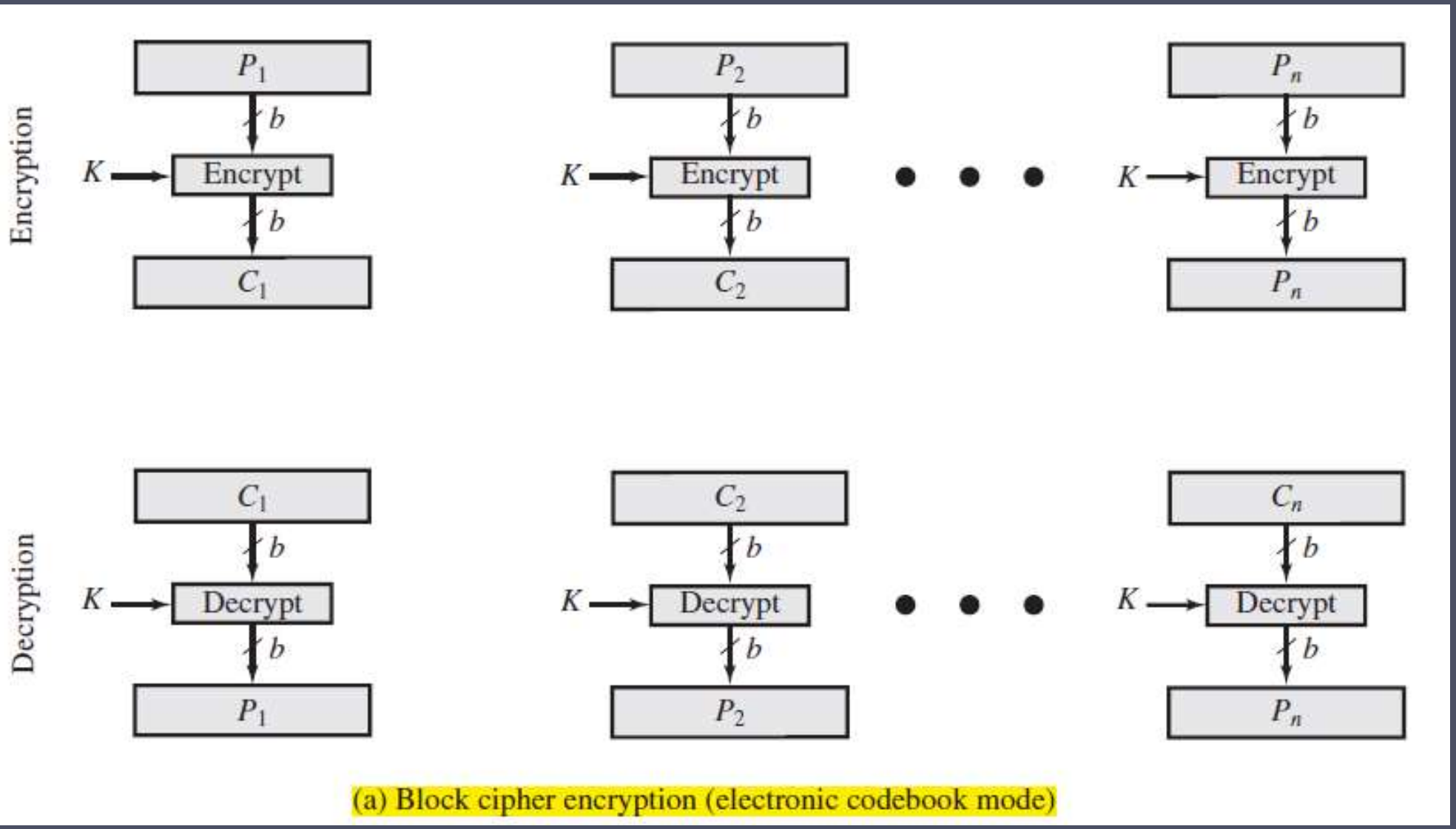
Symmetric encryption is divided into two types

- Block Cipher
- Stream Cipher

Symmetric Block Encryption Algorithms

The most commonly used symmetric encryption algorithms are block ciphers. A block cipher processes the plaintext input in fixed-size blocks and produces a block of ciphertext of equal size for each plaintext block. The algorithm processes longer plaintext amounts as a series of fixed-size blocks. The most important symmetric algorithms, all of which are block ciphers, are the Data Encryption Standard (DES), triple DES, and the Advanced Encryption Standard (AES); see Table 2.1.

Block Cipher Encryption



(a) Block cipher encryption (electronic codebook mode)

Data Encryption Standard

Until recently, the most widely used encryption scheme was based on the Data Encryption Standard (DES) adopted in 1977 by the National Bureau of Standards. DES takes a plaintext block of 64 bits and a key of 56 bits, to produce a ciphertext block of 64 bits.

Comparison of Three Popular Symmetric Encryption Algorithms

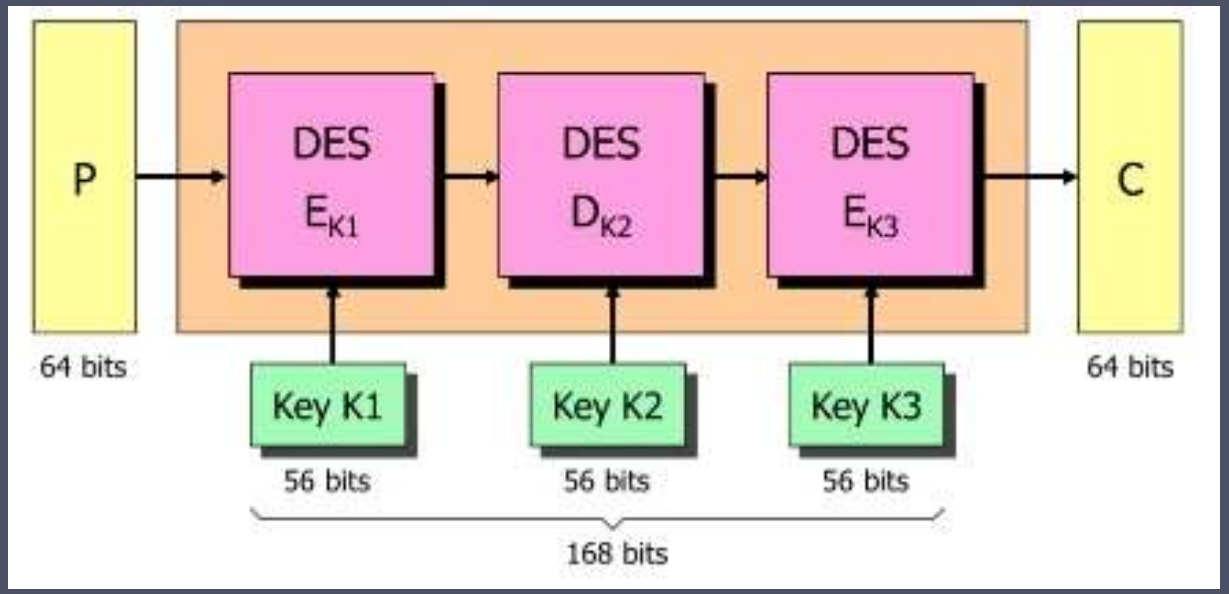
- Table 2.1 Comparison of Three Popular Symmetric Encryption Algorithms

	DES	Triple DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192, or 256

DES = Data Encryption Standard
AES = Advanced Encryption Standard

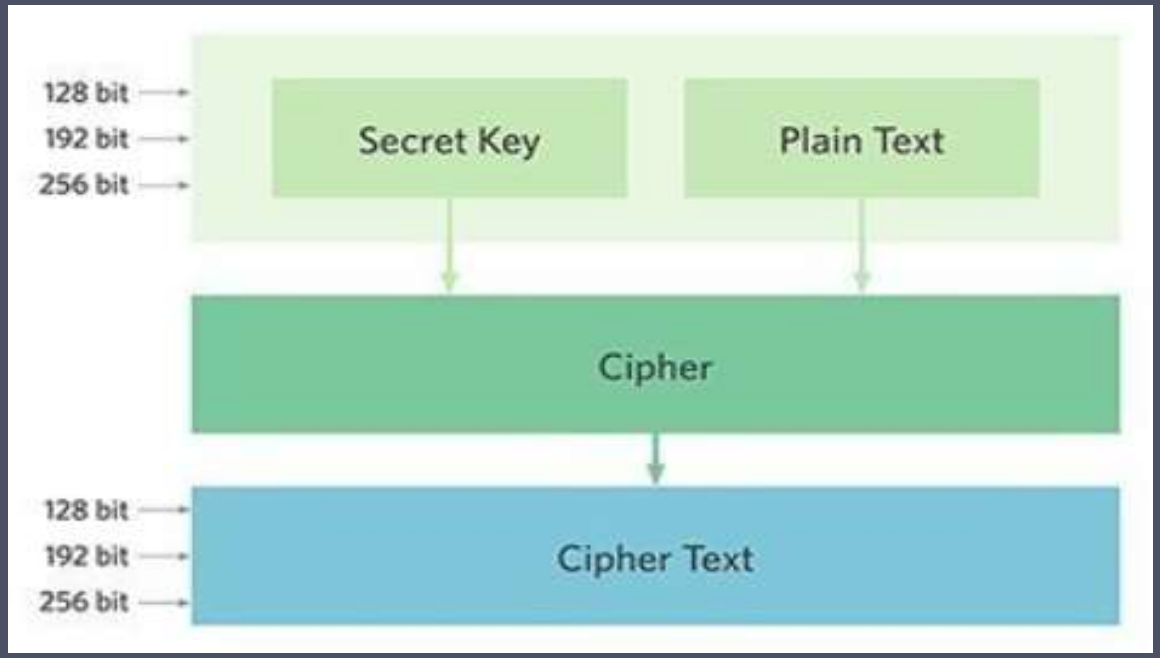
Triple DES

The life of DES was extended by the use of **triple DES** (3DES), which involves repeating the basic DES algorithm three times, using either two or three unique keys, for a key size of 112 or 168 bits. 3DES, which requires three times as many calculations as DES, is correspondingly slower. A secondary drawback is that both DES and 3DES use a 64-bit block size. For reasons of both efficiency and security, a larger block size is desirable.



Advanced Encryption Standard (AES)

Because of its drawbacks, 3DES is not a reasonable candidate for long-term use. As a replacement, NIST in 1997 issued a call for proposals for a new Advanced Encryption Standard (AES).

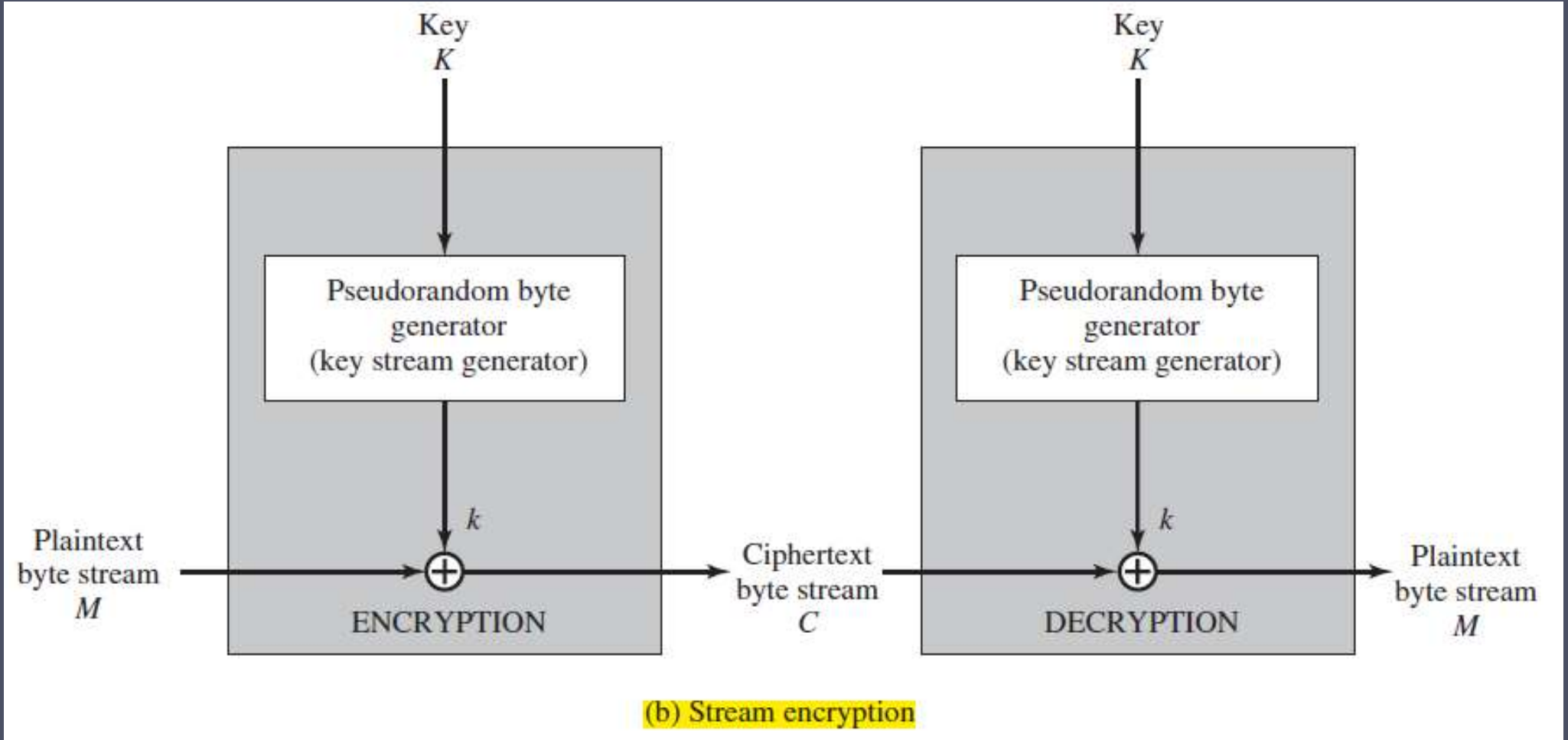


AES must be a symmetric block cipher with a block length of 128 bits and support for key lengths of 128, 192, and 256 bits. NIST selected Rijndael as the proposed AES algorithm. AES is now widely available in commercial products.

Stream Ciphers

A ***stream cipher*** processes the input elements continuously, producing output one element at a time, as it goes along in contrast to a *block cipher that* processes the input elements block by block at a time, producing an output block for each input block. Although block ciphers are far more common, there are certain applications in which a stream cipher is more appropriate.

Stream Cipher



(b) Stream encryption

Block vs Stream Ciphers

Block Cipher

- Processes the input one block of elements at a time
- Produces an output block for each input block
- Can reuse keys
- More common

Stream Cipher

- Processes the input elements continuously
- Produces output one element at a time
- Primary advantage is that they are almost always faster and use far less code
- Encrypts plaintext one byte at a time
- Pseudorandom stream is one that is unpredictable without knowledge of the input key

Message Authentication

- Encryption protects against passive attack (eavesdropping). A different requirement is to protect against active attacks (falsification of data and transactions). Protection against such attacks is known as **message** or **data authentication**.
- A message, file, document, or other collection of data is said to be authentic when it is genuine and came from its alleged source.
- **Message or data authentication** is a procedure that allows communicating parties to verify that received or stored messages are authentic. The two important aspects are to verify that the contents of the message have not been altered and that the source is authentic.

Authentication Using Symmetric Encryption

- It would seem possible to perform authentication simply by the use of symmetric encryption.
- If we assume that only the sender and receiver share a key (which is as it should be), then only the genuine sender would be able to encrypt a message successfully for the other participant, provided the receiver can recognize a valid message.
- In fact, symmetric encryption alone is not a suitable tool for data authentication.

Message Authentication Code (MAC)

- One authentication technique involve the use of a secret key to generate a small block of data, known as a message authentication code, that is appended to the message. This technique assumes that two communicating parties, say A and B, share a common secret key K_{AB}. When
- A has a message to send to B, it calculates the message authentication code as a complex function of the message and the key: $MAC_M = F(K_{AB}, M)$.
- The message plus code are transmitted to the intended recipient. The recipient performs the same calculation on the received message, using the same secret key, to generate a new message authentication code.

Message Authentication Code

The received code is compared to the calculated code (Figure 2.3).

If we assume that only the receiver and the sender know the identity of the secret key, and if the received code matches the calculated code, then:

1. The receiver is assured that the message has not been altered.
2. The receiver is assured that the message is from the alleged sender.

MAC

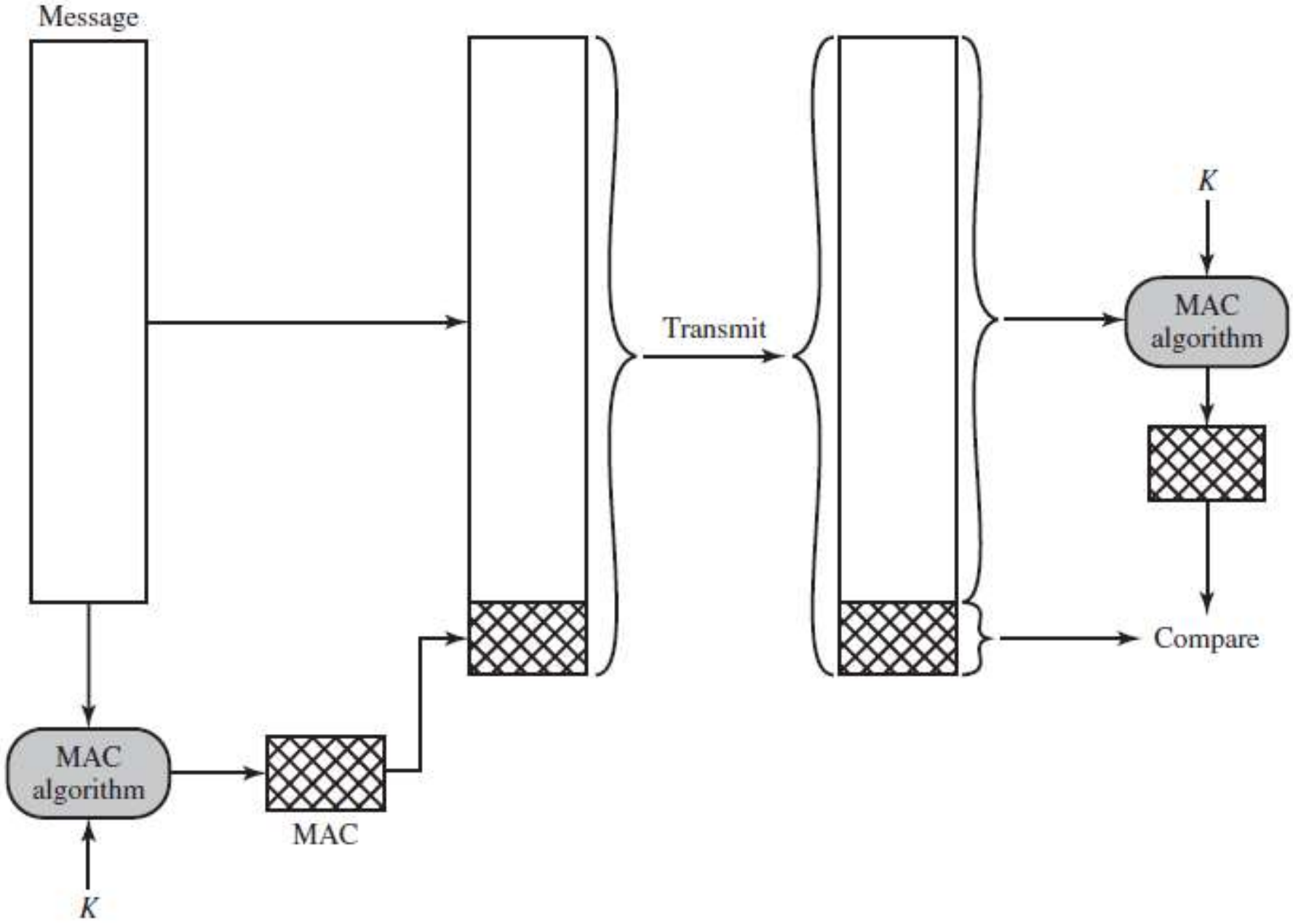


Figure 2.3 Message Authentication Using a Message Authentication Code (MAC)

Hash Function

- An alternative to the message authentication code is the **one-way hash function** *or* **Hash Function**.
- As with the message authentication code, a hash function accepts a variable-size message *M* as input and produces a fixed-size message digest H(*M*) as output (Figure 2.4).
- Unlike the MAC, a hash function does not take a secret key as input.

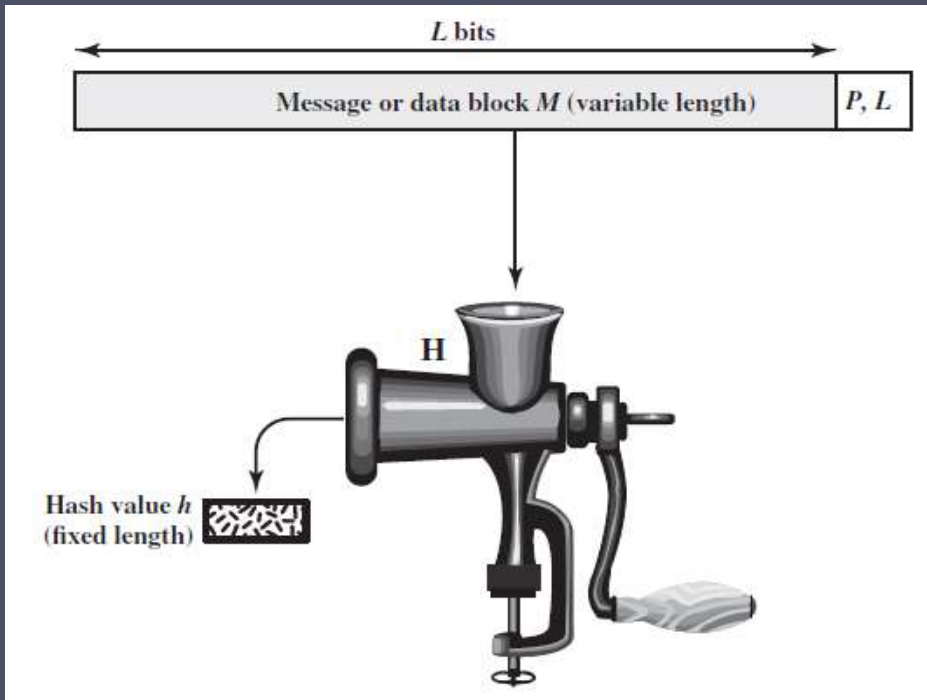


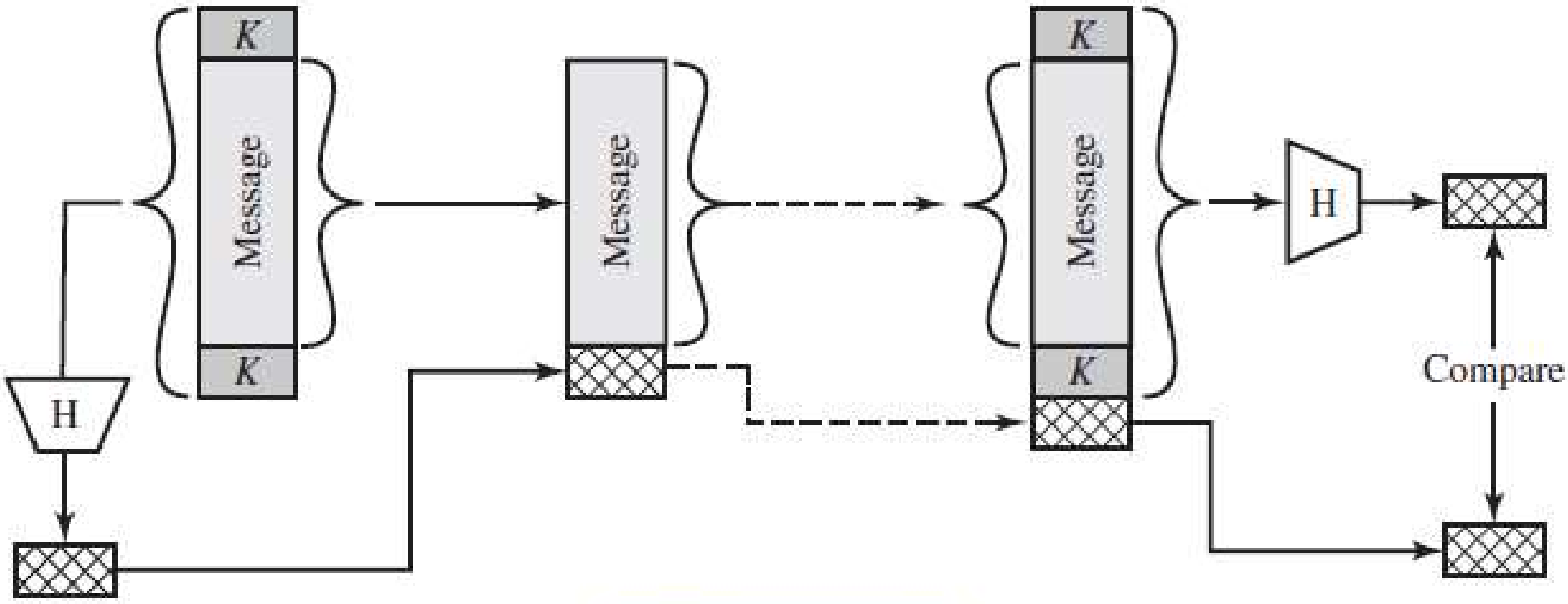
Figure 2.4 Cryptographic Hash Function; $h = H(M)$

Message authentication using Hash function

Figure 2.5 illustrates three ways in which the message can be authenticated using a hash function. The message digest can be encrypted using symmetric encryption (Figure 2.5a); if it is assumed that only the sender and receiver share the encryption key, then authenticity is assured. The message digest can also be encrypted using public-key encryption (Figure 2.5b). The public-key approach has two advantages: It provides a digital signature as well as message authentication; and it does not require the distribution of keys to communicating parties.

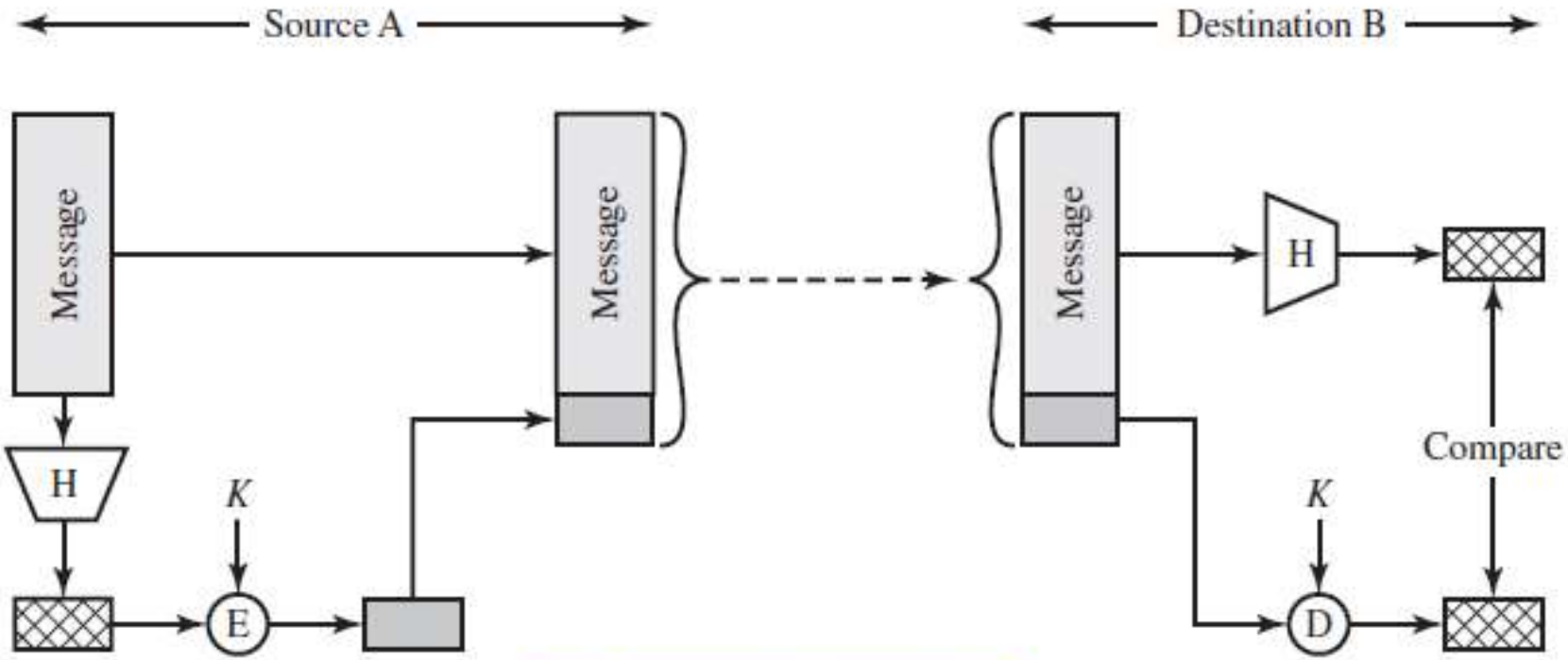
Figure 2.5c shows a technique that uses a hash function but no encryption for message authentication. This technique, known as a keyed hash MAC, assumes that two communicating parties, say A and B, share a common secret key *K*. This secret key is incorporated into the process of generating a hash code.

Message authentication using Hash function

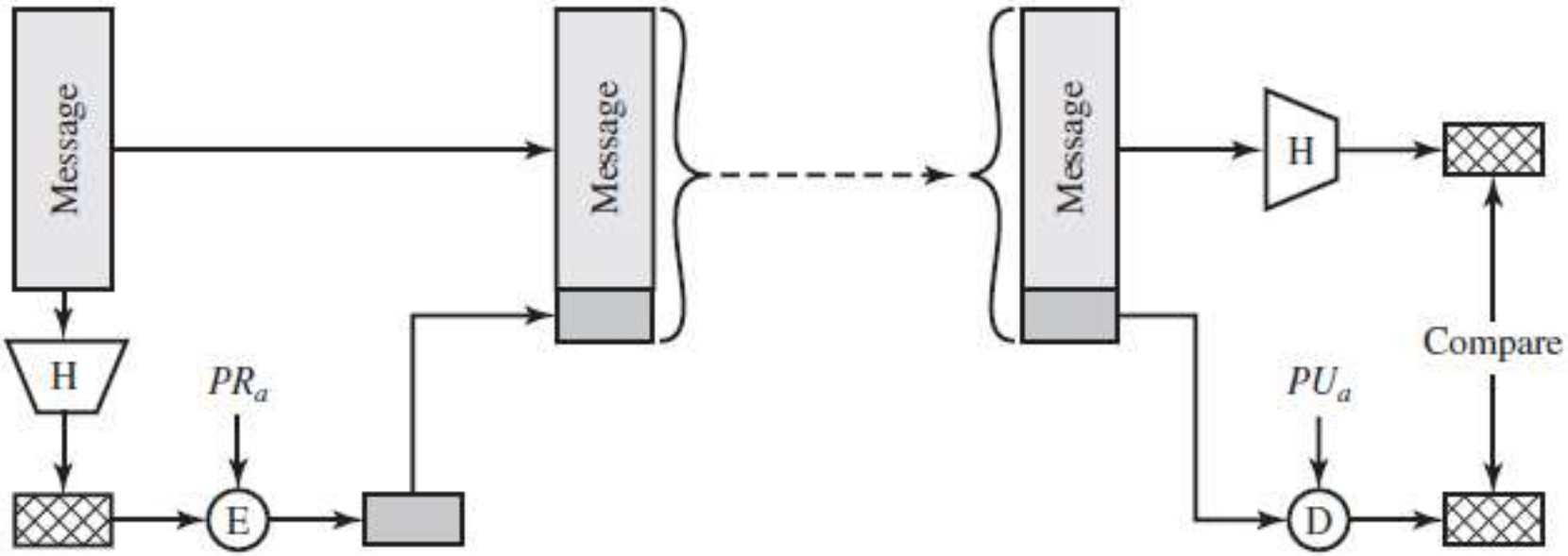


(c) Using secret value

Figure 2.5 Message Authentication Using a One-Way Hash Function



(a) Using symmetric encryption



(b) Using public-key encryption

Hash Function Requirements

The purpose of a hash function is to produce a "fingerprint" of a file, message, or other block of data. To be useful for message authentication, a hash function H must have the following properties:

1. H can be applied to a block of data of any size.
2. H produces a fixed-length output.
3. H(x) is relatively easy to compute for any given x, making both hardware and software implementations practical.
4. For any given code *h*, it is computationally infeasible to find x such that H(x) = *h*. A hash function with this property is referred to as **one-way** or **pre-image resistant**.
5. For any given block x, it is computationally infeasible to find $y \neq x$ with H(y) = H(x). A hash function with this property is referred to as **second pre-image resistant**. This is sometimes referred to as **weak collision resistant**.
6. It is computationally infeasible to find any pair (x, y) such that H(x) = H(y). A hash function with this property is referred to as **collision resistant**. This is sometimes referred to as **strong collision resistant**.

Secure Hash Function Algorithms (SHA)

- In recent years, the most widely used hash function has been the Secure Hash Algorithm (SHA). SHA was developed by the National Institute of Standards and Technology (NIST) and published as a federal information processing standard (FIPS 180) in 1993.
- When weaknesses were discovered in SHA, a revised version was issued as FIPS 180-1 in 1995 and is generally referred to as SHA-1. SHA-1 produces a hash value of 160 bits.
- In 2002, NIST produced a revised version of the standard, FIPS 180–2, that defined three new versions of SHA, with hash value lengths of 256, 384, and 512 bits, known as SHA-256, SHA-384, and SHA-512. These new versions, collectively known as SHA-2.

Public-Key Encryption

- Public-key encryption, first publicly proposed by Diffie and Hellman in 1976, is the first truly revolutionary advance in encryption in literally thousands of years.
- Public-key algorithms are based on mathematical functions rather than on simple operations on bit patterns, such as are used in symmetric encryption algorithms.
- More importantly, public-key cryptography is **asymmetric**, involving the use of two separate keys, in contrast to symmetric encryption, which uses only one key.

A public-key encryption

A public-key encryption scheme has six ingredients (Figure 2.6a):

1. **Plaintext:** This is the readable message or data that is fed into the algorithm as input.
2. **Encryption algorithm:** The encryption algorithm performs various transformations on the plaintext.
3. **Public and private key:** This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact Transformations performed by the encryption algorithm depend on the public or private key that is provided as input.
4. **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.
5. **Decryption algorithm:** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

A public-key encryption

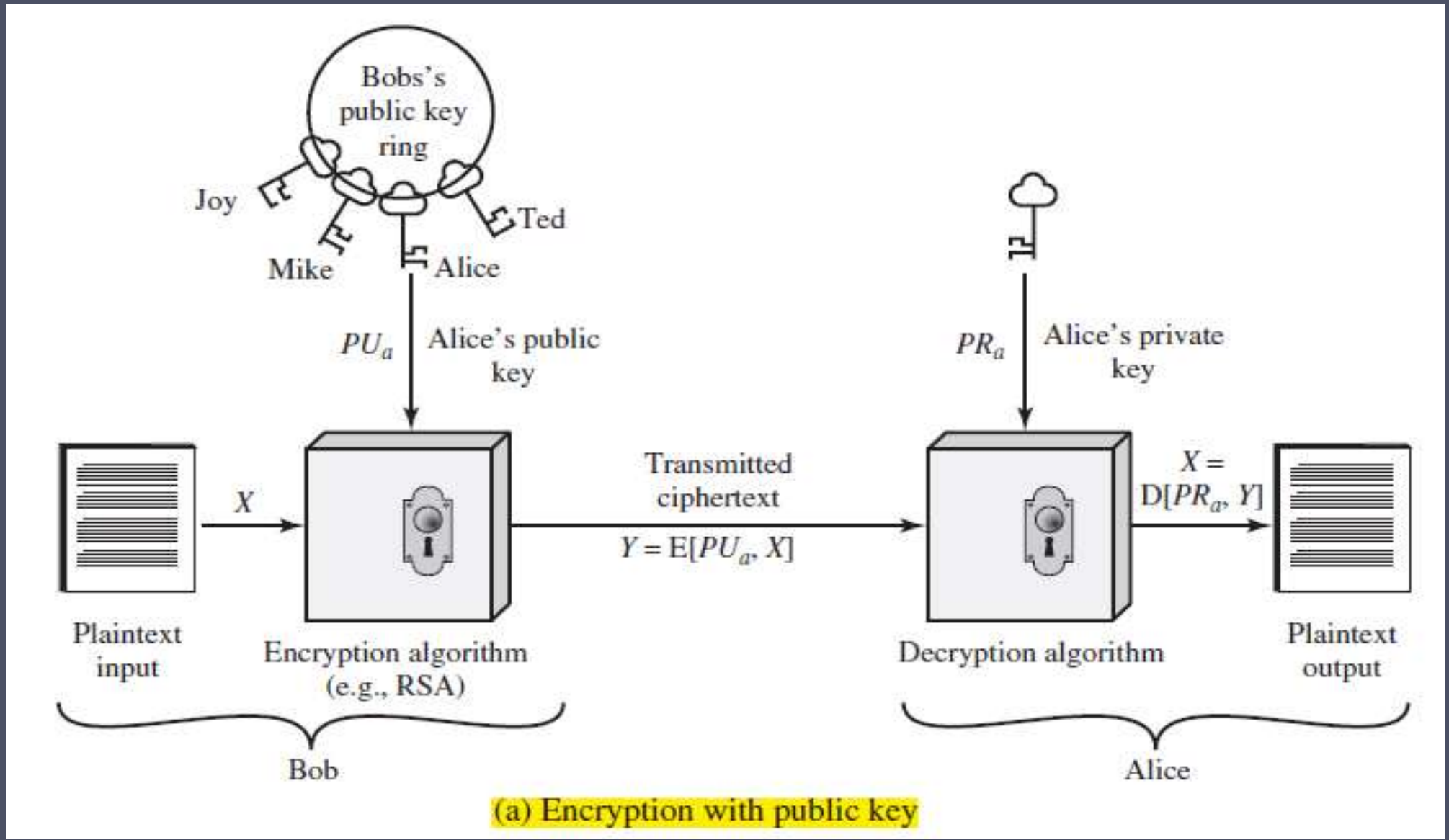
As the names suggest, the public key of the pair is made public for others to use, while the private key is known only to its owner. A general-purpose public-key cryptographic algorithm relies on one key for encryption and a different but related key for decryption.

The essential steps are the following:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private.

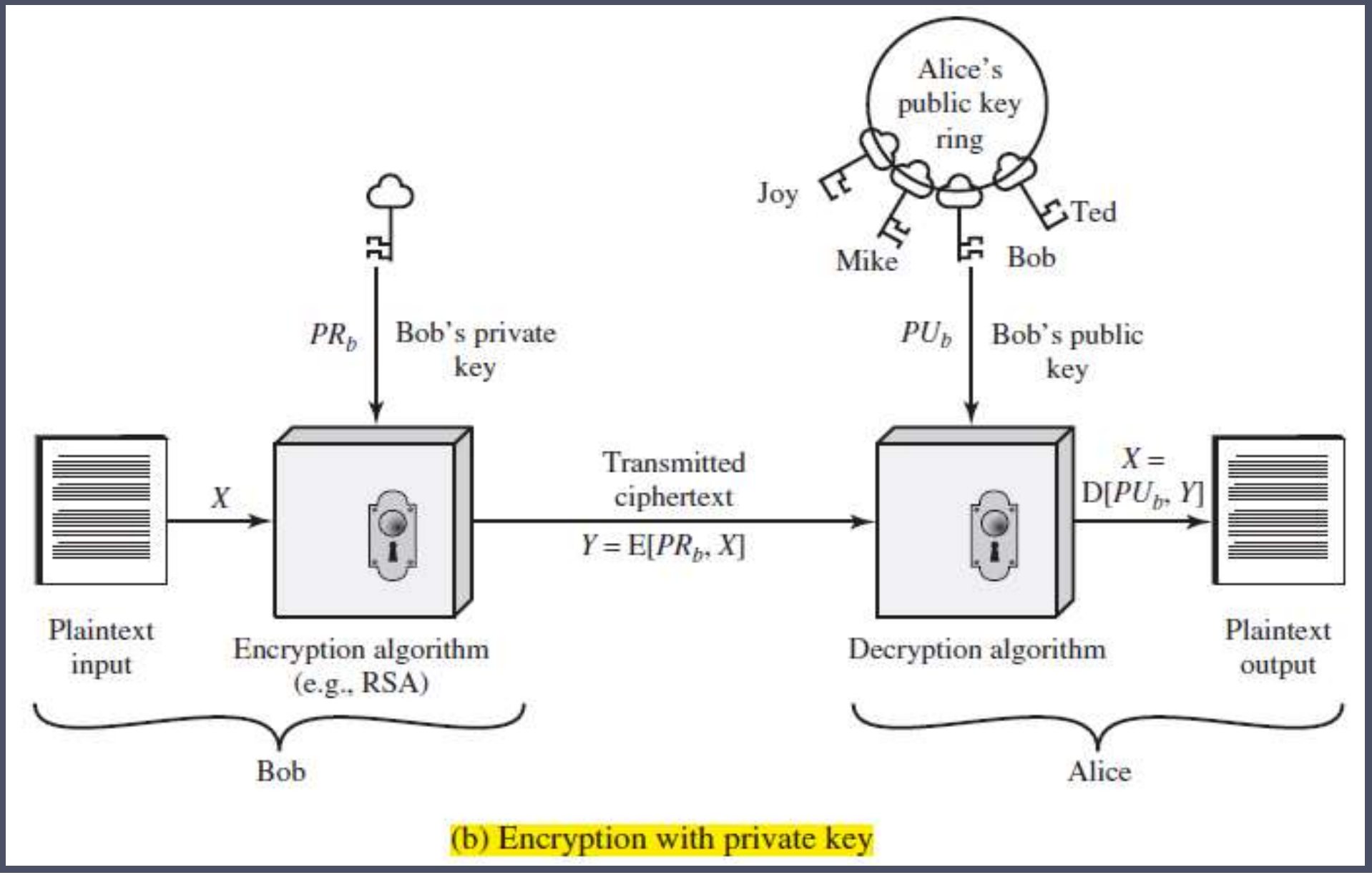
Confidentiality using Public Key Encryption

Scheme of Figure 2.6a is directed toward providing **confidentiality**: Only the intended recipient should be able to decrypt the ciphertext because only the intended recipient is in possession of the required private key.



Authentication using Public Key Encryption

The scheme of Figure 2.6b is directed toward providing **authentication** and/ or **data integrity**.



(b) Encryption with private key

The Security of RSA

Four possible approaches to attacking the RSA algorithm are as follows:

- **Brute force:** This involves trying all possible private keys.
- **Mathematical attacks:** There are several approaches, all equivalent in effort to factoring the product of two primes.
- **Timing attacks:** These depend on the running time of the decryption algorithm.
- **Chosen ciphertext attacks:** This type of attack exploits properties of the RSA algorithm.

The defense against the brute force approach is the same for RSA as for other cryptosystems; namely, use a large key space. Thus, the larger the number of bits in *d*, the better. However, because the calculations involved, both in key generation and in encryption/decryption, are complex, the larger the size of the key, the slower the system will run.

Diffie-Hellman Key Exchange

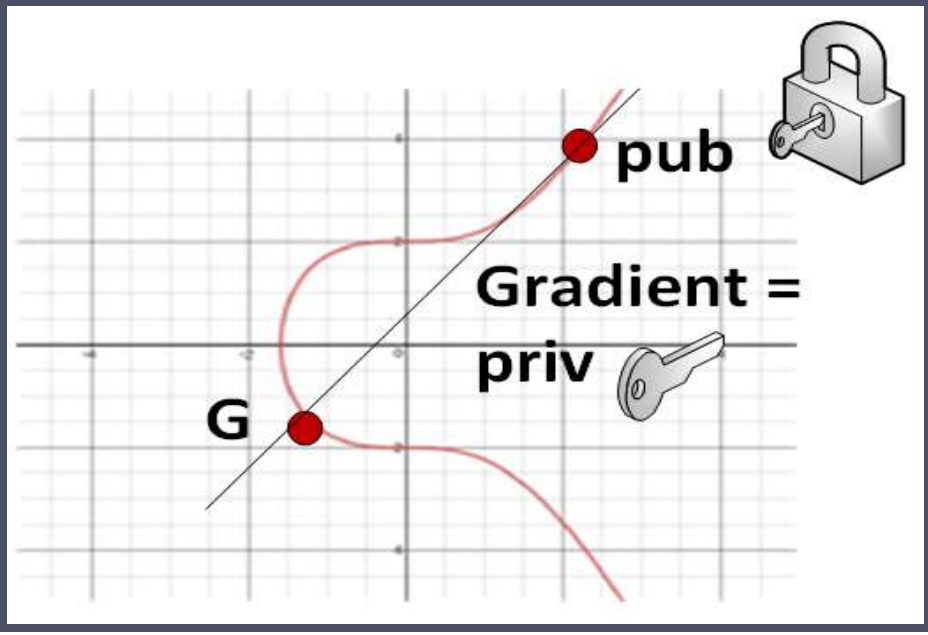
- The first published public-key algorithm appeared in the seminal paper by Diffie and Hellman that defined public-key cryptography and is generally referred to as Diffie-Hellman key exchange.
- The purpose of the algorithm is to enable two users to exchange a secret key securely that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of the keys.
- The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms.

Digital Signature Standard

- The **Digital Signature Standard (DSS)** is a Federal Information Processing Standard specifying a suite of algorithms that can be used to generate digital signatures established by the U.S. National Institute of Standards and Technology (NIST) in 1994.
- The National Institute of Standards and Technology (NIST) has published Federal Information Processing Standard FIPS PUB 186, known as the Digital Signature Standard (DSS).
- The DSS makes use of the SHA-1 and presents a new digital signature technique, the **Digital Signature Algorithm (DSA)**.
- The DSS uses an algorithm that is designed to provide only the digital signature function. Unlike RSA, it cannot be used for encryption or key exchange.

Elliptic-Curve Cryptography

- The principal attraction of **Elliptic-Curve Cryptography (ECC)** compared to RSA is that it appears to offer equal security for a far smaller bit size,
- Thereby reducing processing overhead. On the other hand, although the theory of ECC has been around for some time, it is only recently that products have begun to appear and that there has been sustained cryptanalytic interest in probing for weaknesses.
- Thus, the confidence level in ECC is not yet as high as that in RSA. The technique is based on the use of a mathematical construct known as the elliptic curve.



Thank You!