Cryptographic Tools

January 21, 2022

Cryptographic Tools

An important element in many computer security services and applications is the use of cryptographic algorithms.

In this lecture, we have an overview of the various types of algorithms, together with a discussion of their applicability.

Confidentiality with Symmetric Encryption

- The technique provides **confidentiality** for **transmitted or stored data**.

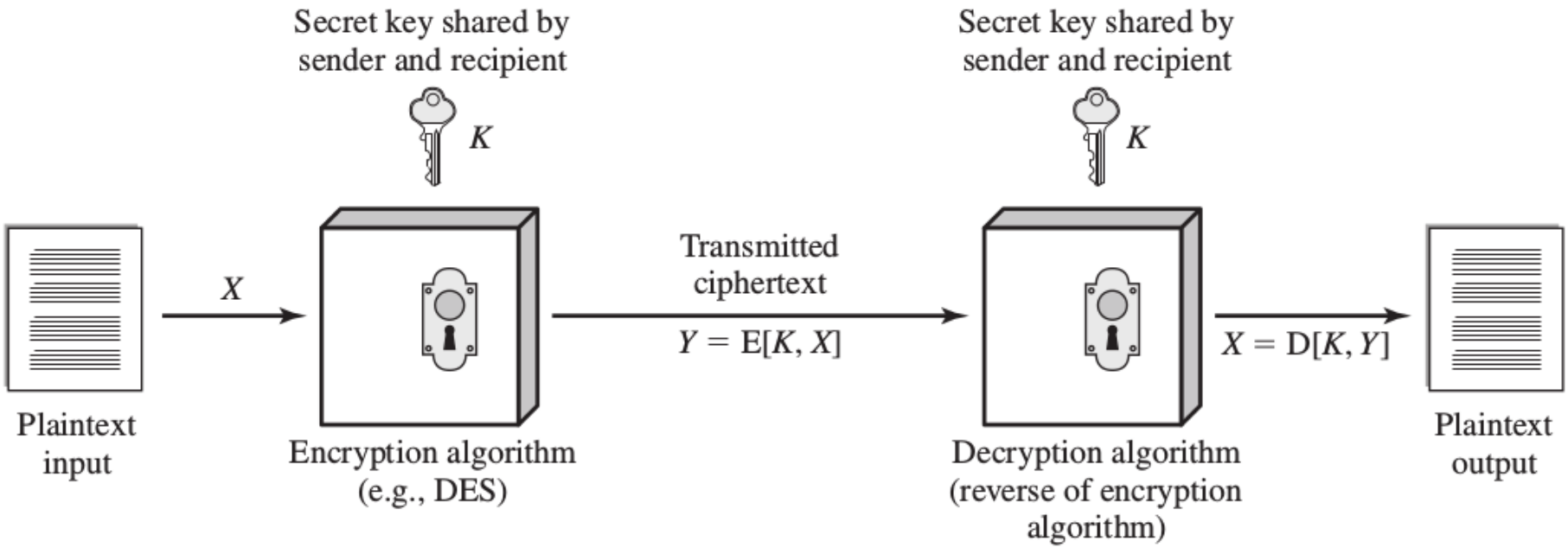


Figure: Symmetric Encryption

Confidentiality with 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.

General approaches to attacking a Symmetric Encryption Algorithm

1. Cryptanalysis: Cryptanalytic attacks rely on the nature of the algorithm plus 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.

2. Brute-force attack, is to try every possible key on a piece of cipher-text until an intelligible translation into plaintext is obtained.

Symmetric Block-encryption Algorithms

1. 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.

2. 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).

Data Encryption Standard

1. DES takes a plaintext block of **64 bits** and a key of **56 bits**, to produce a ciphertext block of 64 bits.
2. Concerns about the strength of DES fall into two categories: concerns about the algorithm itself and concerns about the use of a 56-bit key.

Data Encryption Standard

1. A more serious concern is key length. With a key length of 56 bits, there are 2^{56} possible keys, which is approximately 7.2 * 10^{16} keys.

2. Given the speed of commercial, off-the-shelf processors this key length is woefully inadequate.

Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/μs	Time Required at 10^{13} decryptions/μs
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	$2^{55}\,\mu s = 1.125$ years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	$2^{127}\,\mu s = 5.3 \times 10^{21}$ years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	$2^{167}\,\mu s = 5.8 \times 10^{33}$ years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	$2^{191}\,\mu s = 9.8 \times 10^{40}$ years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	$2^{255}\,\mu s = 1.8 \times 10^{60}$ years	1.8×10^{56} years

Figure: Average Time required for exhaustive key search

Triple DES

1. Triple DES (3DES) involves repeating the basic DES algorithm three times, using either two or three unique keys, for a key size of 112 or 168 bits.
2. 3DES with its 168-bit key length, it overcomes the vulnerability to brute-force attack of DES.
3. There is a high level of confidence that 3DES is very resistant to cryptanalysis.

   If security were the only consideration, then 3DES would be an appropriate choice for a standardized encryption algorithm for decades to come.

Triple DES

1. The principal drawback of 3DES is that the algorithm is relatively sluggish in software.
2. 3DES, which requires three times as many calculations as DES, is correspondingly slower.
3. 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.

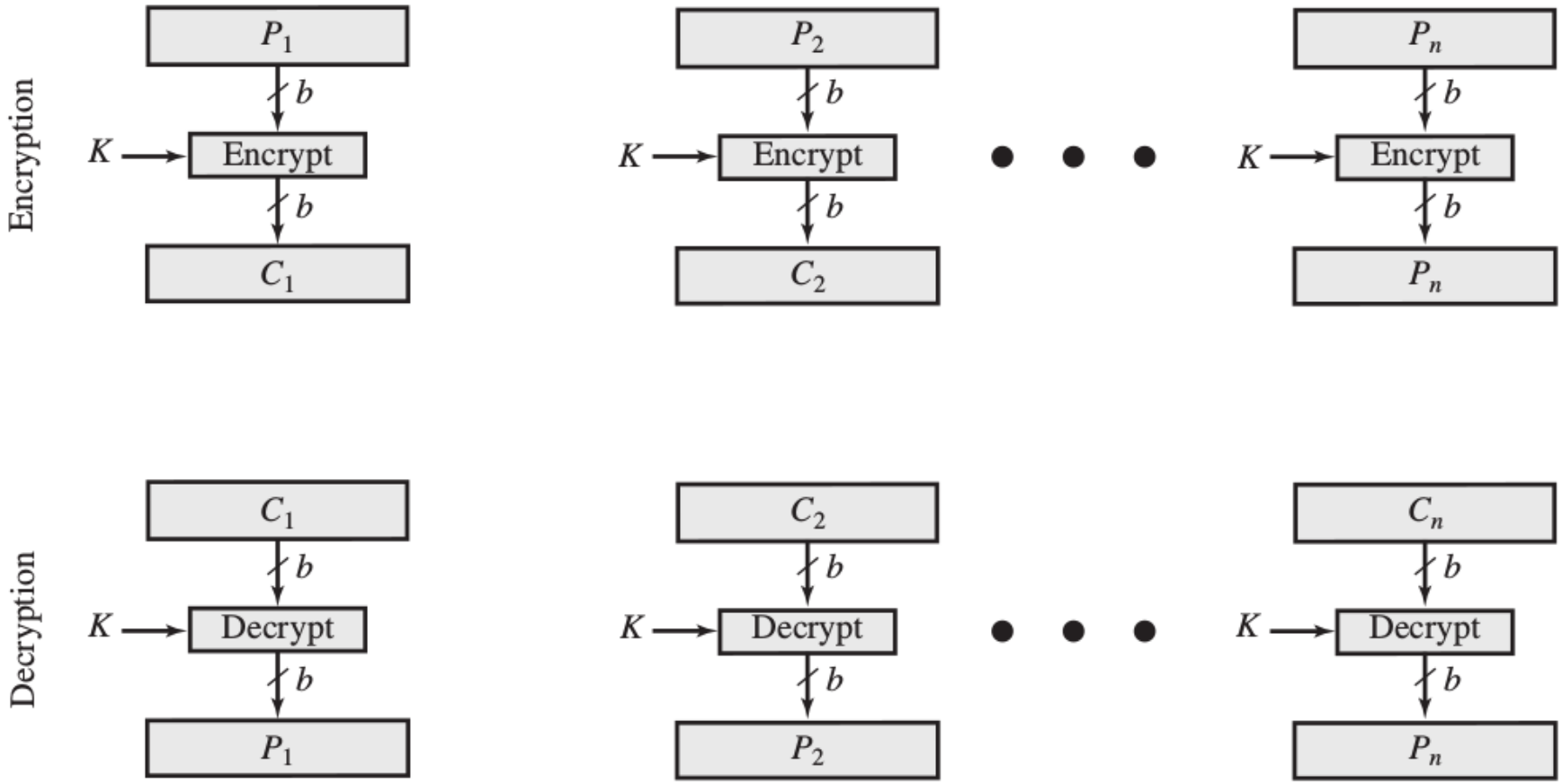
Advanced Encryption Standard

1. **Requirements of 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.
2. Evaluation criteria included security, computational efficiency, memory requirements, hardware and software suitability, and flexibility.
3. NIST selected Rijndael as the proposed AES algorithm. AES is now widely available in commercial products.

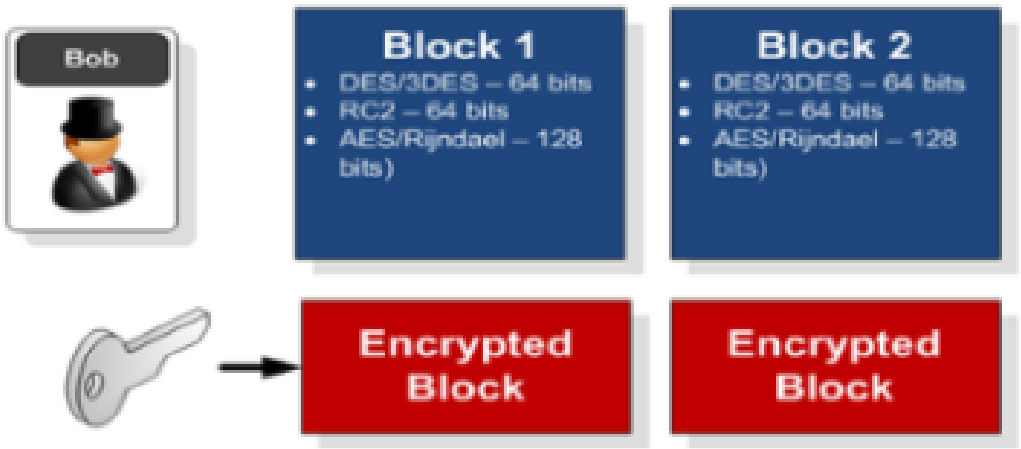
Security Issues

1. Typically, symmetric encryption is applied to a **unit of data larger than a single 64-bit or 128-bit block.**
2. E-mail messages, network packets, database records, and other plaintext sources **must be broken up into a series of fixed-length block** for encryption by a symmetric block cipher.
3. The simplest approach to multiple-block encryption is known as **electronic codebook** (ECB) mode, in which plaintext is handled **b** bits at a time and each block of plaintext is encrypted using the same key.

ECB Mode



ECB Mode



Electronic Code Book (ECB) method. This is weak, as the same cipher text appears for the same blocks.

Hello → 5ghd%43f=
Hello → 5ghd%43f=

Security Issues

1. For lengthy messages, the ECB mode may not be secure. A cryptanalyst may be able to exploit **regularities** in the plaintext to ease the task of decryption.

2. For example, if it is known that the message always starts out with certain predefined fields, then **the cryptanalyst may have a number of known plaintext-ciphertext pairs to work with.**

3. To increase the security of symmetric block encryption for large sequences of data, a number of alternative techniques have been developed, called **modes of operation**.

Other Mode

Mode	Description	Typical Application
Electronic Code book (ECB)	Each block of 64 plaintext bits is encoded independently using the same key.	• Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.	• General-purpose block-oriented transmission • Authentication
Cipher Feedback (CFB)	Input is processed s bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	• General-purpose stream-oriented transmission • Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding DES output.	• Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	• General-purpose block-oriented transmission • Useful for high-speed requirements

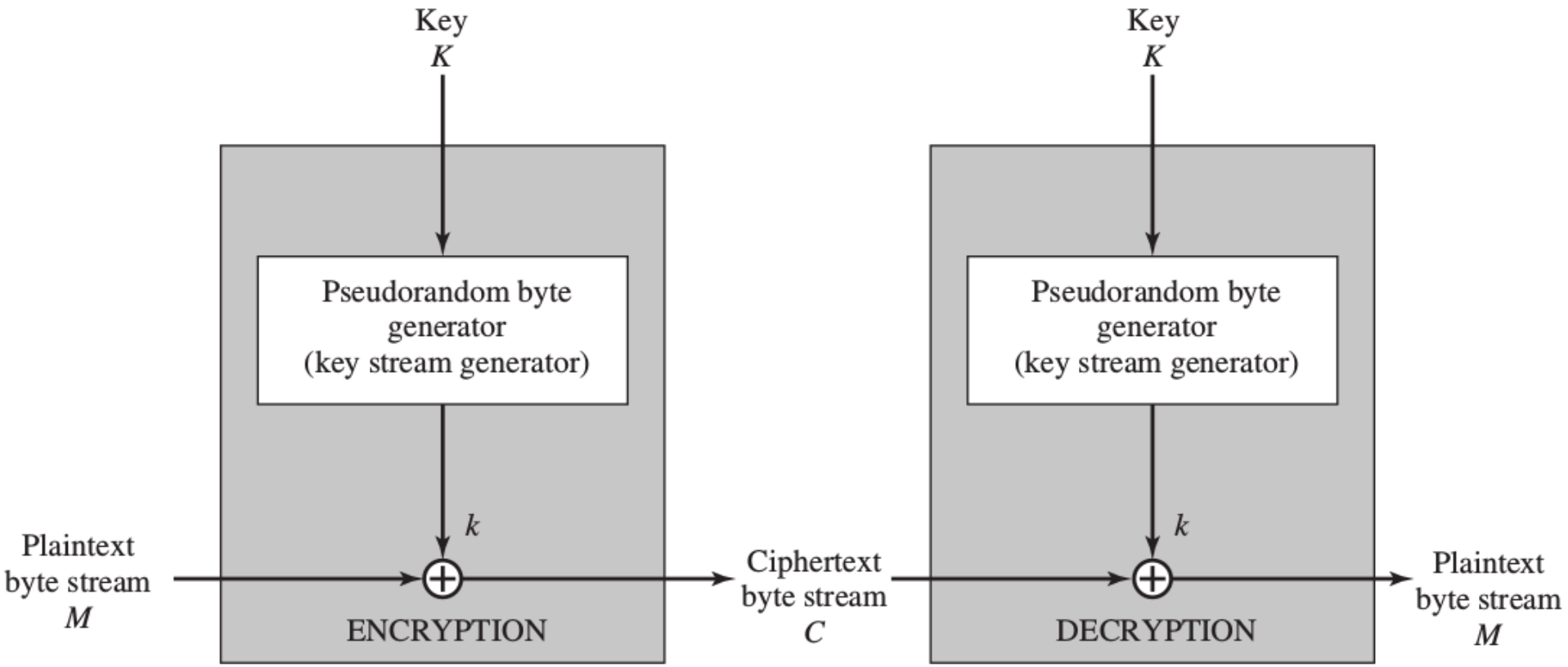
Stream Ciphers

1. A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along.

2. A typical **stream cipher encrypts plaintext one byte at a time**, although a stream cipher may be designed to operate on one bit at a time or on units larger than a byte at a time.

3. In this structure, a key is input to a pseudorandom bit generator that produces a stream of 8-bit numbers that are apparently random. The output of the generator, called a keystream, is combined one byte at a time with the plaintext stream using the bitwise exclusive OR (XOR) operation.

Stream Ciphers



Message Authentication and Hash Functions

1. Encryption protects against passive attack (eavesdropping). A different requirement is to protect against active attack (falsification of data and transactions). Protection against such attacks is known as **message or data authentication.**

2. A message, file, document, or other collection of data is said to be authentic when it is genuine and came from its alleged source.

3. Message or data authentication is a procedure that allows communicating parties to verify that received or stored messages are authentic

Message Authentication and Hash Functions

1. The two important aspects are to **verify that the contents of the message have not been altered** and that the **source is authentic**.
2. We may also wish to verify a message's timeliness (it has not been artificially delayed and replayed) and sequence relative to other messages flowing between two parties.
3. All of these concerns come under the category of data integrity.

Authentication Using Symmetric Encryption

1. 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.
2. Further, if the message includes an error-detection code and a sequence number, the receiver is assured that no alterations have been made and that sequencing is proper.
3. If the message also includes a **timestamp**, the receiver is assured that the message has not been delayed beyond that normally expected for network transit.

Message Authentication without Encryption

There are situations in which message authentication without confidentiality is preferable:

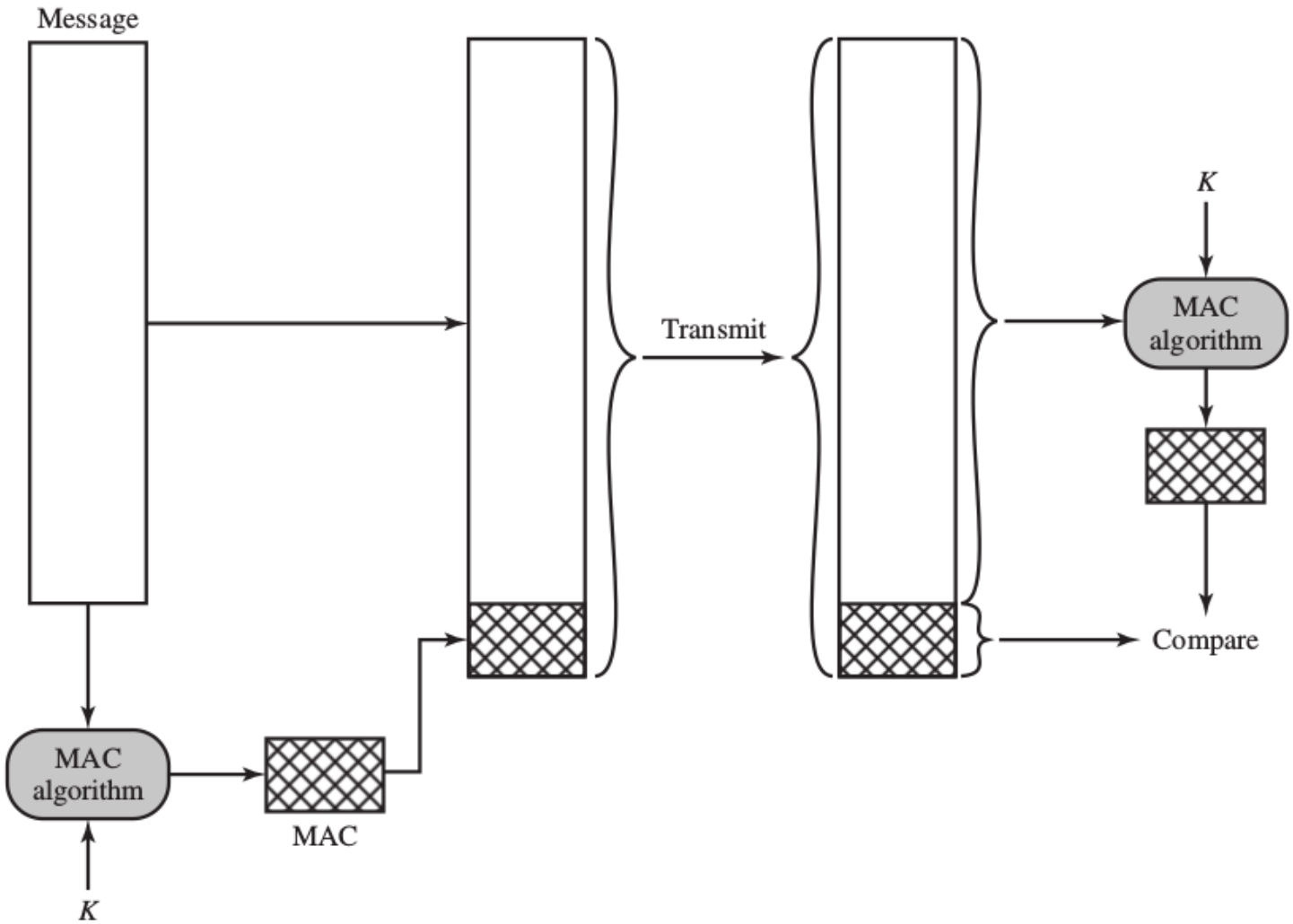
1. There are a number of applications in which the same message is **broadcast** to a number of destinations.
2. An exchange in which one side has a **heavy load and cannot afford the time to decrypt** all incoming messages.

Message Authentication Code

1. This authentication technique involves 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.

2. 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:
   $MAC_M = \mathrm{F}(K_{AB}, \mathrm{M})$.

Message Authentication Code



Message Authentication Code

1. Because messages may be any size and the message authentication code is a small fixed size, there must theoretically be many messages that result in the same MAC.

   However, it should be **infeasible in practice** to find pairs of such messages with the same MAC. This is known as **collision resistance**.

MAC Ensures:

- The receiver is assured that the message has not been altered.
- The receiver is assured that the message is from the alleged sender.
- If the message includes a sequence number, then the receiver can be assured of the proper sequence.

Message Authentication Code

1. DES is used to generate an encrypted version of the message, and the last number of bits of ciphertext are used as the code. A 16-or 32-bit code is typical.

2. One difference is that the authentication algorithm need not be reversible, as it must for decryption. It turns out that because of the mathematical properties of the authentication function, it is less vulnerable to being broken than encryption.

One way hash function

1. A hash function accepts a variable-size message **M** as input and produces a fixed-size message digest **H(M)** as output.
2. Typically, the **message is padded** out to an integer multiple of some fixed length (e.g., 1024 bits) and the padding includes the value of the length of the original message in bits.
3. The length field is a security measure to increase the difficulty for an attacker to produce an alternative message with the same hash value.

One way hash function

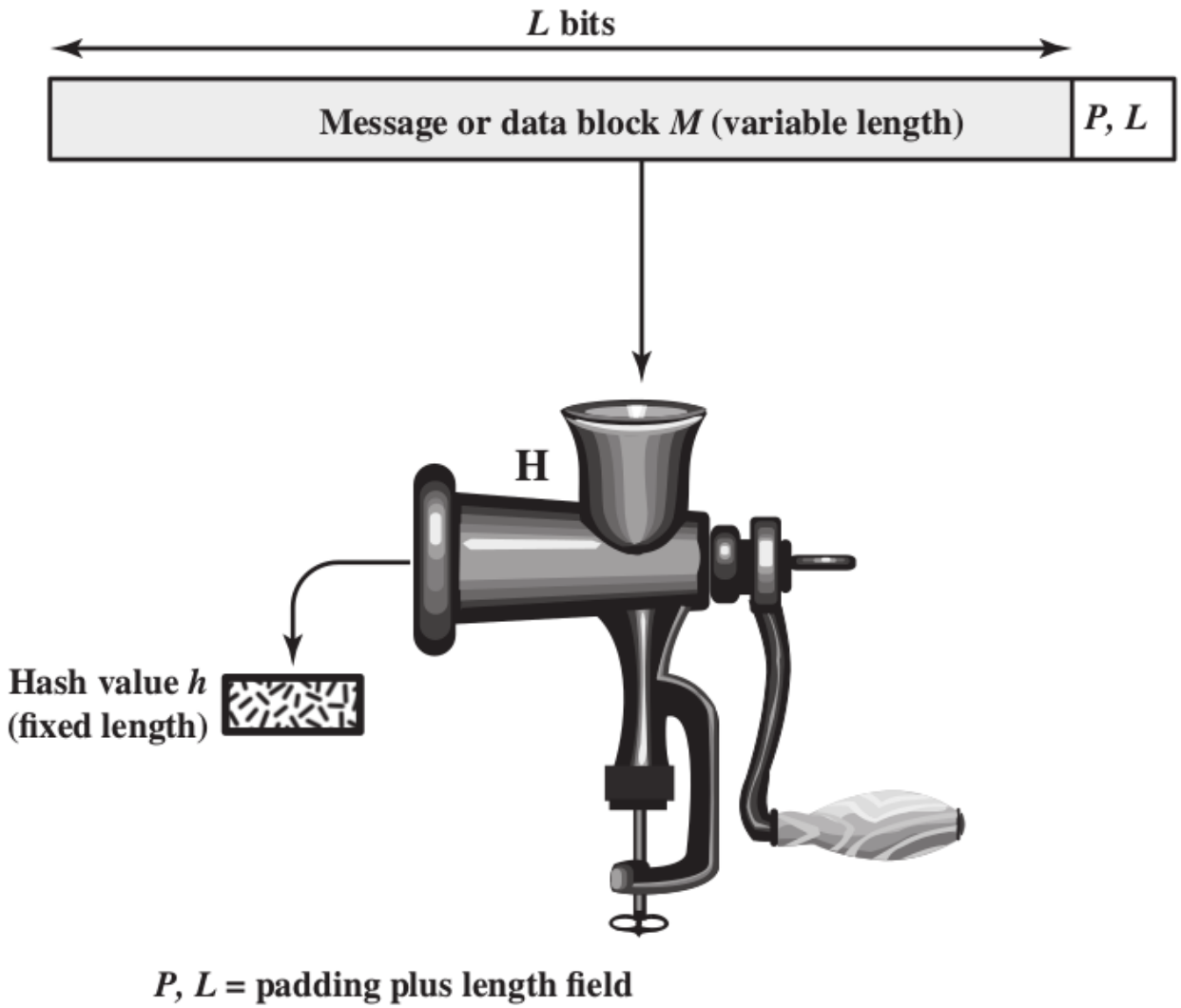
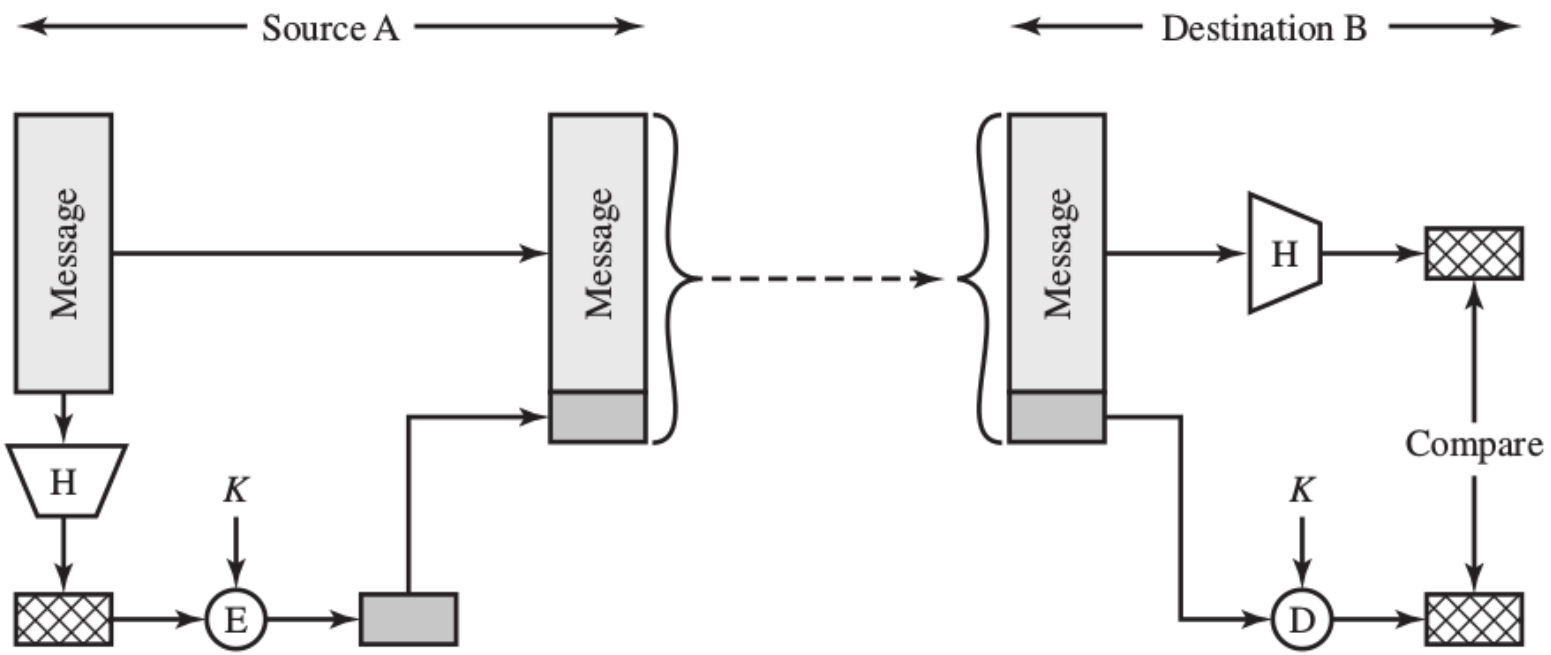


Figure: Cryptographic Hash Function; h = H(M)

One way hash function

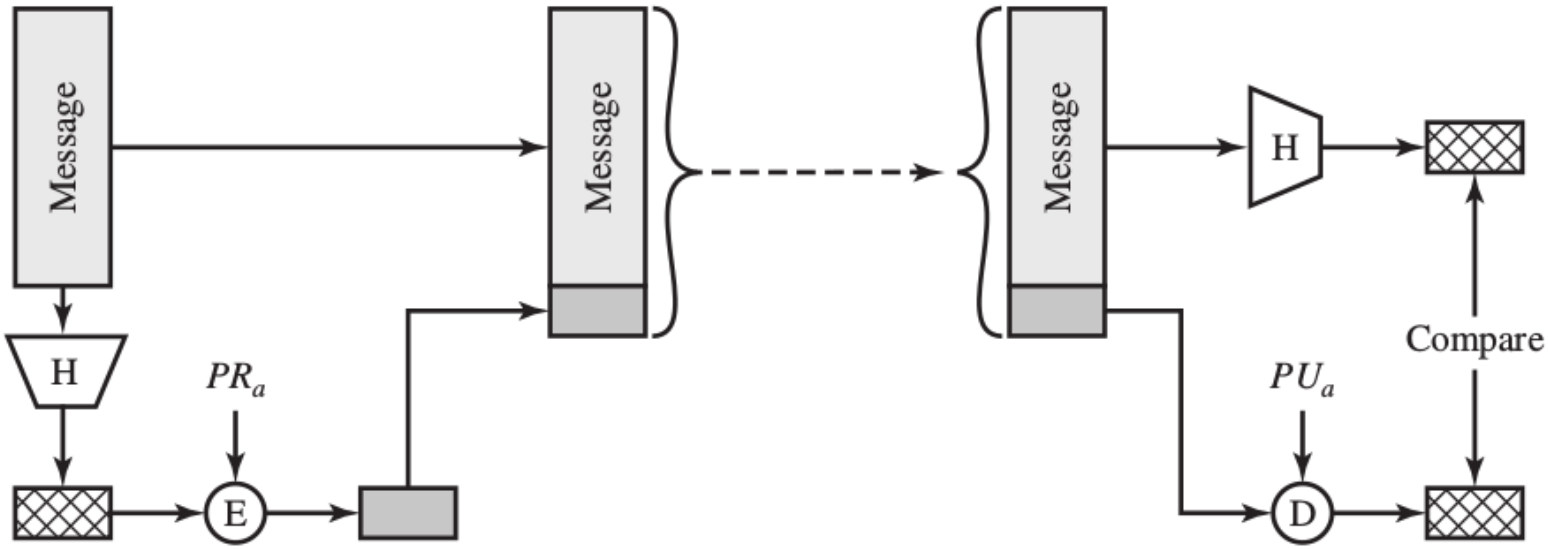
Unlike the MAC, a hash function does not take a secret key as input.



(a) Using symmetric encryption

One way hash function

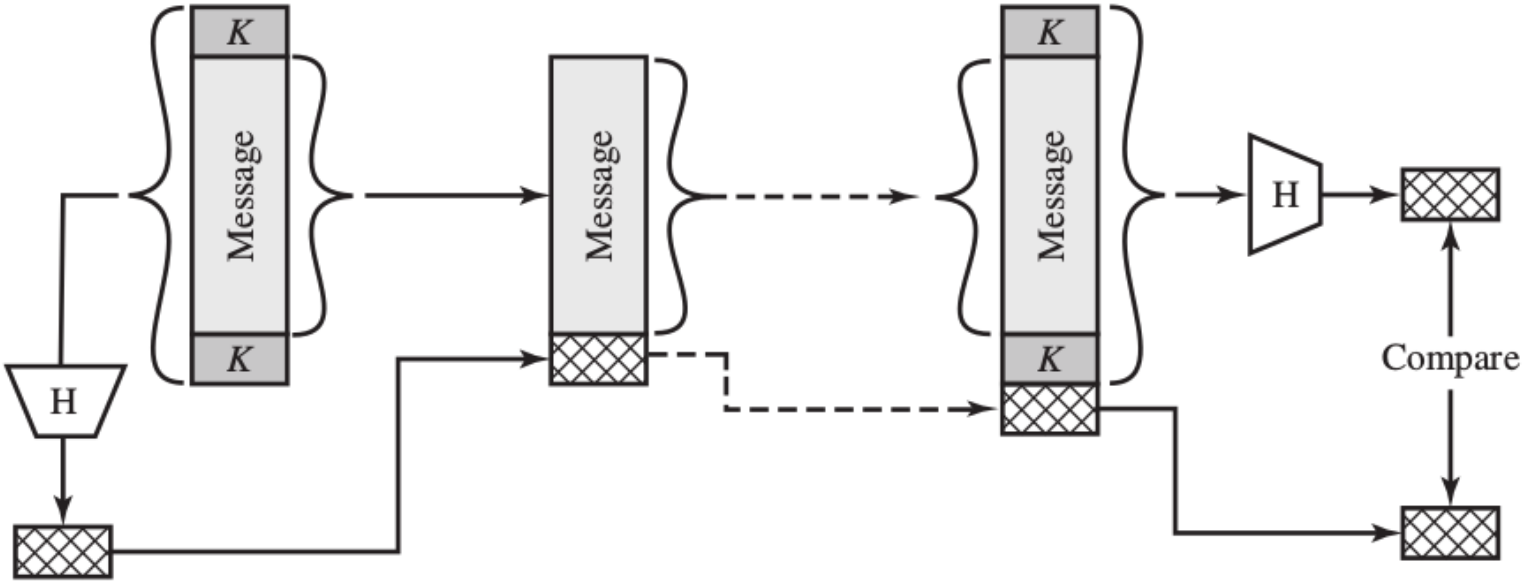
Unlike the MAC, a hash function does not take a secret key as input.



(b) Using public-key encryption

One way hash function

This technique, known as a keyed hash MAC, assumes that two communicating parties, say **A and B**, share a common secret key **K**.



(c) Using secret value

Secure hash functions

1. The one-way hash function, or secure hash function, is important not only in message authentication but in **digital signatures**.

Hash function requirements

Hash function is used 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.**

Hash function requirements

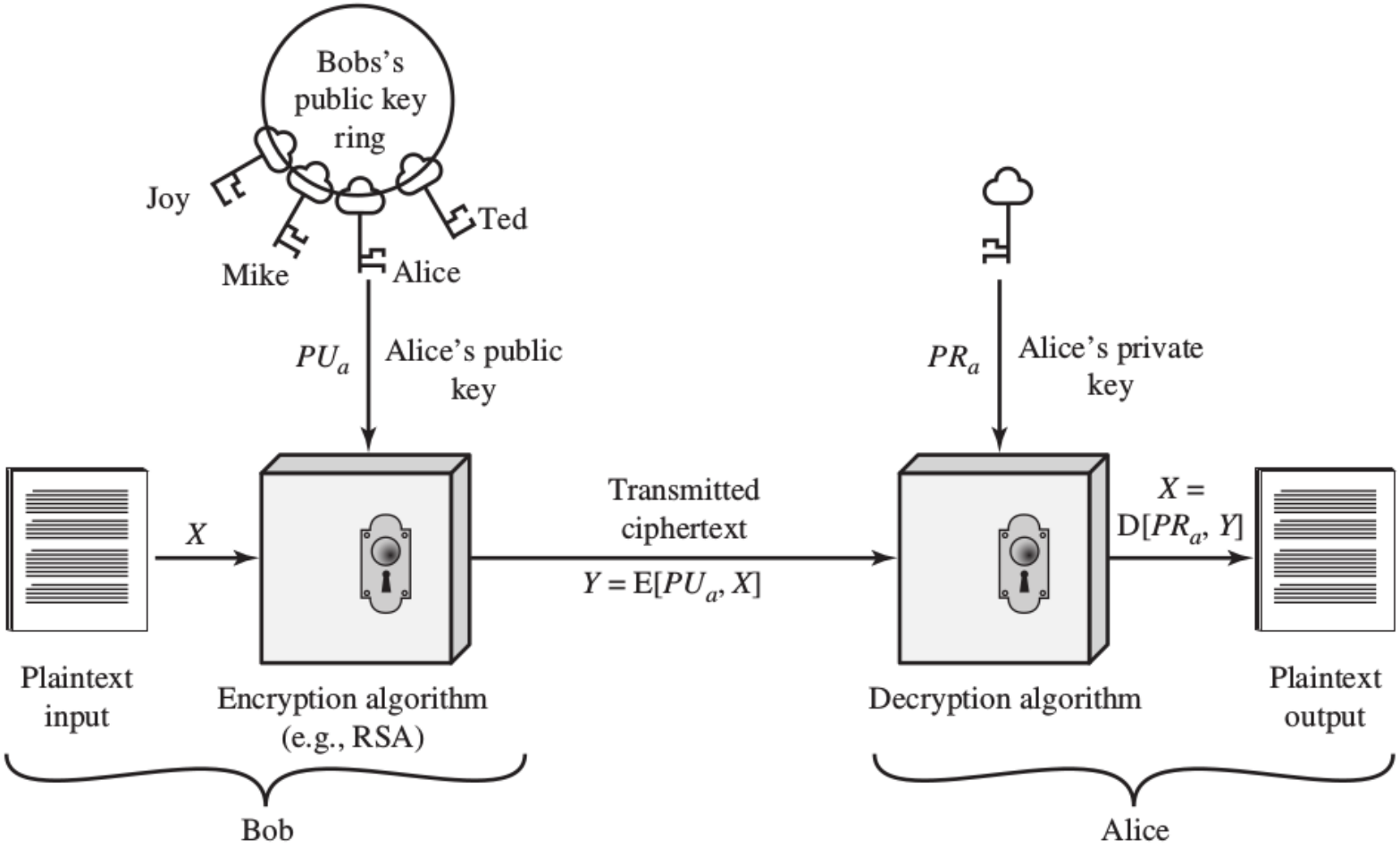
- The 4th property is important if the authentication technique involves the use of a secret value. The secret value itself is not sent; however, if the hash function is not one-way, an attacker can easily discover the secret value.
- The 5th property guarantees that it is impossible to find an alternative message with the same hash value as a given message. This **prevents forgery** when an encrypted hash code is used.
- A strong hash function protects against an attack in which one party generates a message for another party to sign.

  Example: Alice signs the first message and Bob is then able to claim that the second message is authentic.

Public-Key Encryption

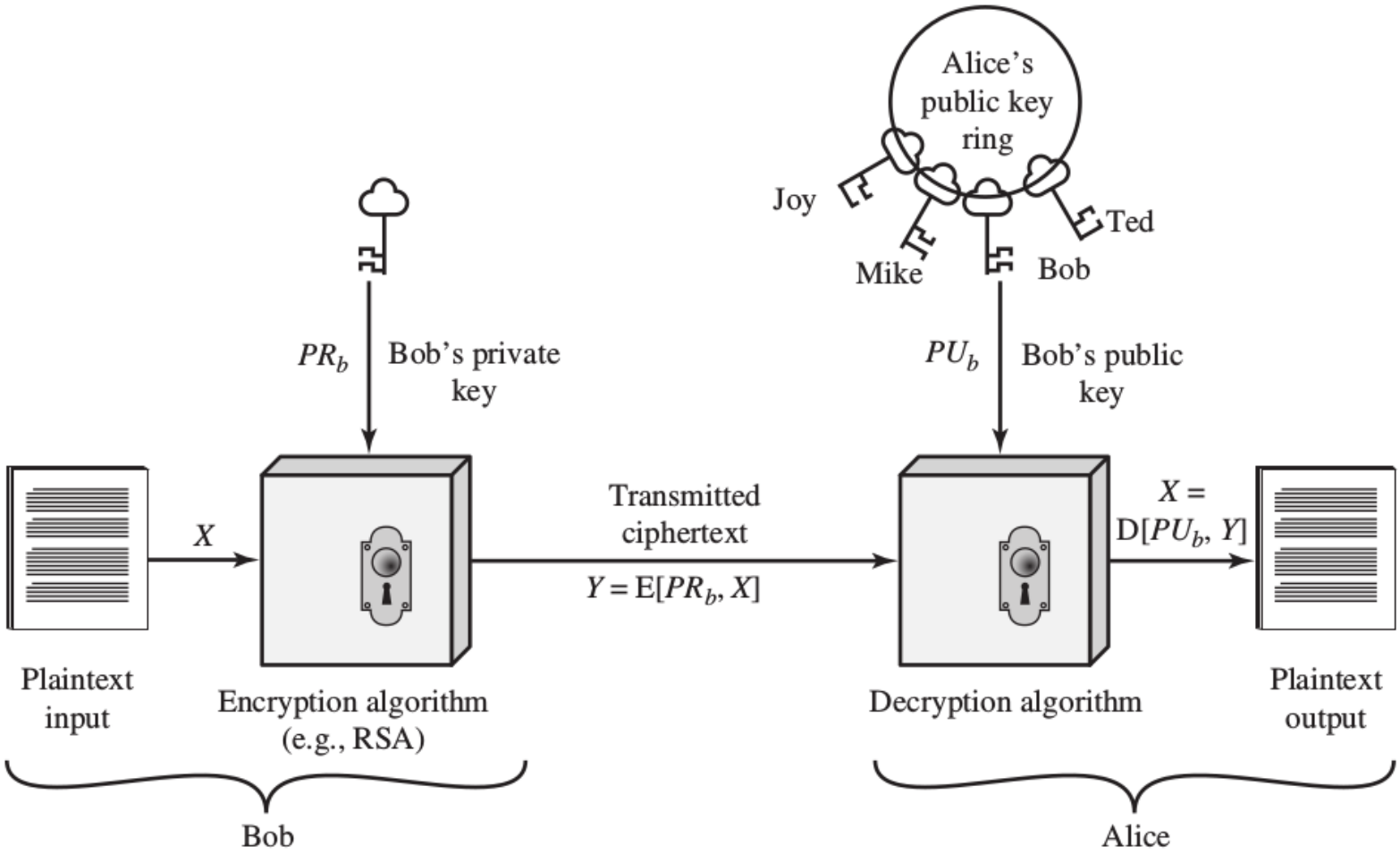
- Public-key algorithms are based on **mathematical functions** rather than on **simple operations on bit patterns**.
- Public-key cryptography is **asymmetric**, involving the use of two separate keys.

Public-Key Encryption



(a) Encryption with public key

Public-Key Encryption



(b) Encryption with private key

Applications of Public-Key Cryptosystems

Algorithm	Digital Signature	Symmetric Key Distribution	Encryption of Secret Keys
RSA	Yes	Yes	Yes
Diffie-Hellman	No	Yes	No
DSS	Yes	No	No
Elliptic Curve	Yes	Yes	Yes

Digital Signature

- Public-key encryption can be used for authentication.
- Suppose that Bob wants to send a message to Alice and needs to be certain that the message is indeed from him.
- Bob can uses a secure hash function, such as **SHA-512**, to generate a hash value for the message and then **encrypts the hash code** with his private key, creating a **digital signature.**

Digital Signature

- When Alice receives the message plus signature, she (1) calculates a hash value for the message;
- (2) decrypts the signature using Bob's public key; and
- (3) compares the calculated hash value to the decrypted hash value. If the two hash values match, Alice is assured that the message must have been signed by Bob.

Public-Key certificates

- The point of public-key encryption is that the **public key** is **public**.
- That is, any participant can send his or her public key to any other participant or broadcast the key to the community at large.
- Is there any weakness in this approach?

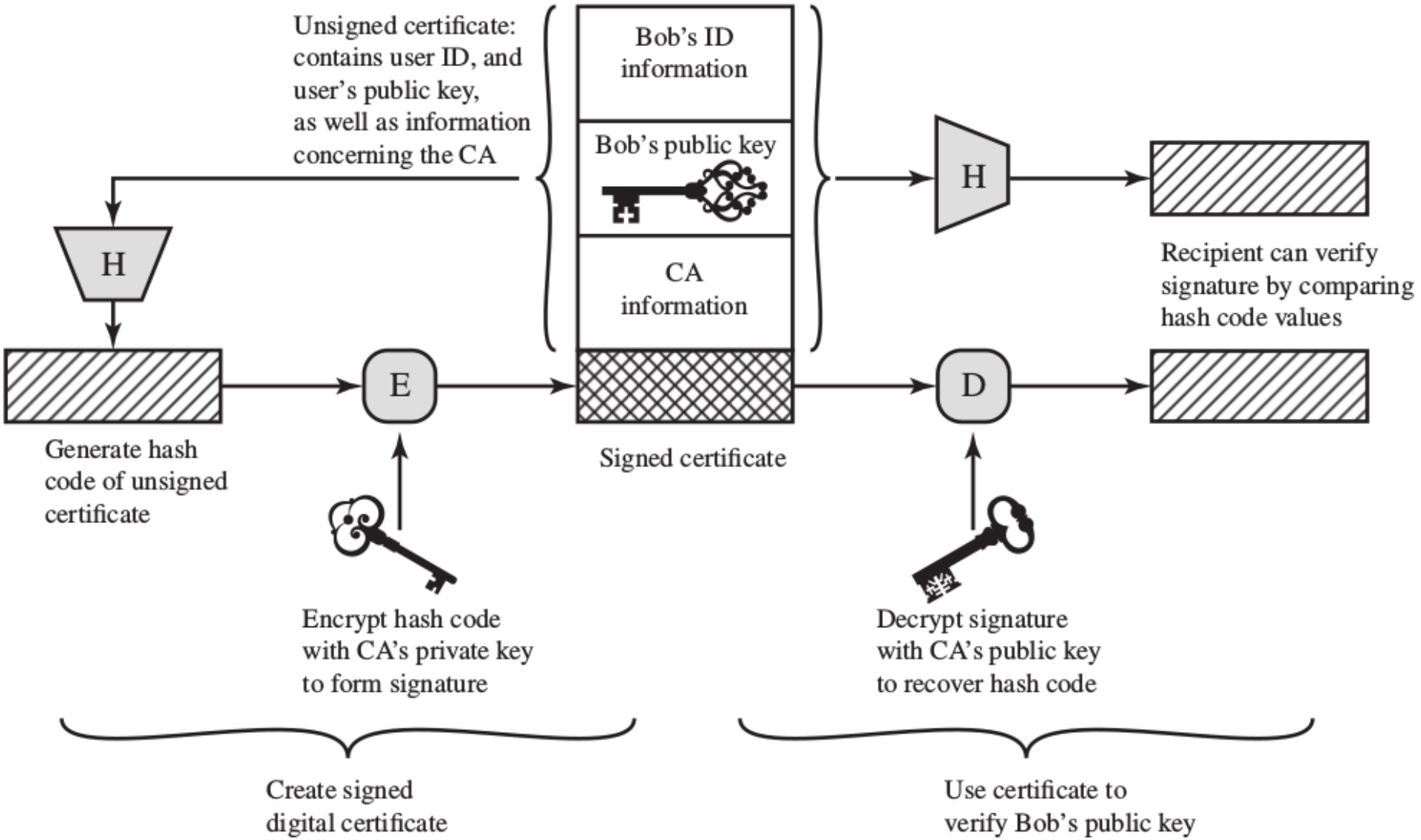
Public-Key certificates

- Anyone can **forge such a public announcement** !!
- That is, some user could pretend to be Bob and send a public key to another participant or broadcast such a public key.
- Until Bob discovers the forgery and alerts other participants, **the forger is able to read all encrypted messages** intended for Bob and can use the forged keys for authentication.

Public-Key certificates

- The solution to this problem is the public-key certificate.
- A certificate consists of a **public key plus a user ID of the key owner**, with the whole block signed by a **trusted third party**.
- The certificate also includes **some information about the third party** plus an indication of the **period of validity** of the certificate.

Public-Key Certificates



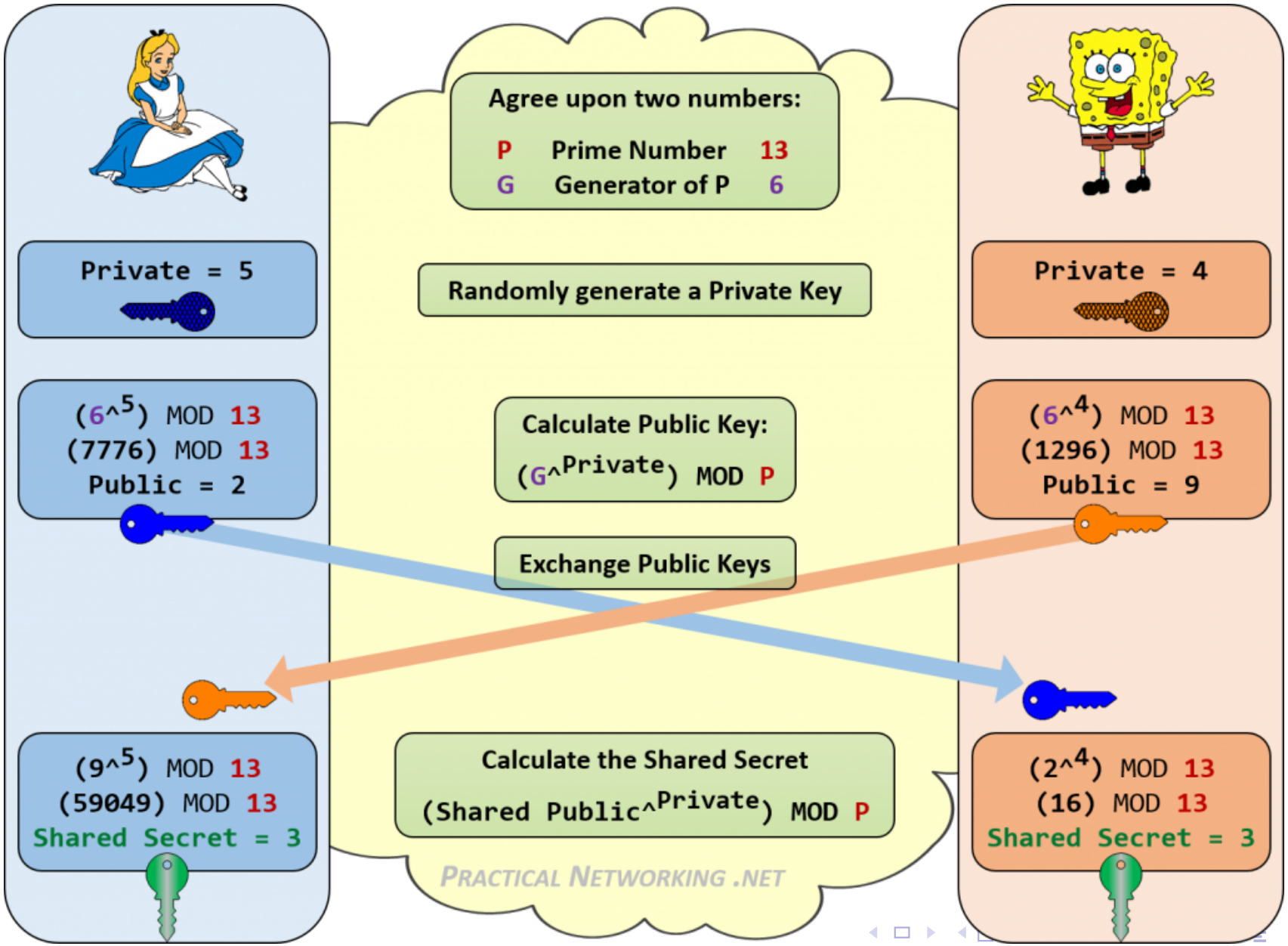
Public-Key certificates

- The scheme universally accepted for formatting public-key certificates is the X.509 standard.
- X.509 certificates are used in most network security applications, including IP Security (IPsec), Transport Layer Security (TLS), Secure Shell (SSH), and Secure/Multipurpose Internet Mail Extension (S/MIME).

Symmetric Key Exchange Using Public-Key Encryption

- Public-key encryption can be used to share symmetric keys.

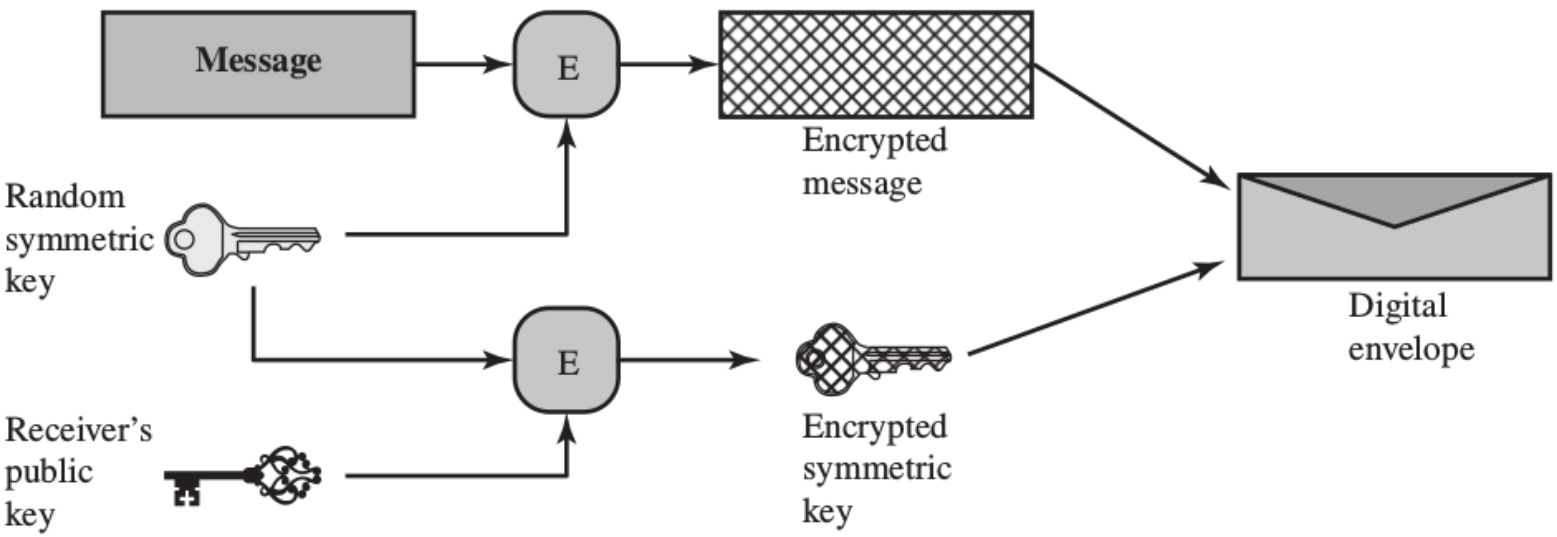
Diffie-Hellman Key Exchange



Diffie-Hellman Key Exchange

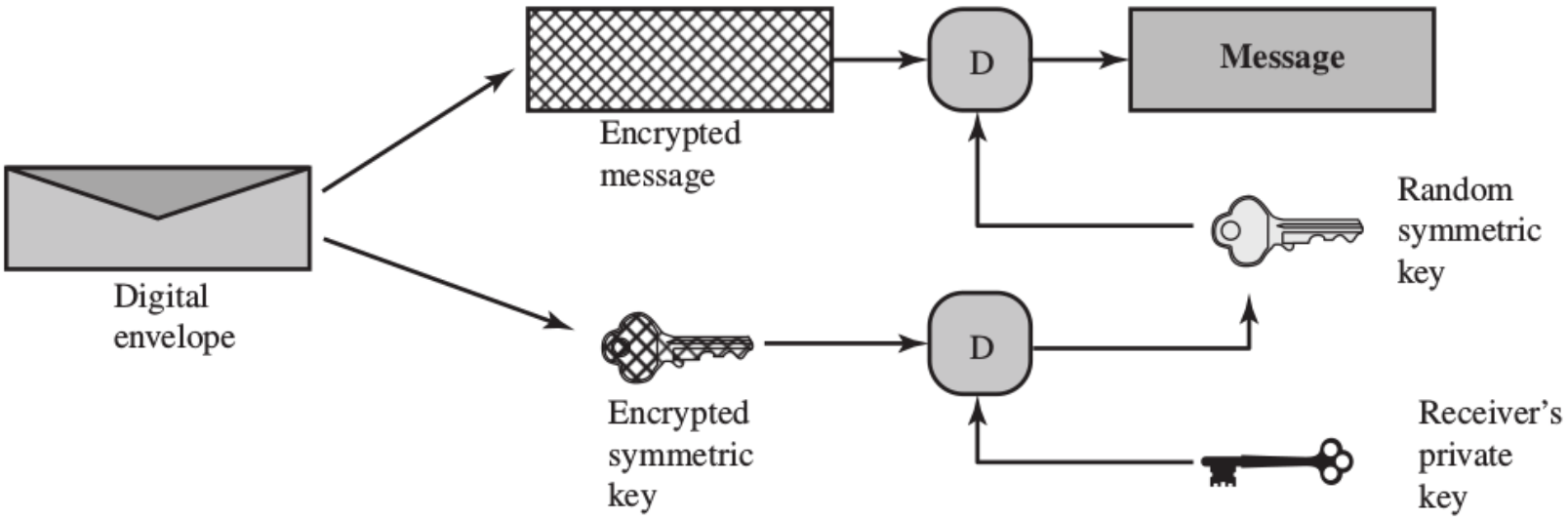
- Diffie-Hellman provides no authentication of the two communicating partners. There are variations to Diffie-Hellman that overcome this problem.

Digital Envelopes



(a) Creation of a digital envelope

Digital Envelopes



(b) Opening a digital envelope

The End