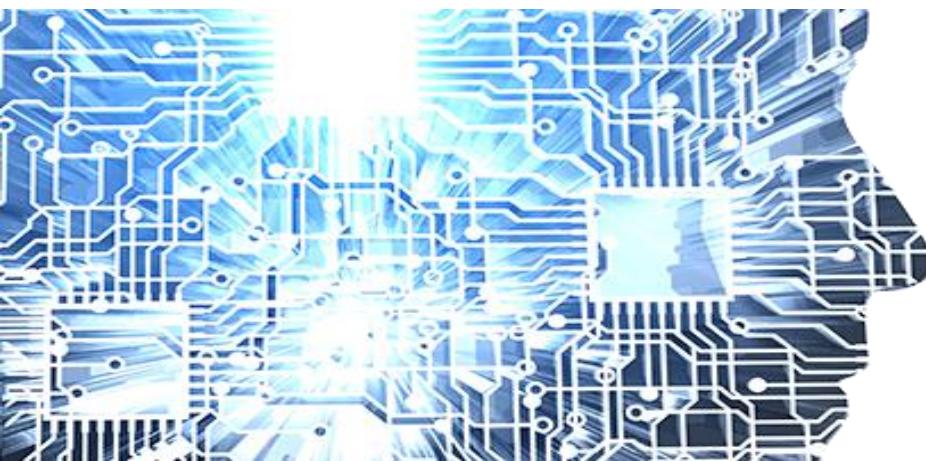


CHƯƠNG 4

MẬT MÃ

VÀ

XÁC THỰC THÔNG TIN



**Bộ môn: Tin học quản lý
Khoa Thống kê – Tin học
Đại học Kinh Tế - Đại học Đà Nẵng**



NỘI DUNG CHƯƠNG 4

1. Tổng quan về mã hóa thông tin
2. Mã hóa đối xứng
3. Mã hóa bất đối xứng
4. Chữ ký số
5. Chứng chỉ số
6. Hàm băm

1. Tổng quan về mã hóa thông tin

1. Mật mã học
2. Một số thuật ngữ
3. Một số vấn đề chính trong an toàn thông tin
4. Các loại mã hóa thông thường

1.1. Mật mã học

- Mật mã (*Cryptography*) là ngành khoa học nghiên cứu các kỹ thuật toán học nhằm cung cấp các dịch vụ bảo vệ thông tin.



1.2. Một số thuật ngữ

Encryption: Mã hóa

Cryptography: Mật mã (*phương pháp mã hóa*)

Cryptanalysis: Phá mã (*giải mã không cần khóa*)

Cryptology: mật mã học (*bao gồm mật mã và phá mã, thám mã*)

Cryptosystem: Hệ thống mã hóa

Key: Khóa (*Thông tin chỉ dành cho người gửi hoặc người nhận*)

1.2. Một số thuật ngữ

Secret key: Khóa bí mật

Symmetric key: Khóa đối xứng

Public key: Khóa công cộng

Plaintext: Thông điệp ban đầu

Ciphertext: Thông điệp đã được mã hóa

Cipher: Thuật toán chuyển đổi từ *plaintext* thành *ciphertext*

Encrypt: Mã hóa (chuyển từ *plaintext* thành *ciphertext*)

Decrypt: Giải mã (chuyển đổi từ *ciphertext* thành *plaintext*)

1.3. Một số vấn đề chính trong an toàn thông tin

- ❖ **Bảo mật dữ liệu (Secrecy)**: đảm bảo dữ liệu được giữ bí mật.
- ❖ **Toàn vẹn thông tin (Integrity)**: bảo đảm tính toàn vẹn dữ liệu trong liên lạc hoặc giúp phát hiện rằng thông tin đã bị sửa đổi.
- ❖ **Xác thực (Authentication)**: xác thực các đối tác trong liên lạc và xác thực nội dung dữ liệu trong liên lạc.

1.3. Một số vấn đề chính trong an toàn thông tin

- ❖ Chống thoái thác trách nhiệm (*Non-repudiation*): đảm bảo một đối tác bất kỳ trong hệ thống không thể từ chối trách nhiệm về hành động mà mình đã thực hiện
- ❖ Tính riêng tư (*Privacy*): giữ bí mật thông tin về định danh, hành động, vị trí...

1.4. Các loại mã hóa thông thường

□ Mã hóa Caesar

Mục tiêu của mật mã cổ điển: Truyền thông an toàn

Julius Ceasar (100-44 BC)



DWWDFN DW GDZQ



ATTACK AT DAWN

Giải pháp: Mã hóa thông điệp!

ATTACK AT DAWN

Giải mã thông điệp đã mã hóa!

1.4. Các loại mã hóa thông thường

□ Mã hóa Caesar

Mục tiêu của mật mã cổ điển: Truyền thông an toàn

Julius Ceasar (100-44 BC)



DWWDFN DW GDZQ



ATTACK AT DAWN

Giải pháp: Mã hóa thông điệp!

ATTACK AT DAWN

Giải mã thông điệp đã mã hóa!

Có ba đối tượng cần được quan tâm:

(1) Người gửi, (2) Người nhận và (3) Người nghe lén

1.4. Các loại mã hóa thông thường

☐ Mã hóa Caesar

- ❖ **Secret key:** là một số ngẫu nhiên trong {1,...,26}, số 3

Julius Ceasar (100-44 BC)



DWWDFN DW GDZQ



Thông điệp: ATTACK AT DAWN

Mã hóa: Key: + 3 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

Ciphertext: DWWDFN DW GDZQ

1.4. Các loại mã hóa thông thường

☐ Mã hóa Caesar

- ❖ **Secret key:** là một số ngẫu nhiên trong {1,...,26}, số 3

Julius Ceasar (100-44 BC)



DWWDFN DW GDZQ



Giải mã:

Ciphertext: DWWDFN DW GDZQ

Key: - 3

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

Thông điệp: ATTACK AT DAWN

1.4. Các loại mã hóa thông thường

❑ Mã hóa Caesar

- ❖ Mã hóa Caesar là phương pháp dịch chuyển từng ký tự theo xoay vòng 3 ký tự.



Hãy mã hóa thông điệp sau:

ATTACK AT DAWN

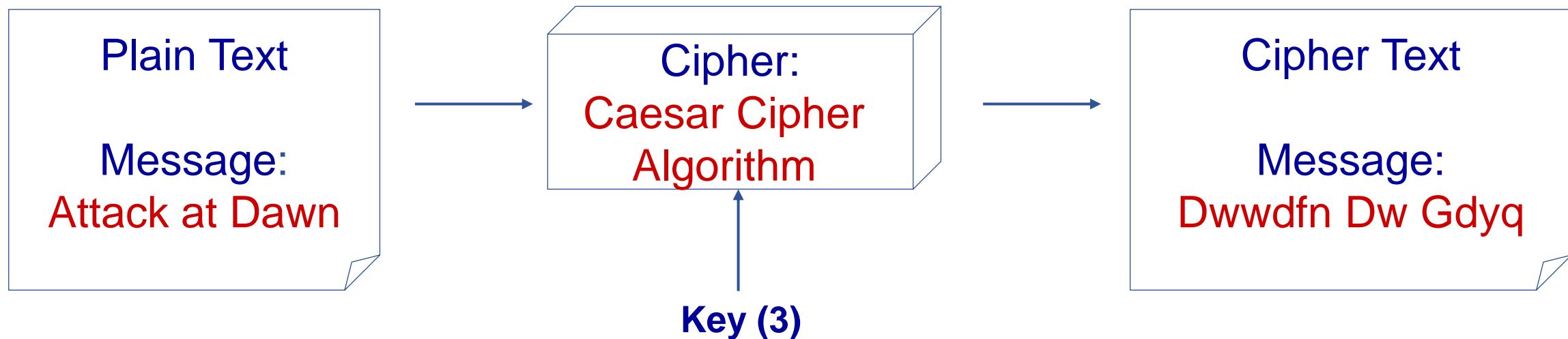
Kết quả mã hóa:

DWWDFN DW GDZQ

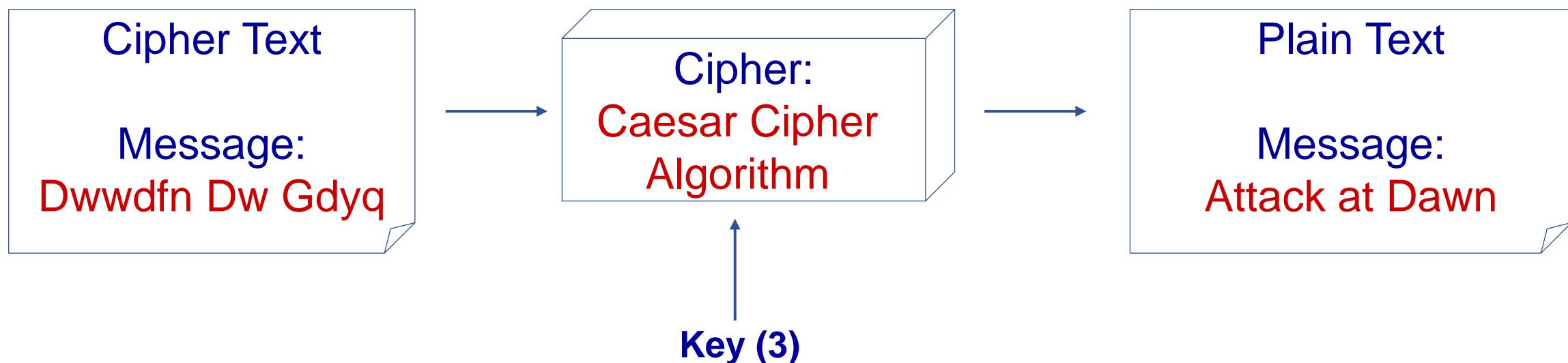
1.4. Các loại mã hóa thông thường

❑ Mã hóa Caesar

Mã hóa

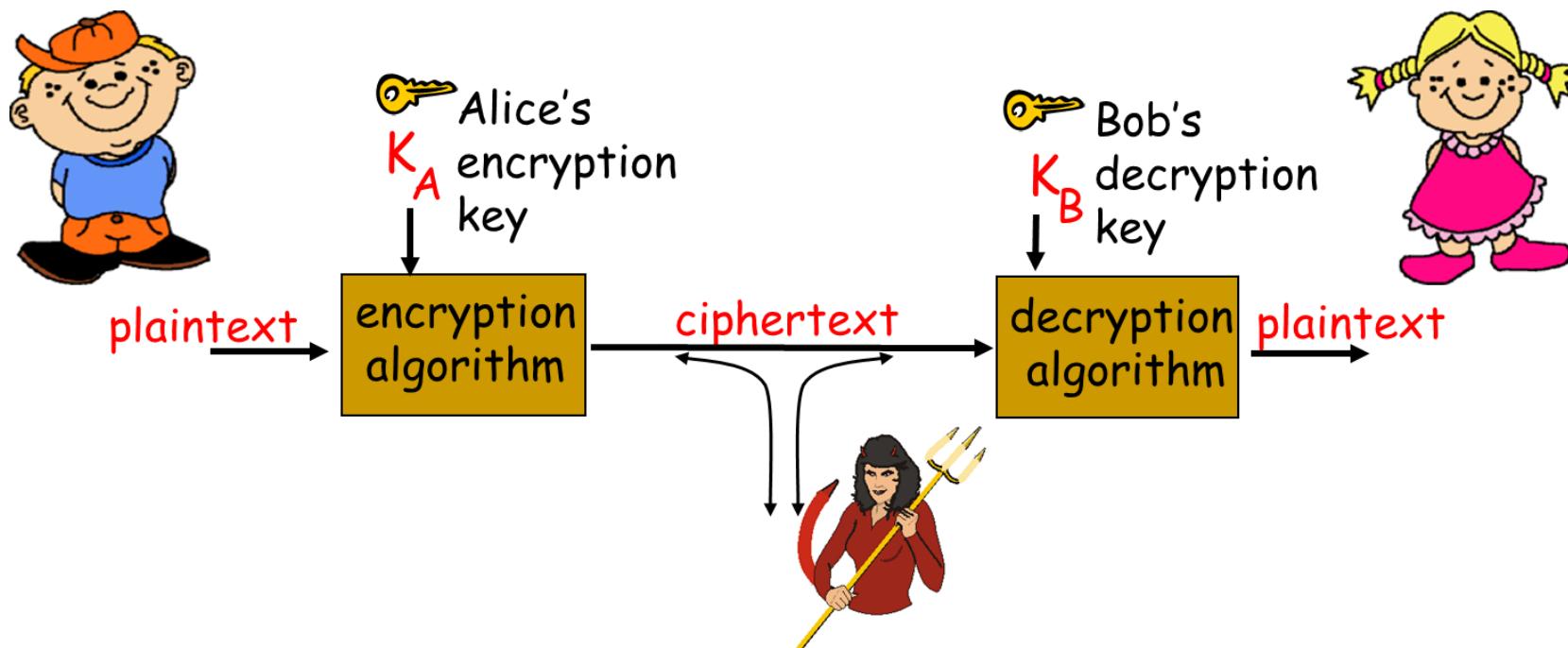


Giải mã



1.4. Các loại mã hóa thông thường

□ Mật mã hiện đại



- ❖ **Mã hóa đối xứng:** khóa của người gửi và người nhận giống nhau.
- ❖ **Mã hóa khóa công cộng:** khóa mã hóa là công cộng (*public*), khóa giải mã là bí mật (*secret/ private*).

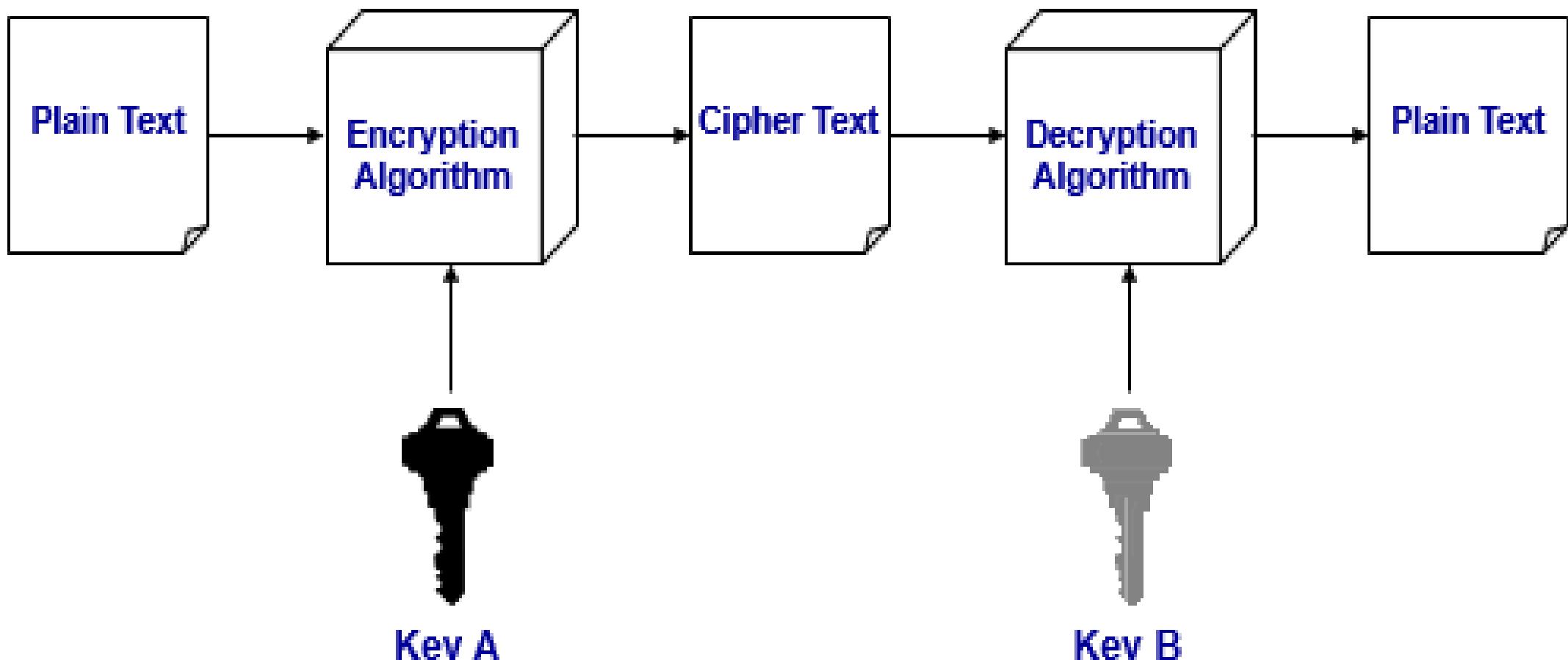
1.4. Các loại mã hóa thông thường

□ **Mã hóa (Cipher)**

- ❖ **Cipher** là phương pháp mã hóa thông điệp, chuyển đổi thông điệp ban đầu (*Plaintext*) trở thành thông điệp đã được mã hóa (*Ciphertext*).
- ❖ **Khóa (Key)** là bí mật và là đầu vào cho thuật toán có các đặc điểm sau:
 - ❖ Khóa là chuỗi số hoặc ký tự; Nếu sử dụng cùng một khóa để mã hóa và giải mã thì được gọi là đối xứng.
 - ❖ Nếu sử dụng các khóa khác nhau để mã hóa và giải mã thì được gọi là bất đối xứng.

1.4. Các loại mã hóa thông thường

□ Mã hóa (Cipher)



1.4. Các loại mã hóa thông thường

□ Mã hóa đối xứng

- ❖ Là phương pháp sử dụng cùng khóa (Key) để thực hiện việc mã hóa và giải mã.
 - Ví dụ: Caesar Cipher
- ❖ Các kiểu mã hóa:
 - Mã hóa khối (*Block Ciphers*)
 - Mã hóa dữ liệu theo từng khối tại một thời điểm (*thường là 64 bit hoặc 128 bit*)
 - Được sử dụng cho các thông điệp đơn.
 - Mã hóa dòng (*Stream Ciphers*)
 - Mã hóa một bit hoặc một byte tại một thời điểm.
 - Được sử dụng nếu dữ liệu là một luồng thông tin liên tục.

1.4. Các loại mã hóa thông thường

□ Mã hóa đối xứng

- ❖ Độ mạnh thuật toán được xác định bằng **kích thước của khóa** → Khóa càng dài thì càng khó bẻ khóa.
 - Chiều dài của khóa được mô tả bằng các bit: Thông thường kích thước của khóa từ 48 bit đến 448 bit.
 - Tập khóa có thể có để mã hóa được gọi là **không gian khóa**
 - Khóa 40 bit có thể có 2^{40} khóa.
 - Khóa 128 bit có thể có 2^{128} khóa.
 - Mỗi bit được thêm vào khóa thì tăng gấp đôi tính bảo mật.
- ❖ Để bẻ khóa, hacker phải sử dụng **brute-force**
 - Một siêu máy tính có thể crack được một khóa 56 bit trong vòng 24 giờ.
 - Mất 2^{72} lần thời gian để crack khóa 128-bit.

1.4. Các loại mã hóa thông thường

❑ Mã hóa thay thế

❖ Mã hóa ký tự đơn (Monoalphabetic Cipher)

- Ký tự bất kỳ có thể được thay thế bằng ký tự khác
- Mỗi ký tự phải có một thay thế duy nhất.

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

1.4. Các loại mã hóa thông thường

❑ Mã hóa thay thế

❖ Mã hóa ký tự đơn (Monoalphabetic Cipher)

- Có $26!$ hoán vị các ký tự (~ 1026)
- Tiếp cận Brute Force sẽ bị mất quá nhiều thời gian
- Phân tích thống kê có thể tạo ra một phương án khả dĩ để phá khóa.

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

1.4. Các loại mã hóa thông thường

❑ Mã hóa thay thế

❖ Mã hóa Ceasar đa ký tự (*Polyalphabetic Ceasar Cipher*)

- Được phát triển bởi Blaise de Vigenere, phương pháp này cũng được gọi là Vigenere cipher.
- Sử dụng một dãy các mã hóa đơn ký tự song song.
 - Ví dụ: C₁, C₂, C₂, C₁, C₂, C₂, ...

Plain Text 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



C₁(k=6) 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

C₂(k=20) 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

Message:

Bob, I love you.
Alice

Cipher:
Monoalphabetic
Cipher

Encrypted
Message:

Hiv, O fiby suo.
Urcwk

Key

1.4. Các loại mã hóa thông thường

❑ Mã hóa chuyển vị

❖ Chuyển vị theo cột (*Columnar Transposition*)

- Sắp xếp lại thông điệp ban đầu theo các cột.
- Ví dụ về cách chuyển đổi các ký tự:

Nếu các chữ cái không phải là bội số của kích thước chuyển vị thì thêm vào một số ký tự không thường xuyên, chẳng hạn x hoặc z.

Plain Text

T H I S I
S A M E S
S A G E T
O S H O W
H O W A C
O L U M N
A R T R A
N S P O S
I T I O N
W O R K S

Cipher Text

T S S O H
O A N I W
H A A S O
L R S T O
I M G H W
U T P I R
S E E O A
M R O O K
I S T W C
N A S N S

1.4. Các loại mã hóa thông thường

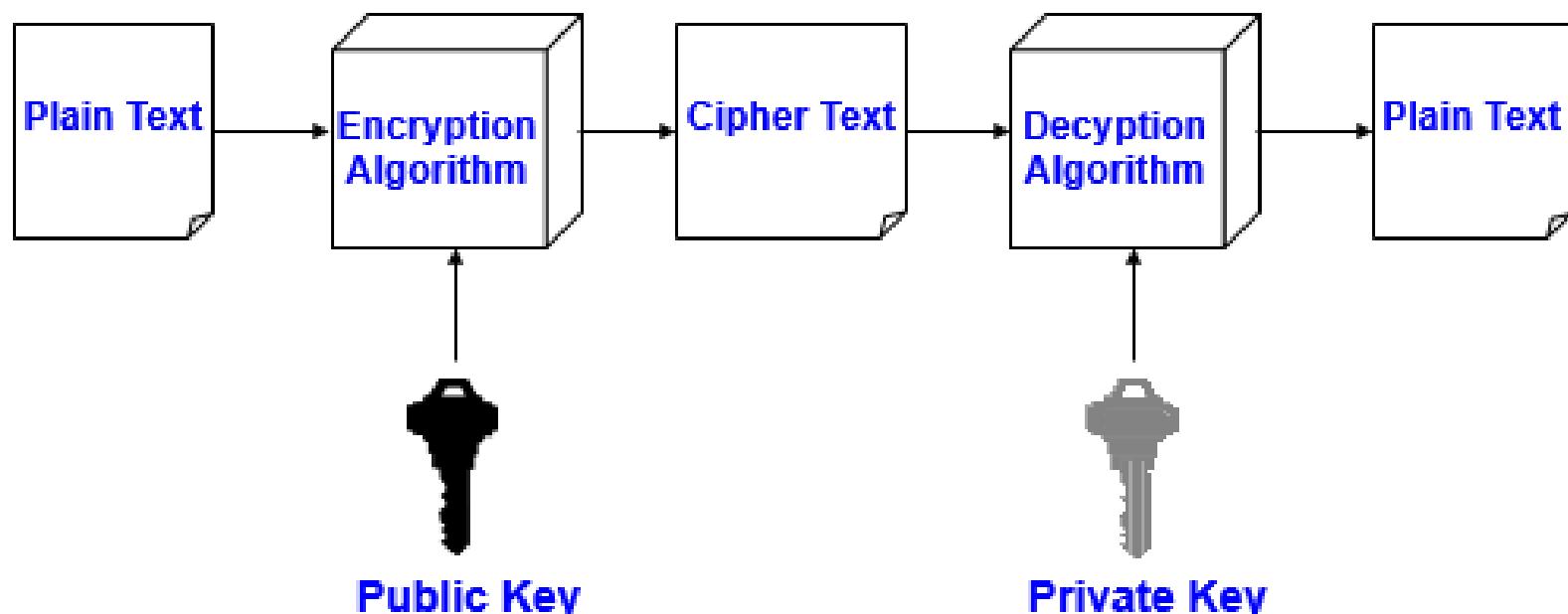
❑ Mã hóa bất đối xứng

❖ Sử dụng một cặp khóa để mã hóa

- Khóa công cộng (Public key)
- Khóa riêng tư (Private key)

❖ Thông điệp được mã hóa sử dụng khóa công cộng, việc giải mã chỉ dùng khóa riêng tư.

- Không cần phải bí mật truyền khóa để giải mã
- Mọi thực thể có thể tạo một cặp khóa và phát hành khóa công cộng.



1.4. Các loại mã hóa thông thường

Mã hóa bất đối xứng

- ❖ Hai thuật toán phổ biến nhất là RSA và El Gamal
- ❖ **Thuật toán RSA**
 - Được phát triển bởi Ron Rivest, Adi Shamir, Len Adelman
 - Cả hai khóa (khóa công cộng và khóa riêng tư) có thể hoán đổi cho nhau.
 - Kích thước khóa thay đổi (512, 1024, hoặc 2048 bit)
 - Thuật toán khóa công khai phổ biến nhất
- ❖ **Thuật toán El Gamal**
 - Được phát triển bởi Taher ElGamal
 - Kích thước khóa (512 hoặc 1024 bit)
 - Ít phổ biến hơn RSA

1.4. Các loại mã hóa thông thường

□ Thuật toán RSA

- ❖ Chọn hai số nguyên tố lớn p và q
- ❖ Tính $n = p * q$ và $\phi = (p-1) * (q-1)$
- ❖ Chọn một số e nhỏ hơn n là số nguyên tố cùng nhau với z.
- ❖ Tìm số d sao cho $(e * d - 1)$ chia hết cho z
- ❖ **Khóa được tạo ra sử dụng n, e, d**
 - Khóa công cộng là (n, e)
 - Khóa riêng tư là (n, d)
- ❖ **Mã hóa $c = m^e \text{ mod } n$**
 - m là thông điệp gốc (plain text)
 - C là thông điệp mã hóa (cipher text)
- ❖ **Giải mã $m = c^d \text{ mod } n$**
- ❖ Khóa công cộng được chia sẻ, khóa riêng tư được ẩn giấu

1.4. Các loại mã hóa thông thường

□ Thuật toán RSA

- ❖ Chọn $p = 5$ và $q = 7$
- ❖ Tính $n = p * q = 5 * 7 = 35$ và $\phi(n) = (p-1)(q-1) = 4 * 6 = 24$
- ❖ Chọn $e = 5$
- ❖ số $d = 29$, với $(e * d - 1)$ chia hết cho $\phi(n)$
- ❖ Các khóa được tạo ra
 - Khóa công cộng là (n, e)
 - Khóa riêng tư là (n, d)

1.4. Các loại mã hóa thông thường

□ Thuật toán RSA

- ❖ Mã hóa từ “love” sử dụng $c = m^e \text{ mod } n$
 - Giả sử các ký tự có giá trị từ 1 đến 26

Plain Text	Numeric Representation	m^e	Cipher Text ($c = m^e \text{ mod } n$)
l	12	248832	17
o	15	759375	15
v	22	5153632	22
e	5	3125	10

1.4. Các loại mã hóa thông thường

□ Thuật toán RSA

➤ Giải mã từ “love” sử dụng $m = c^d \bmod n$

- $n=35, d = 29$

Cipher Text	c^d	$(m = c^d \bmod n)$	Plain Text
17	481968572106750915091411825223072000	12	I
15	12783403948858939111232757568359400	15	o
22	8526433190865377019561944997211000000 0	22	v
10	1000	5	e

1.4. Các loại mã hóa thông thường

□ Mã hóa Elgamal

- Hệ mật Elgamal hình thành trên cơ sở bài toán logarithm rời rạc được đề xuất năm 1984
- **Thuật toán tạo khóa:**
 1. Chọn số nguyên tố đủ lớn p có chiều dài là k sao cho bài toán logarithm trong Z_p có độ phức tạp lớn
 2. Chọn $e_1 \in Z_p^*$ là phần tử nguyên thủy
 3. Chọn d là số ngẫu nhiên sao cho $1 < d < p$
 4. Tính $e_2 = e_1^d \text{ mod } p$

Kết quả ta được: Khóa bí mật là d , khóa công khai (e_1, e_2, p)

1.4. Các loại mã hóa thông thường

□ Mã hóa Elgamal

➤ Thuật toán mã hóa và giải mã:

Cho văn bản gốc M là một số nguyên

- **Quá trình mã hóa:**

Chọn một giá trị ngẫu nhiên r .

Tính các giá trị (C_1, C_2) như sau:

$$C_1 = e_1^r \bmod p$$

$$C_2 = (e_2^r \times M) \bmod p$$

Văn bản (C_1, C_2) được gửi đến người nhận

- **Quá trình giải mã:**

$$M = [C_2 \times (C_1^d)^{-1}] \bmod p$$

1.4. Các loại mã hóa thông thường

□ Mã hóa Elgamal

➤ **Ví dụ:** Cho văn bản gốc $M = 7$, số nguyên tố $p = 11$
Chọn giá trị cơ sở $e_1 = 2$. Chọn khóa riêng $d = 3$.

Tính $e_2 = e_1^d = 2^3 = 8$

Khóa công khai: (2,8,11); Khóa riêng d=3.

Quá trình mã hóa:

Chọn giá trị ngẫu nhiên $r=4$.

$$C_1 = e_1^r \bmod p = 2^4 \bmod 11 = 5$$

$$C_2 = (e_2^r \times M) \bmod p = (8^4 \times 7) \bmod 11 = 6$$

Văn bản mã hóa $C=(5,6)$ được gửi đến người nhận

Quá trình giải mã:

$$M = [C_2 \times (C_1^d)^{-1}] \bmod p = 6 \times (5^3)^{-1} \bmod 11 = 7$$

1.4. Các loại mã hóa thông thường

So sánh Mã hóa đối xứng và bất đối xứng

Mã hóa đối xứng

Tốc độ xử lý nhanh

Mã khóa ngắn

Khó trao đổi
mã khóa

Mã hóa bất đối xứng

Tốc độ xử lý chậm

Mã khóa dài

Trao đổi mã khóa
dễ dàng

1.4. Các loại mã hóa thông thường

Hệ thống mã hóa tin cậy

❖ Phải dựa trên một nền tảng toán học

→ Các thuật toán mã hóa tốt có nguồn gốc từ những nguyên lý vững chắc.

❖ Phải được phân tích bởi các chuyên gia

→ Vì người viết thuật toán không dự liệu hết những tấn công.

❖ Phải bền vững theo thời gian

- Theo thời gian, con người sẽ đánh giá được các nền tảng toán học của một thuật toán và các thuật toán được xây dựng trên các nền tảng đó.
- Các lỗ hổng của thuật toán phải được phát hiện sớm khi phát hành.

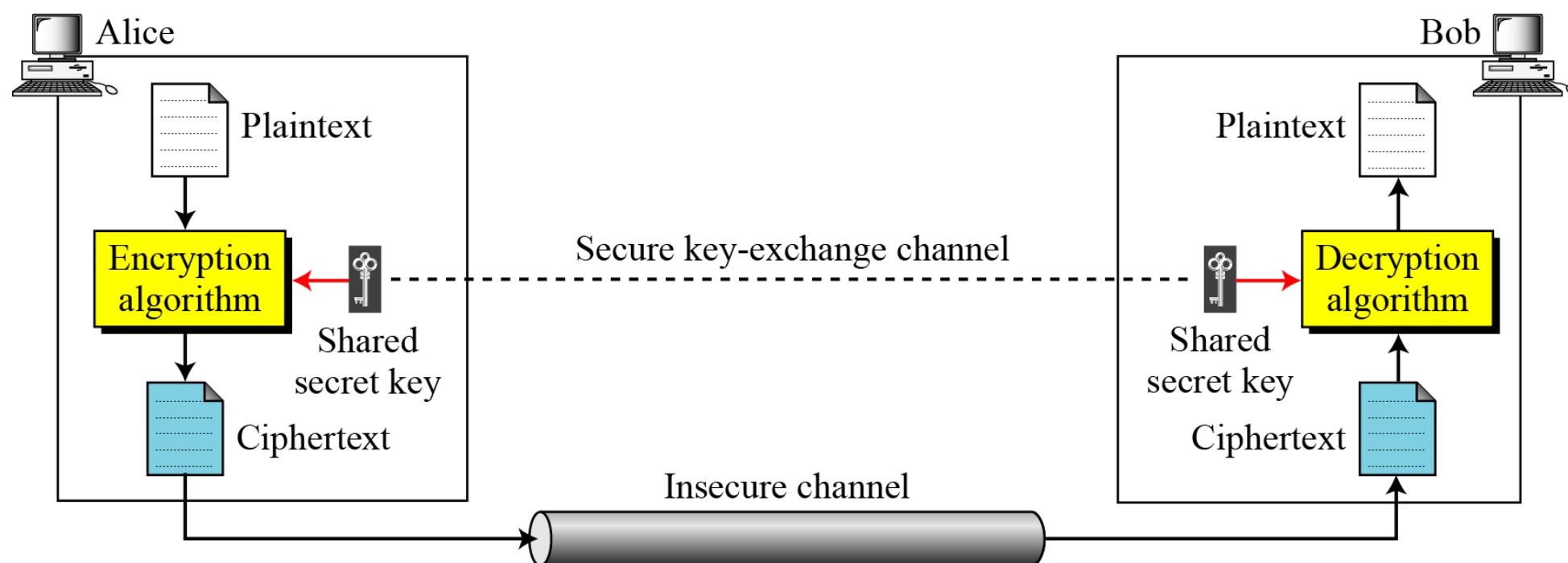
Trường \mathbb{Z}_m (modulo)

2. Mã hóa đối xứng

1. Khái quát
2. Phương pháp mã hóa dịch chuyển
3. Mã hóa chuyển vị
4. Mã hóa tích
5. Quy trình mã hóa

2.1. Khái quát

- ❖ Thông điệp ban đầu từ Alice gửi đến Bob được gọi là thông điệp gốc (*Plaintext*). Để tạo bản mã (*Ciphertext*), Alice sử dụng thuật toán mã hóa và tạo khóa bí mật K; sau đó gửi qua kênh bảo mật cho Bob.
- ❖ Bob sử dụng thuật toán giải mã và khóa bí mật K để khôi phục lại thông điệp gốc.



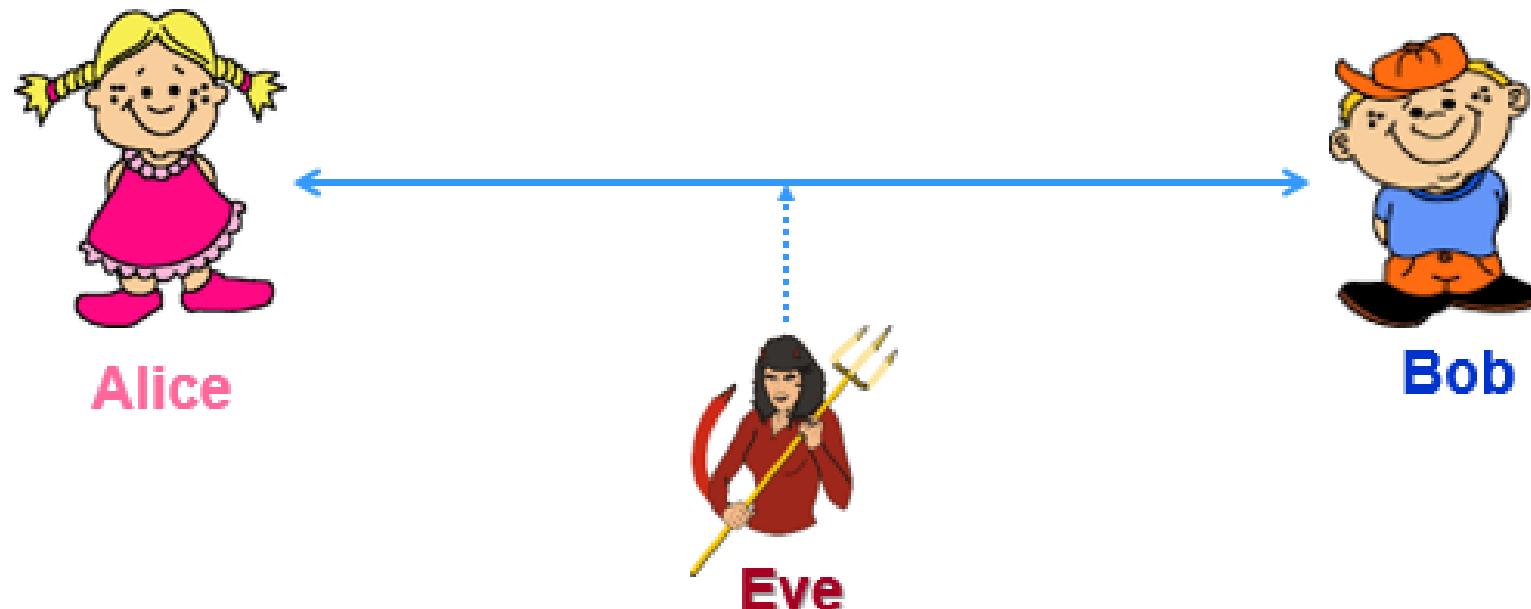
2.1. Khái quát

- ❖ Gọi P là thông điệp gốc, C là bản mã và K là khóa, ta có:
 - $C = E_k(P)$ và $P = D_k(C)$, với $D_k(E_k(x)) = E_k(D_k(x)) = x$
 - E, D là tập các luật mã hóa và giải mã.
- ❖ Giả sử Bob giải mã được P_1 , khi đó $P_1 = P$ vì:
 - Alice: $C = E_k(P)$
 - Bob: $P_1 = D_k(C) = D_k(E_k(P)) = P$

2.1. Khái quát

❑ NGUYÊN LÝ KERCKHOFF

- ❖ Theo nguyên lý Kerckhoff, phải luôn giả định rằng có người “nghe lén” (Eve) biết được thuật toán mã hóa và giải mã. Khi đó, việc chống lại sự tấn công không phải chỉ dựa trên việc bảo mật khóa K.



Alice và Bob trao đổi với nhau trong khi Eve tìm cách “nghe lén”.

2.1. Khái quát

❑ Các phương pháp truyền thống

❖ Các phương pháp truyền thống sử dụng:

- Phép thay thế (*substitution*): thay thế 1 từ/ký tự bằng 1 từ/ký tự khác.
- Phép thay đổi vị trí (*transposition*): các ký tự được thay đổi vị trí.

❖ Việc thay thế/thay đổi vị trí có thể được thực hiện:

- Đơn ký tự (mono-alphabetic)
- Đa ký tự (poly-alphabetic)

2.2. Phương pháp mã hóa dịch chuyển

❑ Shift Cipher:

- ❖ Một trong những phương pháp lâu đời nhất được sử dụng để mã hóa
- ❖ Thông điệp được mã hóa bằng cách **dịch chuyển xoay** vòng từng ký tự đi *k* vị trí trong bảng chữ cái.
- ❖ Trường hợp với *k=3* gọi là phương pháp *mã hóa Caesar*.

2.2. Phương pháp mã hóa dịch chuyển

- ❖ Phương pháp đơn giản.
- ❖ Thao tác xử lý mã hóa và giải mã được thực hiện nhanh chóng.
- ❖ Không gian khóa $K = \{0, 1, 2, \dots, n-1\} = Z_n$
- ❖ Dễ bị phá vỡ bằng cách thử mọi khả năng khóa k .

Cho $P = C = K = Z_n$

Với mỗi khóa $k \in K$, định nghĩa:

$$e_k(x) = (x + k) \bmod n \text{ và } d_k(y) = (y - k) \bmod n \text{ với } x, y \in Z_n$$

$$E = \{e_k, k \in K\} \text{ và } D = \{d_k, k \in K\}$$

2.2. Phương pháp mã hóa dịch chuyển

❖ Ví dụ:

- Mã hóa một thông điệp được biểu diễn bằng các chữ cái từ A đến Z (26 chữ cái), ta sử dụng Z_{26} .
- Thông điệp được mã hóa sẽ không an toàn và có thể dễ dàng bị giải mã bằng cách thử lần lượt 26 giá trị khóa k .
- Tính trung bình, thông điệp đã được mã hóa có thể bị giải mã sau khoảng $26/2 = 13$ lần thử khóa.

Cho $P = C = K = \mathbf{Z}_n$

Với mỗi khóa $k \in K$, định nghĩa:

$$e_k(x) = (x + k) \bmod n \text{ và } d_k(y) = (y - k) \bmod n \text{ với } x, y \in \mathbf{Z}_n$$

$$E = \{e_k, k \in K\} \text{ và } D = \{d_k, k \in K\}$$

2.2. Phương pháp mã hóa dịch chuyển

❖ Cho bản mã

JBCRCLQRWCRVNB**JENBWRWN**

❖ Lần lượt thử các khóa $k = 0, 1, 2, \dots, 25$

jbcrcrlqrwcrvnbjenbwrwn

iabqbkpqvqbqumaidmavqvm

hzapajopuaptlzhclzupul

gyzozinotzoskygbkytotk

fxynyhmnsynrjxfajxsnsj

ewxmxmlrxmqiweziwrnri

dvwlwfklqlqlphvdyhvqlqh

cuvkvejkpvkogucxgupkpg

btujudijoujnftbwftojof

a**stitch****intime****saves**nine ← $k = 9$

2.2. Phương pháp Mã hóa thay thế (substitution)

- ❖ Mã hóa thay thế được thực hiện bằng cách thay thế một ký tự bằng một ký tự khác.
- ❖ Phương pháp mã hóa thay thế được phân loại là mã hóa đơn ký tự (*monoalphabetic*) và mã hóa đa ký tự (*polyalphabetic*).
- ❖ Đối với phương pháp mã hóa đơn ký tự, mối quan hệ là 1-1 (*one-to-one*) giữa một ký tự trong plaintext và một ký tự trong ciphertext.
- ❖ Ví dụ minh họa cho thông điệp gốc và thông điệp đã được mã hóa:

Plaintext: hello

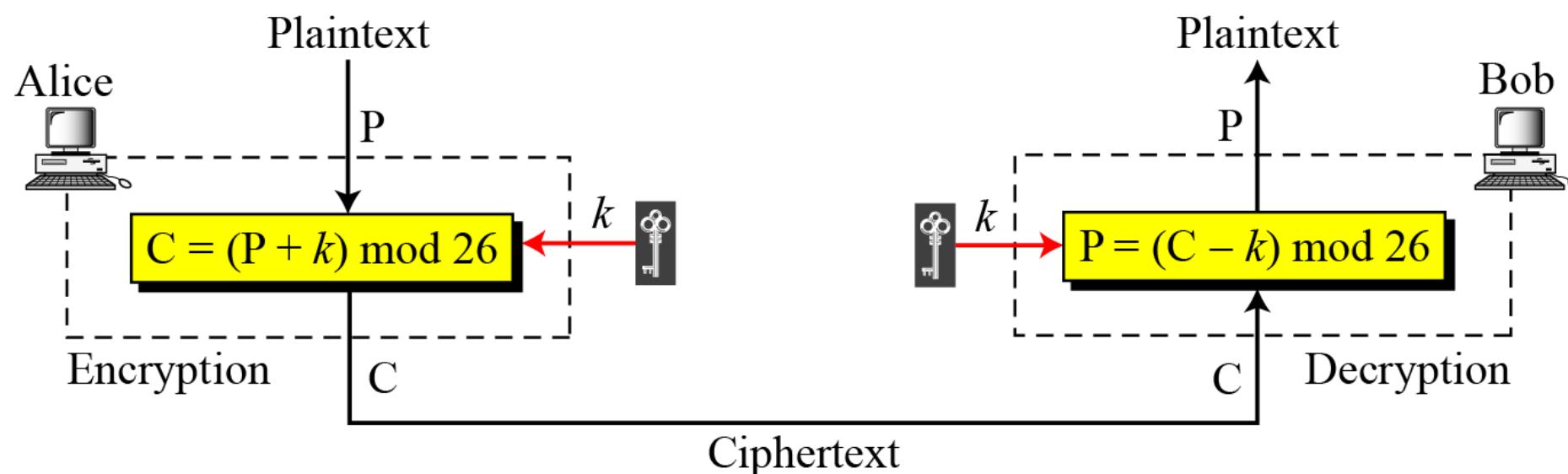
Ciphertext: KHOOR

2.2. Phương pháp Mã hóa thay thế

□ Mã hóa thay thế đơn ký tự

- ❖ Trong trường hợp đơn giản, mã hóa thay thế ký tự đơn trên phép cộng và thực hiện như mã hóa dịch chuyển (shift cipher) và cũng có thể gọi là mã hóa Ceasar.

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



2.2. Phương pháp Mã hóa thay thế

□ Mã hóa thay thế đơn ký tự

- ❖ Sử dụng mã hóa thay thế đơn ký tự bằng phép toán cộng, với khóa key = 15, để mã hóa thông điệp “hello”.

Plaintext: h → 07

Encryption: $(07 + 15) \text{ mod } 26$

Ciphertext: 22 → W

Plaintext: e → 04

Encryption: $(04 + 15) \text{ mod } 26$

Ciphertext: 19 → T

Plaintext: l → 11

Encryption: $(11 + 15) \text{ mod } 26$

Ciphertext: 00 → A

Plaintext: l → 11

Encryption: $(11 + 15) \text{ mod } 26$

Ciphertext: 00 → A

Plaintext: o → 14

Encryption: $(14 + 15) \text{ mod } 26$

Ciphertext: 03 → D

- ❖ Giải mã thông điệp “WTAAD”, với khóa key = 15.

Ciphertext: W → 22

Decryption: $(22 - 15) \text{ mod } 26$

Plaintext: 07 → h

Ciphertext: T → 19

Decryption: $(19 - 15) \text{ mod } 26$

Plaintext: 04 → e

Ciphertext: A → 00

Decryption: $(00 - 15) \text{ mod } 26$

Plaintext: 11 → l

Ciphertext: A → 00

Decryption: $(00 - 15) \text{ mod } 26$

Plaintext: 11 → l

Ciphertext: D → 03

Decryption: $(03 - 15) \text{ mod } 26$

Plaintext: 14 → o

2.2. Phương pháp Mã hóa thay thế

□ Mã hóa thay thế đơn ký tự

- ❖ Giả sử thông điệp mã hóa bị lấy cắp, thực hiện phá mã cho bản mã “UVACLYFZLJBYL”
- ❖ Thực hiện phá mã bằng Brute-force như sau:

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

2.2. Phương pháp Mã hóa thay thế

□ Mã hóa thay thế đơn ký tự

- ❖ Tần số ký tự trong tiếng Anh

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		

- ❖ Tần số của nhóm hai ký tự (*diagram*), nhóm ba ký tự (*trigram*).

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

2.2. Phương pháp Mã hóa thay thế

□ Mã hóa thay thế đơn ký tự

- ❖ Sử dụng phương pháp thống kê (*phân tích tần số*), hãy phân tích tần số các ký tự sau:

XLILSYWIMWRSAJSVWEPIJSVJSYVQMPPMSRHSPPEVWMXMWASVX-LQSVILY-VVCFIJSVIXLIWIPPIVVIGIMZIWQSVISJJIVW

- ❖ Sau khi phân tích tần số của bản mã trên, thì nhận thấy tần số của các ký tự lần lượt là I = 14, V = 13, S= 12... Trong đó, ký tự xuất hiện nhiều nhất là I (14 lần), vì vậy khả năng là ký tự E được mã hóa thành I là cao, điều này có nghĩa là khóa key = 4.

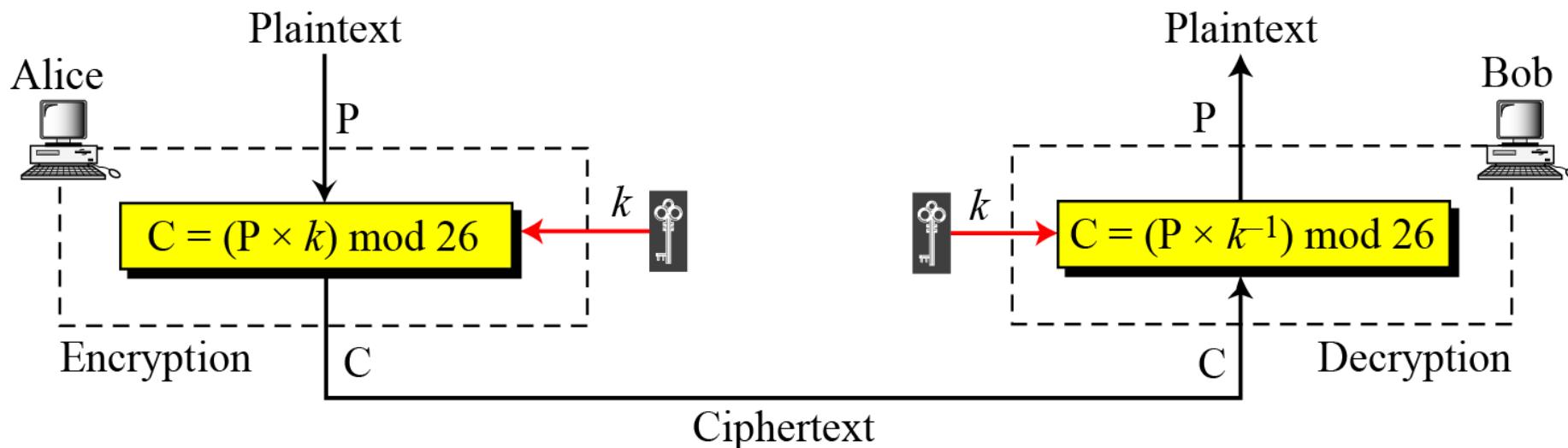
- ❖ Kết quả sau khi giải mã:

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

2.2. Phương pháp Mã hóa thay thế

❑ Mã hóa thay thế đơn ký tự

- ❖ Mã hóa thay thế đơn ký tự trên phép nhân, thông điệp gốc và mã hóa là các số nguyên trong tập Z_{26} , khóa trong tập Z_{26}^* (*vì phải có phần tử nghịch đảo*).



- ❖ Xác định không gian khóa để mã hóa thay thế trên phép nhân của Z_{26} .
 - Tập các phần tử khả nghịch trong Z_{26} gồm:
 - 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25

2.2. Phương pháp Mã hóa thay thế

❑ Mã hóa thay thế đơn ký tự

- ❖ Sử dụng mã hóa thay thế trên phép nhân để mã hóa thông điệp “hello” với khóa key = 7.
- ❖ Kết quả mã hóa: XCZZU

Plaintext: h → 07

Encryption: $(07 \times 07) \bmod 26$

ciphertext: 23 → X

Plaintext: e → 04

Encryption: $(04 \times 07) \bmod 26$

ciphertext: 02 → C

Plaintext: l → 11

Encryption: $(11 \times 07) \bmod 26$

ciphertext: 25 → Z

Plaintext: l → 11

Encryption: $(11 \times 07) \bmod 26$

ciphertext: 25 → Z

Plaintext: o → 14

Encryption: $(14 \times 07) \bmod 26$

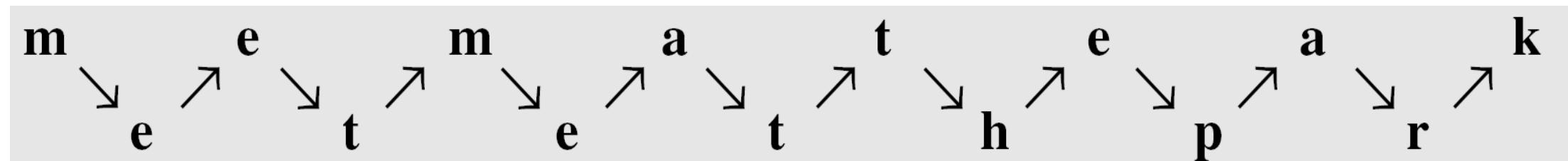
ciphertext: 20 → U

- ❖ Tìm phần tử nghịch đảo của khóa key = 7.
- ❖ Thực hiện giải mã trên phép nhân?

2.3. Mã hóa chuyển vị (Transposition)

Mã hóa chuyển vị không khóa (Keyless Transposition)

- ❖ Mã hóa chuyển vị không thay thế mỗi ký tự bằng ký tự khác, mà thực hiện thay đổi vị trí của các ký tự.
- ❖ Mã hóa chuyển vị bằng cách hoán vị các ký tự.
- ❖ Trong trường hợp đơn giản, bản mã được đọc theo dòng, ví dụ thông điệp “Meet me at the park” được viết như sau:



- ❖ Thông điệp mã hóa: “MEMATEAKETETHPR”

2.3. Mã hóa chuyển vị (Transposition)

- ❖ Người gửi và người nhận thỏa thuận số cột để mã hóa thông điệp và viết theo dòng như sau:

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

- ❖ Thông điệp được mã hóa: “**MMTAEEHREAEKTTP**”
- ❖ Việc mã hóa được thực hiện bằng cách hoán vị các ký tự trong thông điệp ban đầu theo vị trí

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
01	05	09	13	02	06	10	14	03	07	11	15	04	08	12

2.4. Mã hóa tích

- ❖ Mã hóa chỉ sử dụng phép thay thế (**substitution**) hay phép đổi chỗ (**transposition**) không an toàn (do đặc tính của ngôn ngữ).
- ❖ Sử dụng liên tiếp các thao tác mã hóa đơn giản sẽ tạo ra cách mã hóa thông tin an toàn hơn.
 - **Substitution** kết hợp với **Substitution** an toàn hơn 1 phép **Substitution**.
 - **Transposition** kết hợp với **Transposition** an toàn hơn 1 phép **Transposition**.
 - **Substitution** kết hợp **Transposition** cho kết quả an toàn hơn nhiều so với việc chỉ dùng một loại thao tác (*thay thế* hay *đổi chỗ*).
- ❖ Đây là ý tưởng mở đầu cho các phương pháp mã hóa hiện đại.

2.5. Quy trình mã hóa

❑ Quy trình mã hóa theo khối

❖ Data Path

- Thông thường, quy trình mã hóa bao gồm nhiều chu kỳ mã hóa (*round*) liên tiếp nhau; mỗi chu kỳ gồm nhiều thao tác mã hóa.

❖ Key Schedule

- Từ khóa gốc (*secret key*), phát sinh (*có quy luật*) các giá trị khóa sẽ được sử dụng trong mỗi chu kỳ mã hóa (*round key*).

2.5. Quy trình mã hóa

❑ Kiến trúc chu kỳ mã hóa

❖ Kiến trúc phổ biến của chu kỳ mã hóa:

- Kiến trúc Fiestel

- Ví dụ: Blowfish, Camellia, CAST-128, DES, FEAL, KASUMI, LOKI97, Lucifer, MARS, MAGENTA, MISTY1, RC5, TEA, Triple DES, Twofish, XTEA

- Kiến trúc SPN

- Ví dụ: Rijndael – AES, Anubis...

2.5. Quy trình mã hóa

Quy trình Mã hóa theo kiến trúc Feistel

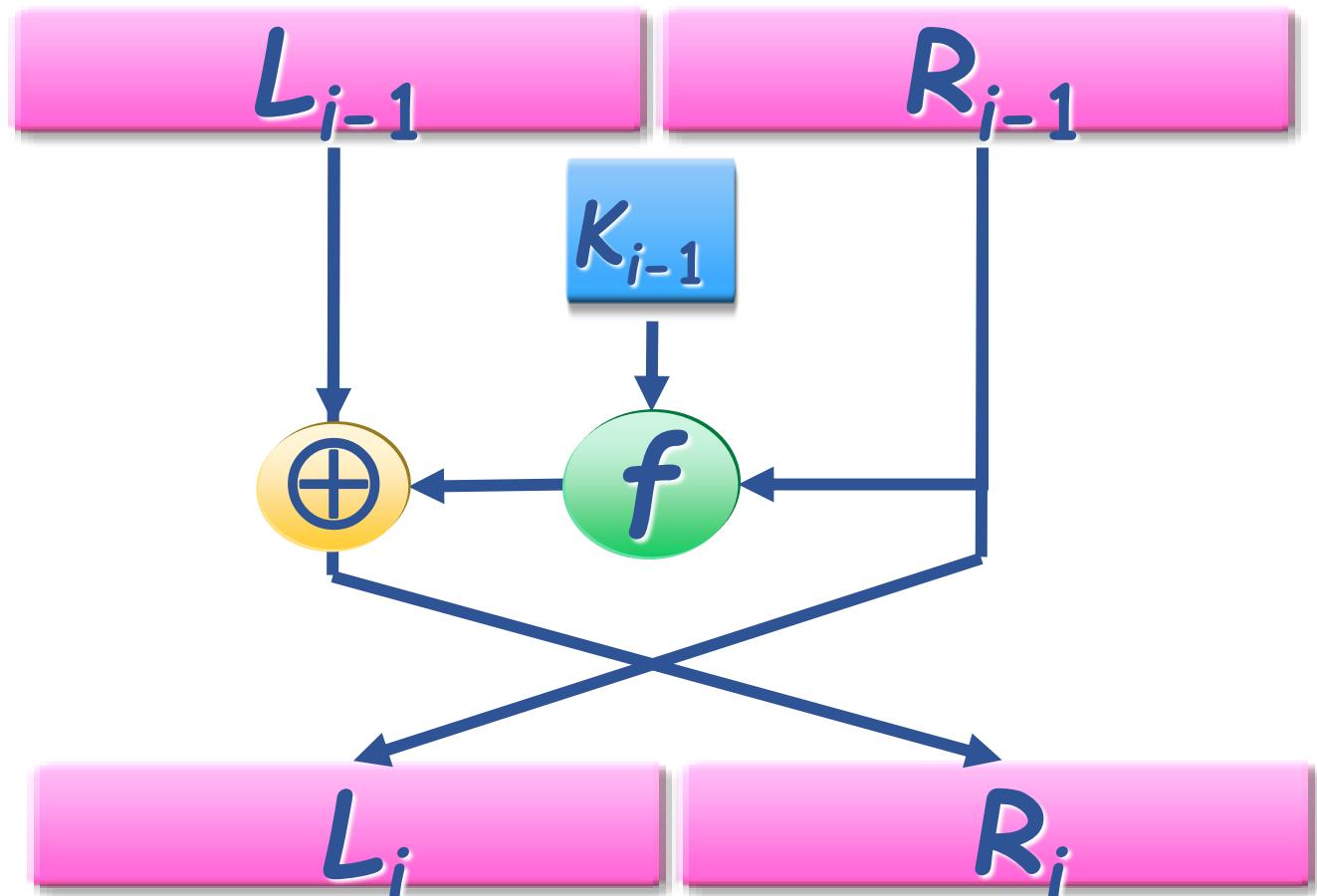
Chu kỳ mã hóa 1

...

Chu kỳ mã hóa i

...

Chu kỳ mã hóa Nr



$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_{i-1})$$

2.5. Quy trình mã hóa

Quy trình giải mã theo kiến trúc Feistel

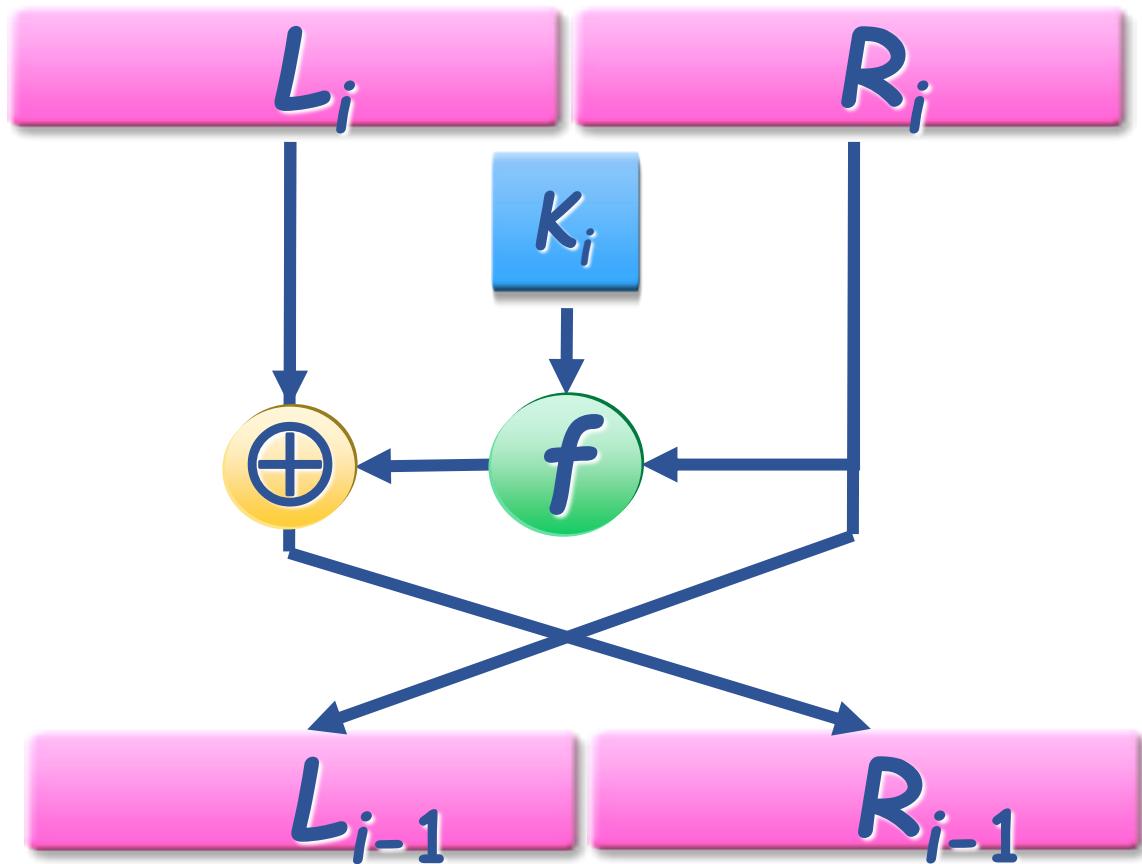
Chu kỳ giải mã Nr

...

Chu kỳ giải mã i

...

Chu kỳ giải mã 1



$$R_{i-1} = L_i$$

$$L_{i-1} = R_i \oplus f(L_i, K_i)$$

DES **(Data Encryption Standard)**

3. Mã hóa bất đối xứng

1. Mở đầu
2. Hệ mã hóa RSA
3. Một số hệ mã công khai khác
4. Mã hóa đối xứng và mã hóa bất đối xứng

3.1. Mở đầu

- ❖ Vấn đề phát sinh trong các hệ thống mã hóa quy ước là việc quy ước chung mã khóa k giữa người gửi A và người nhận B.
- ❖ Trên thực tế, nhu cầu thay đổi nội dung của mã khóa k là cần thiết, do đó, cần có sự trao đổi thông tin về mã khóa k giữa A và B.
- ❖ Để bảo mật mã khóa k , A và B phải trao đổi với nhau trên một kênh liên lạc thật sự an toàn và bí mật.
- ❖ Tuy nhiên, rất khó có thể bảo đảm được sự an toàn của kênh liên lạc nên mã khóa k vẫn có thể bị phát hiện bởi người C!

3.1. Mở đầu

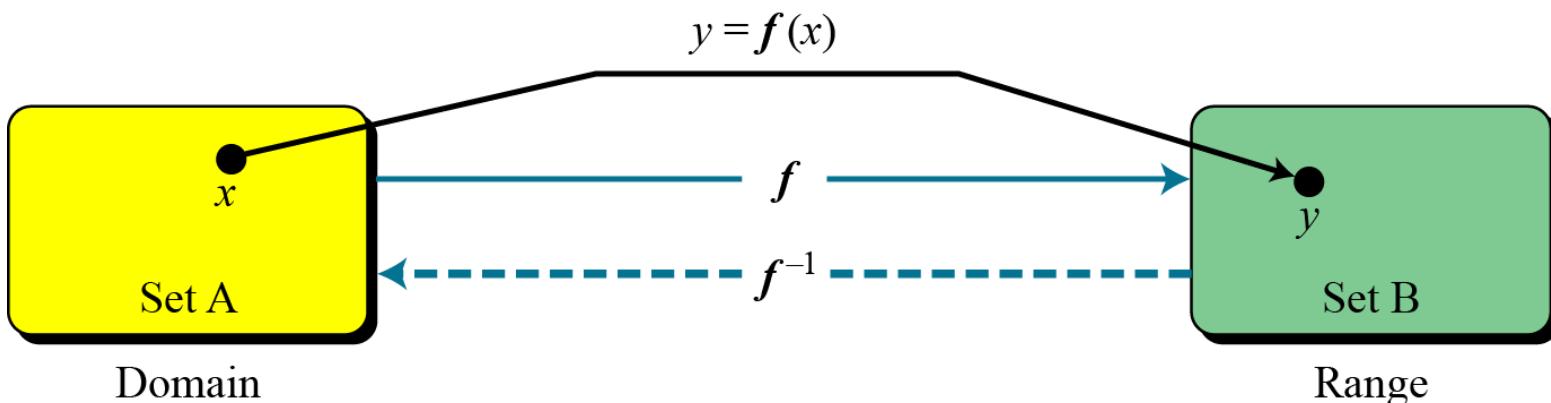
- Ý tưởng về hệ thống mã hóa khóa công cộng được Martin Hellman, Ralph Merkle và Whitfield Diffie tại Đại học Stanford giới thiệu vào năm 1976.
- Sau đó, phương pháp Diffie-Hellman của Martin Hellman và Whitfield Diffie đã được công bố.
- Năm 1977, trên báo "*The Scientific American*", nhóm tác giả Ronald Rivest, Adi Shamir và Leonard Adleman đã công bố phương pháp RSA, phương pháp mã hóa khóa công cộng nổi tiếng và được sử dụng rất nhiều hiện nay trong các ứng dụng mã hóa và bảo vệ thông tin.

3.1. Mở đầu

- ❖ Một hệ thống khóa công cộng sử dụng hai loại khóa trong cùng một cặp khóa:
 - Khóa công cộng (public key) được công bố rộng rãi và được sử dụng trong mã hóa thông tin,
 - Khóa riêng (private key) chỉ do một người nắm giữ và được sử dụng để giải mã thông tin đã được mã hóa bằng khóa công cộng.
- ❖ Các phương pháp mã hóa này khai thác những ánh xạ f mà việc thực hiện ánh xạ ngược f^{-1} rất khó so với việc thực hiện ánh xạ f . Chỉ khi biết được mã khóa riêng thì mới có thể thực hiện được ánh xạ ngược f^{-1} .

3.1. Mở đầu

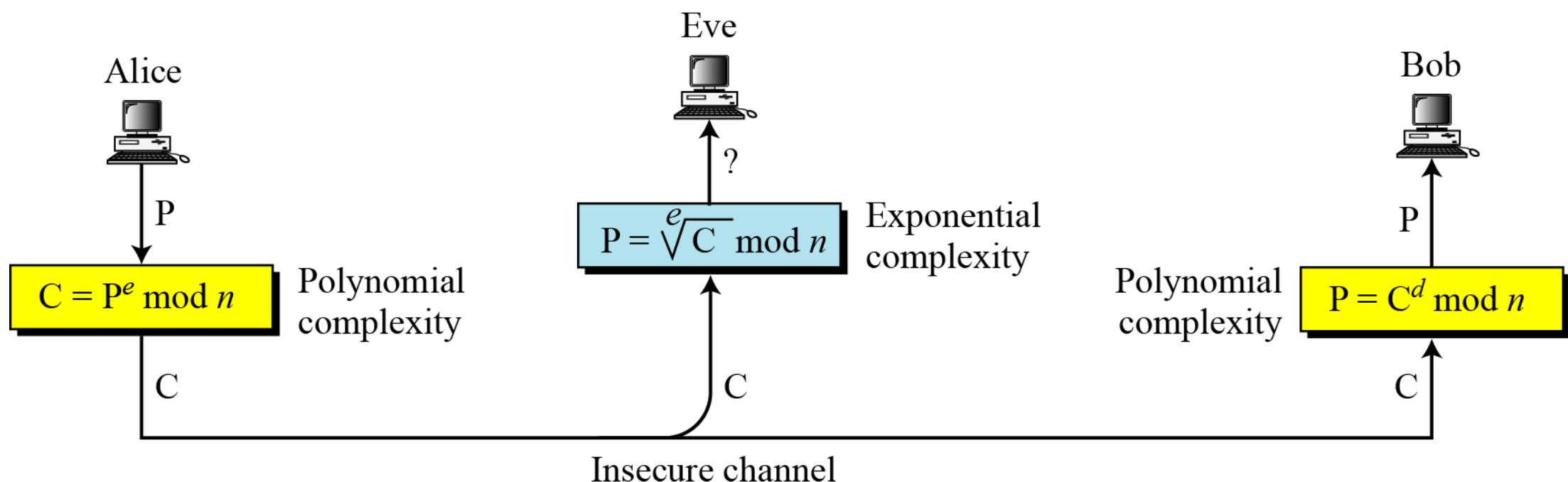
- ❖ Ý tưởng chính của mã hóa bất đối xứng là sử dụng khái niệm về “*chức năng xử lý một cửa*”
- ❖ Một hàm ánh xạ một miền (domain) đến một vùng:



- ❖ **Ví dụ:** khi n là số lớn, $n = p \times q$, nếu có p và q thì dễ dàng tính được n , nhưng nếu có n thì rất khó để tính p hoặc q .
- ❖ **Ví dụ:** cho $y = x^k \bmod n$, nếu có x, k, n thì dễ dàng tính y , nhưng nếu có y, k, n thì rất khó tính được x .

3.2. Hệ mã hóa RSA

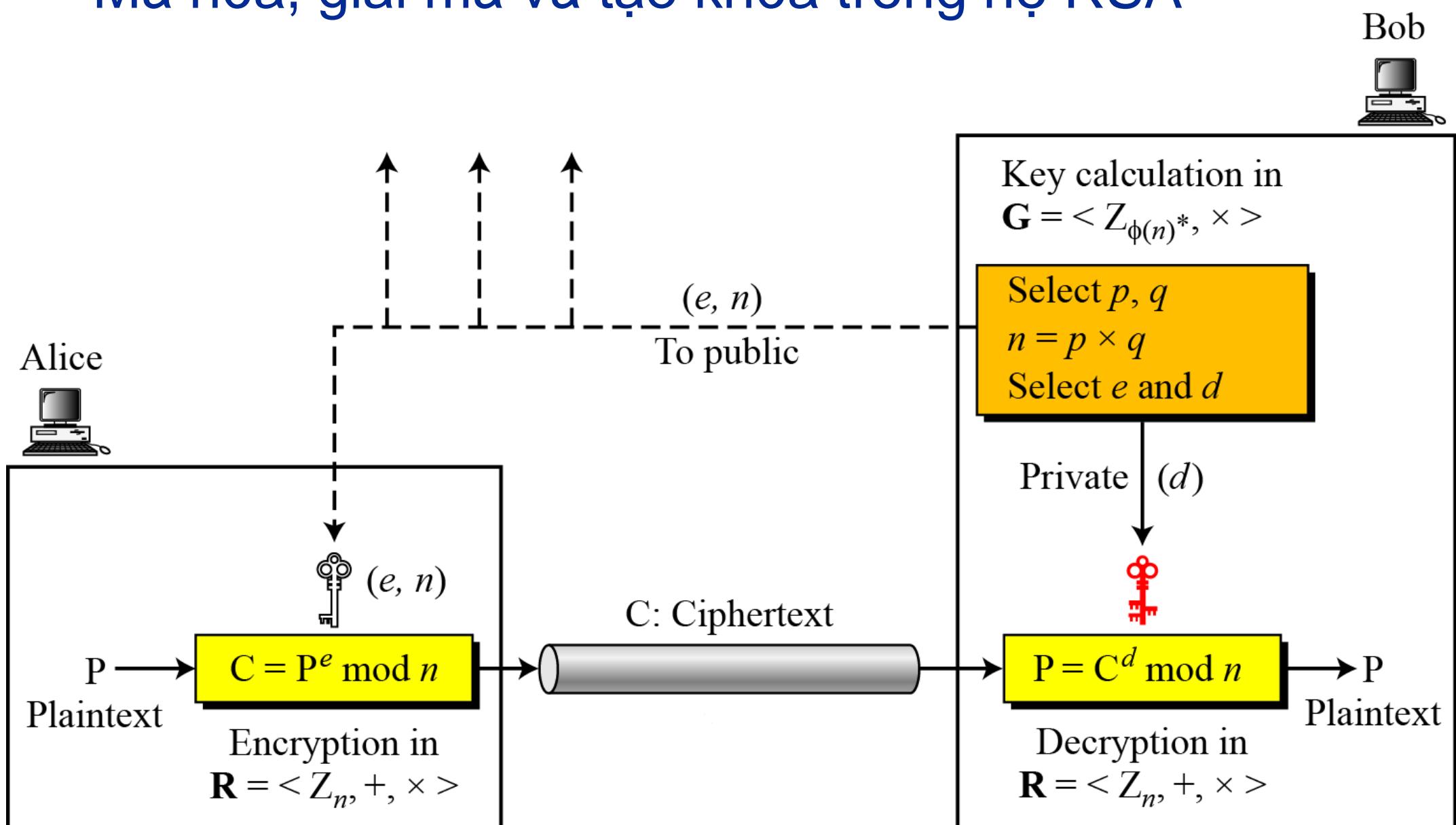
- ❖ Thuật toán khóa công cộng thông dụng nhất là RSA (viết tắt của người phát minh Rivest, Shamir, Adleman).
- ❖ RSA là phương pháp mã hóa/giải mã theo hàm mũ; để phá mã, cần phải thực hiện tính $\sqrt[e]{C} \text{ mod } n$.



3.2. Hệ mã hóa RSA

□ Thủ tục thực hiện mã hóa RSA

Mã hóa, giải mã và tạo khóa trong hệ RSA



3.2. Hệ mã hóa RSA

□ Thuật toán tạo Khóa

RSA_Key_Generation

{

Select two large primes p and q such that $p \neq q$.

$n \leftarrow p \times q$

$\phi(n) \leftarrow (p - 1) \times (q - 1)$

Select e such that $1 < e < \phi(n)$ and e is coprime to $\phi(n)$

$d \leftarrow e^{-1} \bmod \phi(n)$ // d is inverse of e modulo $\phi(n)$

Public_key $\leftarrow (e, n)$ // To be announced publicly

Private_key $\leftarrow d$ // To be kept secret

return Public_key and Private_key

}

3.2. Hệ mã hóa RSA

□ Ví dụ 1

- Bob chọn 7 và 11 là p và q, $n = p \times q = 77$
- Giá trị $\phi(n) = (7-1) \times (11-1) = 60$
- Chọn hai giá trị lũy thừa từ Z_{60}^* : $e = 13$, $d = 37$ (hai phần tử khả nghịch: $e \times d \text{ mod } 60 = 1$)

3.2. Hệ mã hóa RSA

❑ Ví dụ 1

- Thông điệp gửi là 5, thực hiện mã hóa trong $Z_n = Z_{77}$

Plaintext: 5

$$C = 5^{13} = 26 \text{ mod } 77$$

Ciphertext: 26

- Sử dụng khóa bí mật là 37 để giải mã

Ciphertext: 26

$$P = 26^{37} = 5 \text{ mod } 77$$

Plaintext: 5

- Giả sử John sử dụng khóa công cộng để mã hóa và gửi một thông điệp cho Bob, thông điệp gốc là 63.

Plaintext: 63

$$C = 63^{13} = 28 \text{ mod } 77$$

Ciphertext: 28

- Bob nhận được và giải mã với khóa bí mật là 37

Ciphertext: 28

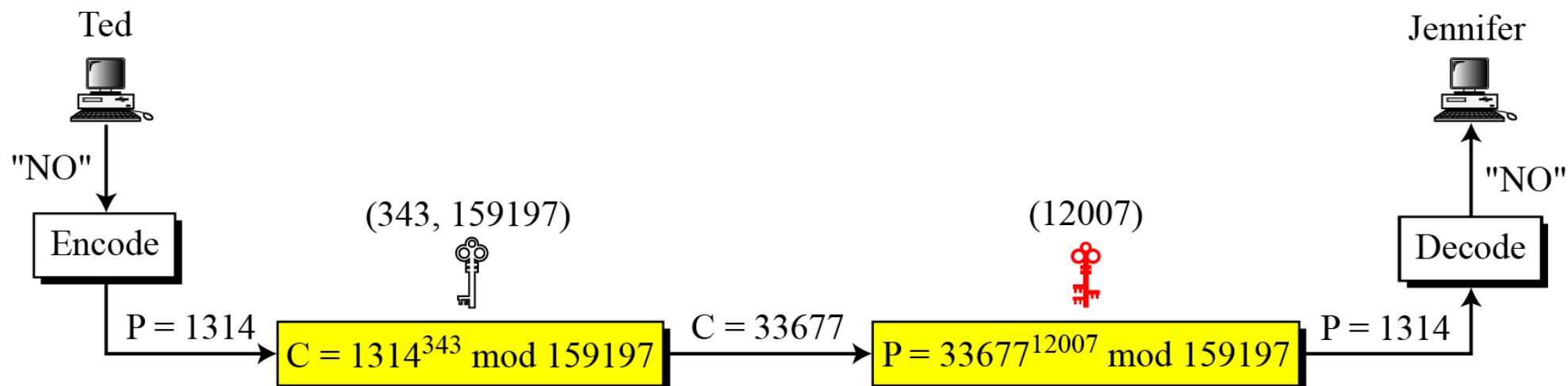
$$P = 28^{37} = 63 \text{ mod } 77$$

Plaintext: 63

3.2. Hệ mã hóa RSA

□ Ví dụ 2

- Giả sử Jennifer tạo một cặp khóa $p = 397$, $q = 401$.
- Giá trị $n = p \times q = 159197$, $\phi(n) = 158400$
- Chọn $e = 343$ và $d = 12007$ trong Z_{158400}
- Ted gửi thông điệp “NO” cho Jennifer nếu biết e và n .
- Thông điệp được gửi được chuyển thành 1314 (mỗi ký tự được mã hóa thành hai ký số từ 00-25).



3.2. Hệ mã hóa RSA

□ Ví dụ thực tế

- Chọn cặp khóa p, q có 512-bit, tính $n = p \times q$ và $\phi(n)$
- Chọn hệ số lũy thừa e và tính d trong $Z_{\phi(n)}$
- Số p có 159 ký tự

$p =$ 961303453135835045741915812806154279093098455949962158225831508796
479404550564706384912571601803475031209866660649242019180878066742
1096063354219926661209

$q =$ 120601919572314469182767942044508960015559250546370339360617983217
314821484837646592153894532091752252732268301071206956046025138871
45524969000359660045617

3.2. Hệ mã hóa RSA

□ Ví dụ thực tế

- Tính $n = p \times q$ có 309 ký số

$n =$ 115935041739676149688925098646158875237714573754541447754855261376
147885408326350817276878815968325168468849300625485764111250162414
552339182927162507656772727460097082714127730434960500556347274566
628060099924037102991424472292215772798531727033839381334692684137
327622000966676671831831088373420823444370953

- Tính $\phi(n) = (p-1) \times (q-1)$ có 309 ký số

$\phi(n) =$ 115935041739676149688925098646158875237714573754541447754855261376
147885408326350817276878815968325168468849300625485764111250162414
552339182927162507656751054233608492916752034482627988117554787657
013923444405716989581728196098226361075467211864612171359107358640
614008885170265377277264467341066243857664128

3.2. Hệ mã hóa RSA

☐ Ví dụ thực tế

- Chọn $e = 35535$, giá trị d là

$e =$	35535
$d =$	580083028600377639360936612896779175946690620896509621804228661113 805938528223587317062869100300217108590443384021707298690876006115 306202524959884448047568240966247081485817130463240644077704833134 010850947385295645071936774061197326557424237217617674620776371642 0760033708533328853214470885955136670294831

- Giả sử thông điệp gốc là: “THIS IS A TEST”, mỗi ký tự được chuyển thành hai ký số (01-26)

$P =$	1907081826081826002619041819
-------	------------------------------

3.2. Hệ mã hóa RSA

❑ Ví dụ thực tế

- ❖ Bản mã được tính $C = P^e$

$C =$	475309123646226827206365550610545180942371796070491716523239243054 452960613199328566617843418359114151197411252005682979794571736036 101278218847892741566090480023507190715277185914975188465888632101 148354103361657898467968386763733765777465625079280521148141844048 14184430812773059004692874248559166462108656
-------	--

- ❑ Bob có thể khôi phục lại thông điệp gốc $P = C^d$

$P =$	1907081826081826002619041819
-------	------------------------------

- ❑ Thông điệp gốc được khôi phục là:

“THIS IS A TEST”

3.3. Một số hệ mã công khai khác

- Hệ mã RABIN
- Hệ mã ELGAMAL
- Hệ mã đường cong ELLIPTIC

3.4. Mã hóa đối xứng và mã hóa bất đối xứng

- ❖ Các phương pháp mã hóa quy ước có ưu điểm xử lý rất nhanh so với các phương pháp mã hóa khóa công cộng.
- ❖ Do khóa dùng để mã hóa cũng được dùng để giải mã nên cần phải giữ bí mật nội dung của khóa và mã khóa được gọi là khóa bí mật (secret key). Ngay cả trong trường hợp khóa được trao đổi trực tiếp thì mã khóa này vẫn có khả năng bị phát hiện. Vấn đề khó khăn đặt ra đối với các phương pháp mã hóa này chính là bài toán trao đổi mã khóa.

3.4. Mã hóa đối xứng và mã hóa bất đối xứng

- ❖ Khóa công cộng dễ bị tấn công hơn khóa bí mật.
- ❖ Để tìm ra được khóa bí mật, người giải mã cần phải có thêm một số thông tin liên quan đến các đặc tính của văn bản nguồn trước khi mã hóa để tìm ra manh mối giải mã thay vì phải sử dụng phương pháp vét cạn mã khóa.
- ❖ Ngoài ra, việc xác định xem thông điệp sau khi giải mã có đúng là thông điệp ban đầu trước khi mã hóa hay không lại là một vấn đề khó khăn.
- ❖ Đối với các khóa công cộng, việc công phá hoàn toàn có thể thực hiện được với điều kiện có đủ tài nguyên và thời gian xử lý.

4. Chữ ký số

- **Mục tiêu của chữ ký số (Digital Signature):**
 - ❖ Xác nhận người dùng (*Authentication*)
 - ❖ Tính toàn vẹn thông tin (*Data Integrity*)
 - ❖ Không thể từ chối trách nhiệm (*Non-Repudiation*)

4. Chữ ký số

❑ Một số khái niệm cơ bản

- ❖ Chữ ký số (Digital Signature) là một chuỗi dữ liệu liên kết với một thông điệp (message) và thực thể tạo ra thông điệp;
- ❖ Giải thuật tạo chữ ký số (Digital Signature generation algorithm) là một phương pháp sinh chữ ký số;
- ❖ Giải thuật kiểm tra chữ ký số (Digital Signature verification algorithm) là một phương pháp xác minh tính xác thực của chữ ký số, có nghĩa là nó thực sự được tạo ra bởi 1 bên chỉ định;

4. Chữ ký số

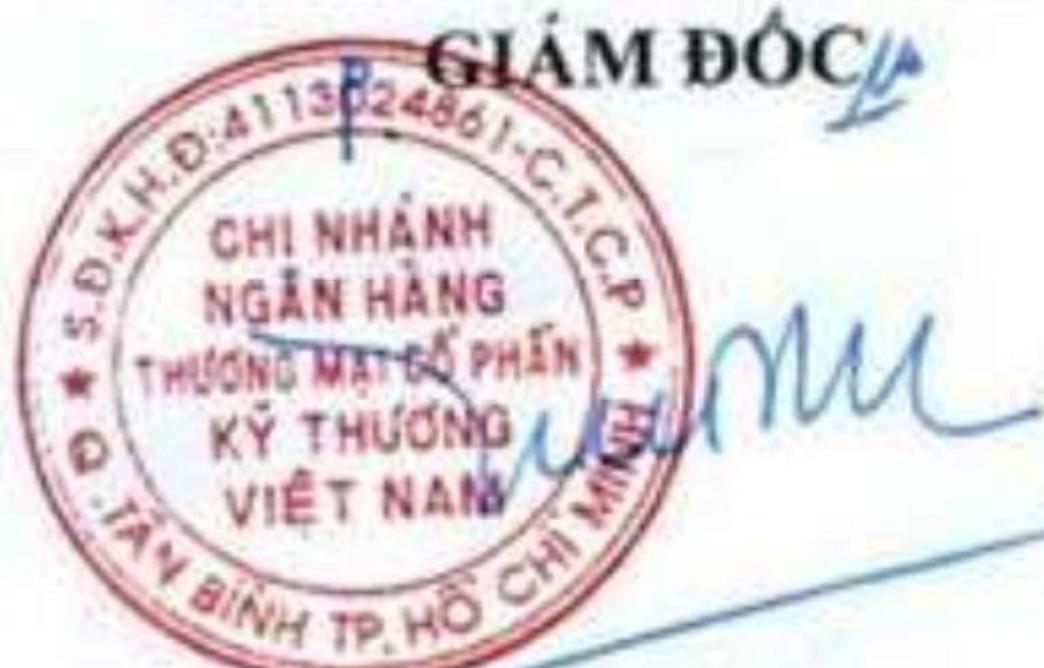
❑ Một số khái niệm cơ bản

- ❖ Một hệ chữ ký số (Digital Signature Scheme) bao gồm giải thuật tạo chữ ký số và giải thuật kiểm tra chữ ký số.
- ❖ Quá trình tạo chữ ký số (Digital signature signing process) bao gồm:
 - Giải thuật tạo chữ ký số
 - Phương pháp chuyển dữ liệu thông điệp thành dạng có thể ký được.
- ❖ Quá trình kiểm tra chữ ký số (Digital signature verification process) bao gồm:
 - Giải thuật kiểm tra chữ ký số
 - Phương pháp khôi phục dữ liệu từ thông điệp

4. Chữ ký số

❑ Ví dụ về chữ ký số

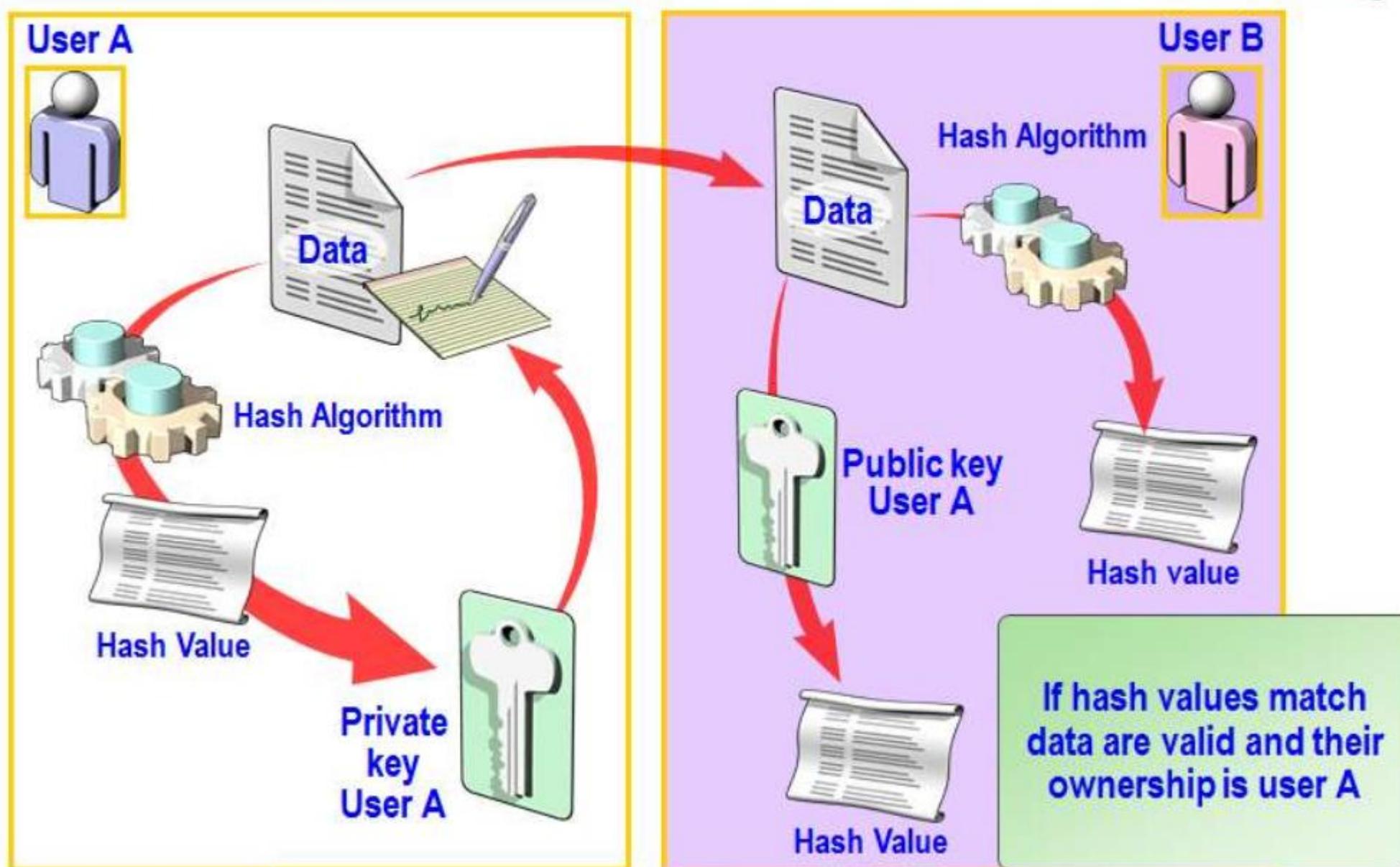
NGÂN HÀNG TMCP KỸ THƯƠNG VIỆT NAM
TECHCOMBANK TÂN BÌNH



Ngô Quang Trường

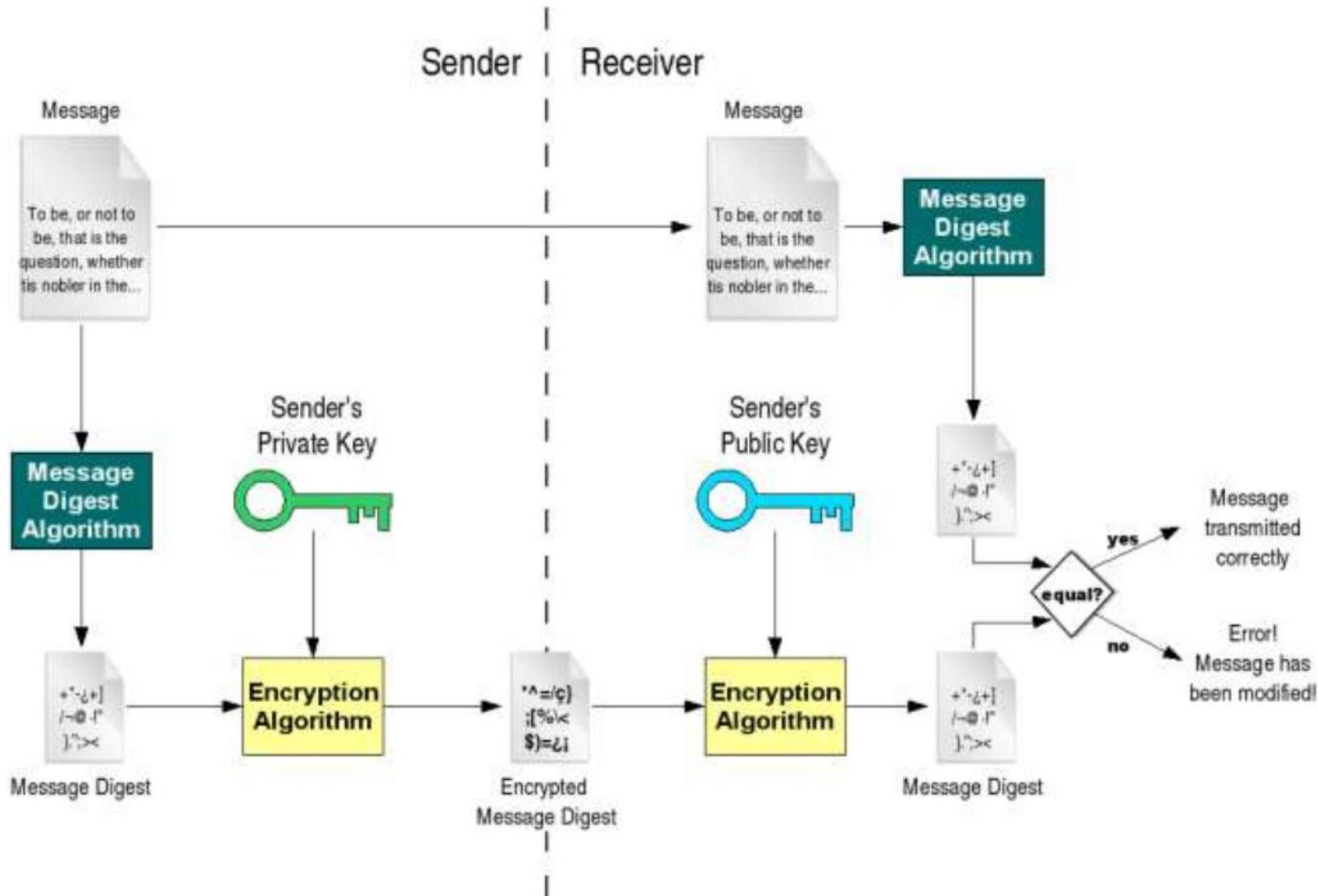
4. Chữ ký số

Digital signature



4. Chữ ký số

Quá trình ký và kiểm tra



4. Chữ ký số

□ Quá trình ký

❖ Các bước của quá trình ký một thông điệp (bên người gửi):

- Tính toán chuỗi đại diện (message digest/hash value) của thông điệp sử dụng một giải thuật băm (Hashing algorithm);
- Chuỗi đại diện được ký sử dụng khóa riêng (Private key) của người gửi và một giải thuật tạo chữ ký (Signature/Encryption algorithm). Kết quả là chữ ký số (Digital signature) của thông điệp hay còn gọi là chuỗi đại diện được mã hóa (Encrypted message digest);
- Thông điệp ban đầu (message) được ghép với chữ ký số (Digital signature) tạo thành thông điệp đã được ký (Signed message);
- Thông điệp đã được ký (Signed message) được gửi cho người nhận.

4. Chữ ký số

Quá trình kiểm tra

- ❖ Các bước của quá trình kiểm tra chữ ký (bên người nhận):
 - Tách chữ ký số và thông điệp gốc khỏi thông điệp đã ký để xử lý riêng;
 - Tính toán chuỗi đại diện MD1 (message digest) của thông điệp gốc sử dụng giải thuật băm (là giải thuật sử dụng trong quá trình ký);
 - Sử dụng khóa công khai (Public key) của người gửi để giải mã chữ ký số → chuỗi đại diện thông điệp MD2;
 - So sánh MD1 và MD2:
 - Nếu $MD1 = MD2 \rightarrow$ chữ ký kiểm tra thành công. Thông điệp đảm bảo tính toàn vẹn và thực sự xuất phát từ người gửi (do khóa công khai được chứng thực).
 - Nếu $MD1 <> MD2 \rightarrow$ chữ ký không hợp lệ. Thông điệp có thể đã bị sửa đổi hoặc không thực sự xuất phát từ người gửi.

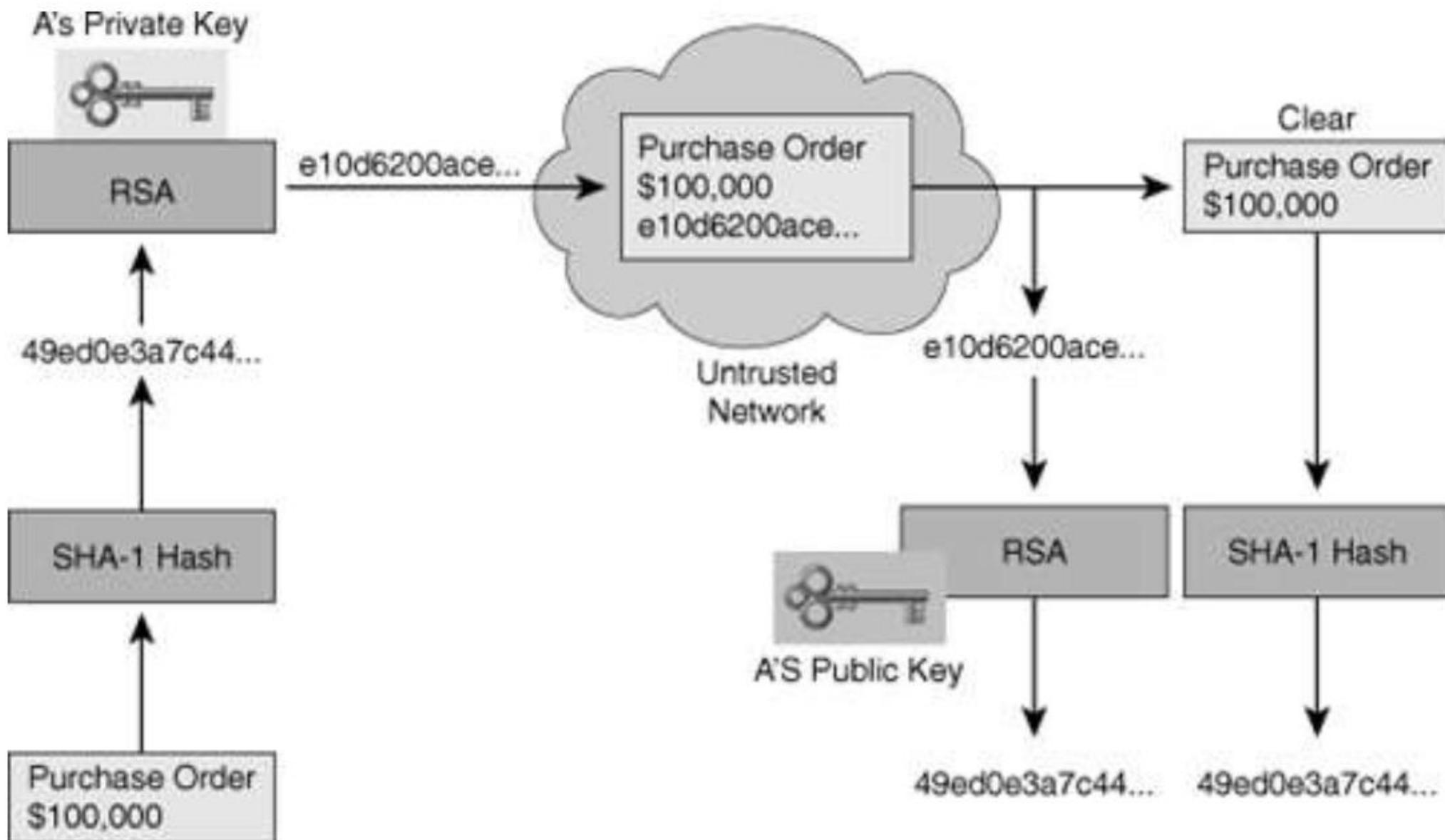
4. Chữ ký số

❑ Giải thuật chữ ký số RSA

- ❖ RSA là giải thuật cho phép thực hiện 2 tính năng:
 - Mã hóa thông điệp:
 - Người gửi mã hóa thông điệp sử dụng khóa công khai của người nhận;
 - Người nhận giải mã thông điệp sử dụng khóa riêng của mình.
 - Tạo chữ ký số:
 - Người gửi tạo chữ ký số sử dụng khóa bí mật của mình;
 - Người nhận kiểm tra chữ ký sử dụng khóa công khai của người gửi.

4. Chữ ký số

□ Giải thuật chữ ký số RSA



4. Chữ ký số

Mã hóa khóa công khai

- ❖ Public key: Mọi người đều có thể sử dụng được
- ❖ Private key: Chỉ người chủ sở hữu cặp khóa mới có thể sử dụng → Bảo mật thông tin

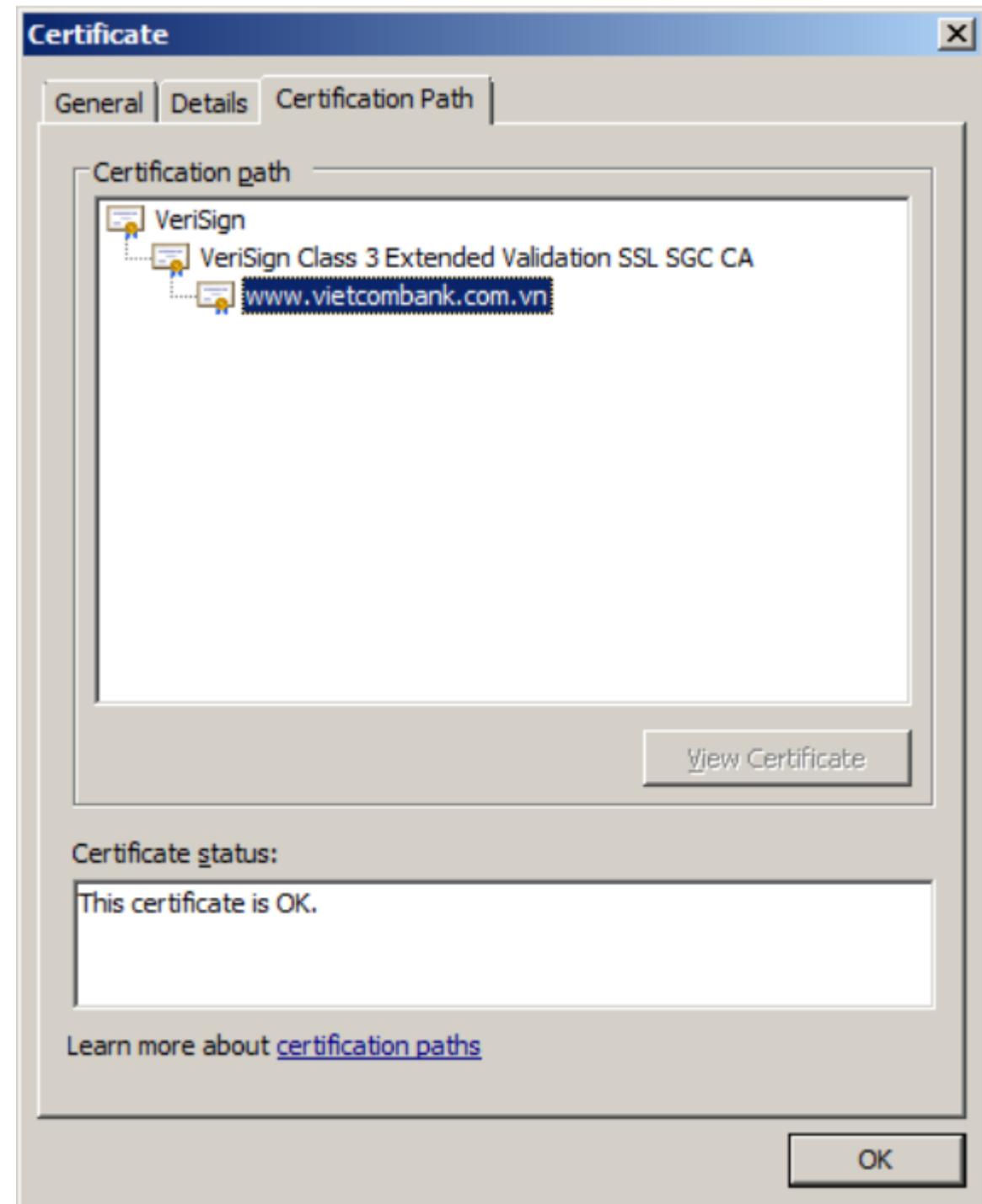
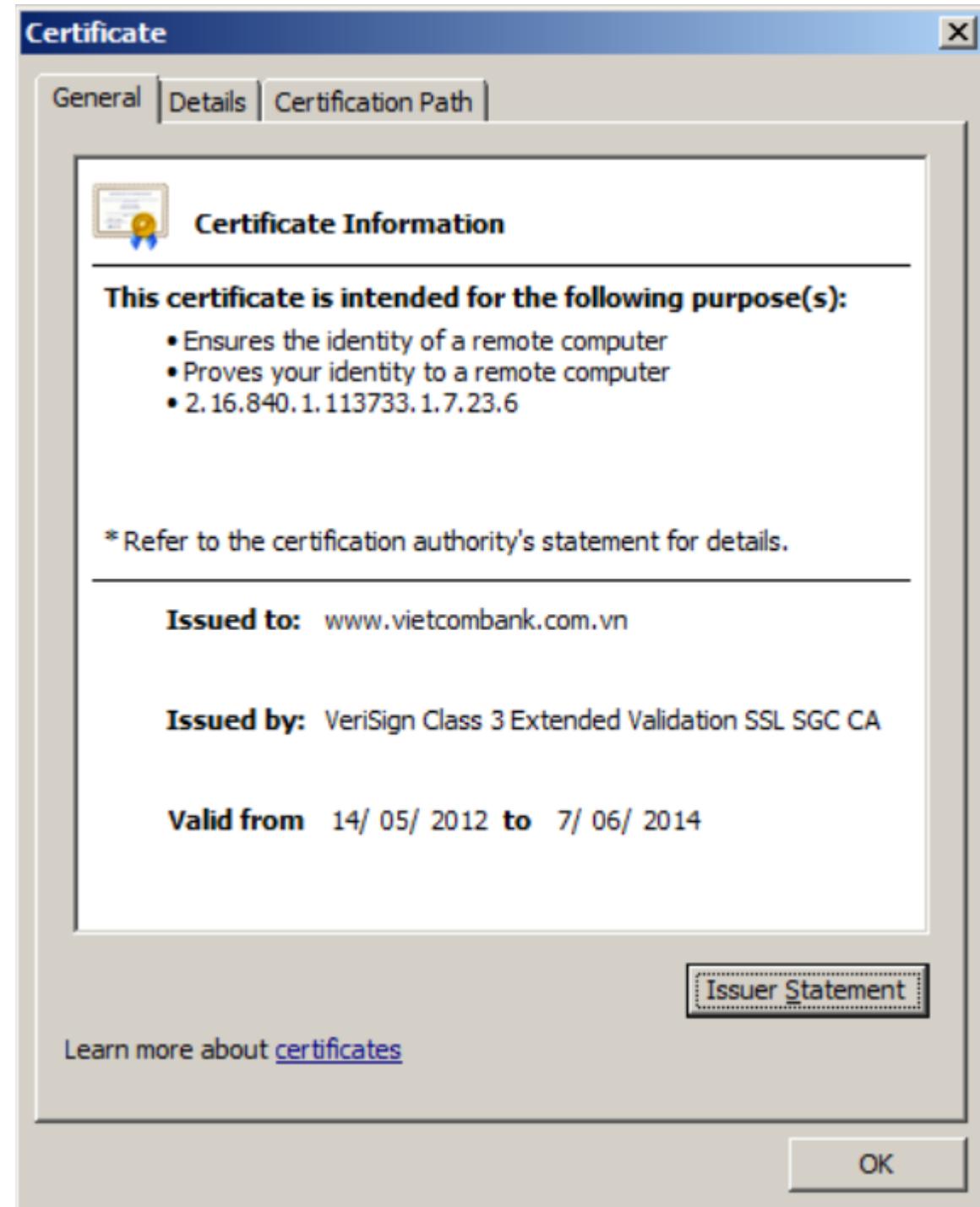
Ý tưởng chữ ký số

- ❖ Private key: Chỉ người chủ sở hữu cặp khóa mới có thể ký
- ❖ Public key: Mọi người đều có thể kiểm tra chữ ký

5. Chứng chỉ số

- ❑ Chứng chỉ số (Digital certificate), còn gọi là chứng chỉ khóa công khai (Public key certificate), hay chứng chỉ nhận dạng (Identity certificate) là một tài liệu điện tử sử dụng một chữ ký số để liên kết một khóa công khai và thông tin nhận dạng của một thực thể:
 - Chữ ký số: là chữ ký của một bên thứ 3 tin cậy, thường gọi là CA – Certificate Authority;
 - Khóa công khai: là khóa công khai trong cặp khóa công khai của thực thể;
 - Thông tin nhận dạng: là tên, địa chỉ, tên miền hoặc các thông tin định danh của thực thể.
- ❑ Chứng chỉ số có thể được sử dụng để xác minh chủ thẻ thực sự của một khóa công khai.

5. Chứng chỉ số

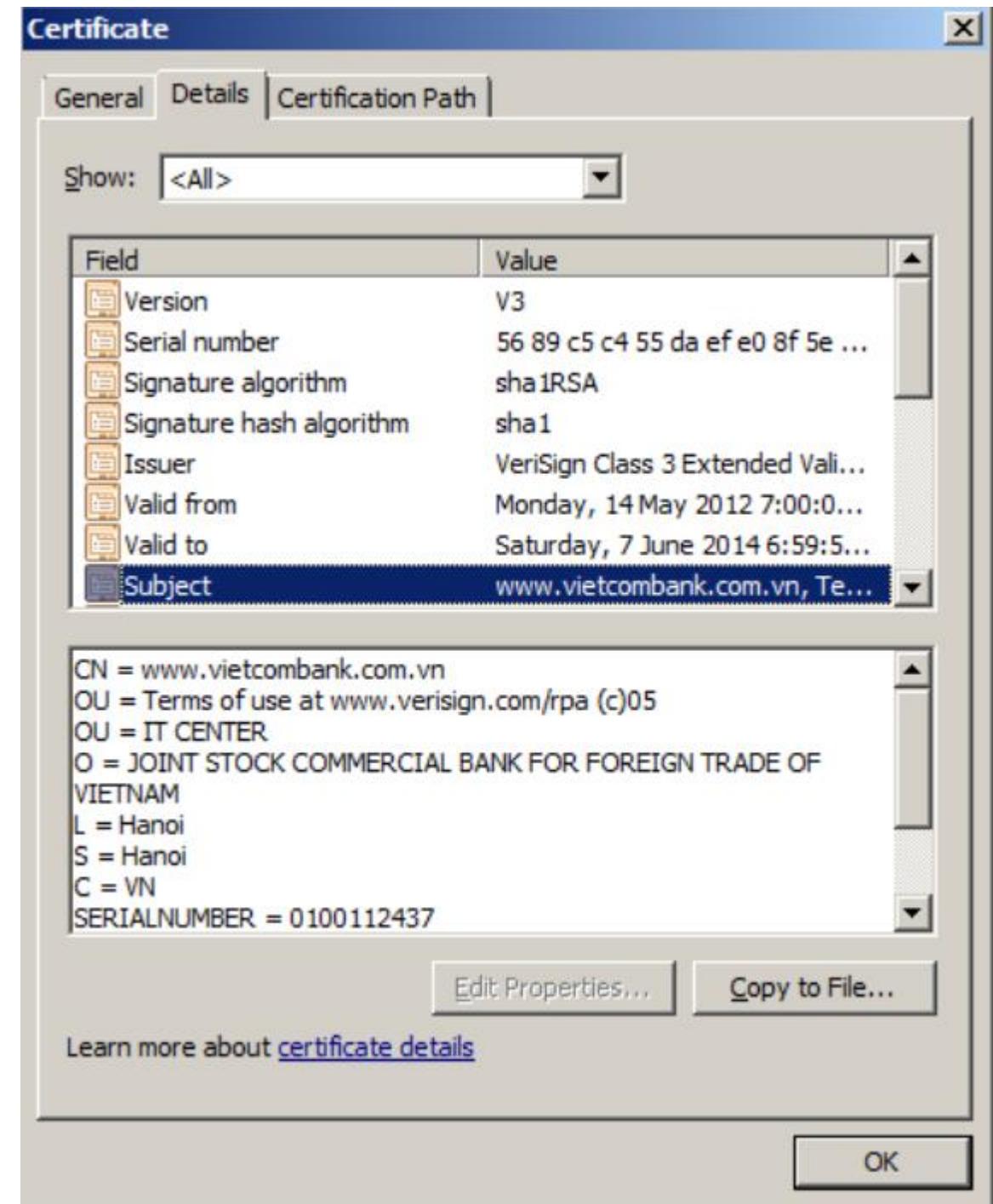


5. Chứng chỉ số

☐ Nội dung:

❖ Chứng chỉ số gồm các trường chính sau:

- **Serial Number:** Số nhận dạng của chứng chỉ số;
- **Subject:** Thông tin nhận dạng một cá nhân hoặc một tổ chức;
- **Signature Algorithm:** Giải thuật tạo chữ ký;
- **Signature Hash Algorithm:** Giải thuật tạo chuỗi băm cho tạo chữ ký;
- **Signature:** Chữ ký của người/tổ chức cấp chứng chỉ;
- **Issuer:** Người/tổ chức có thẩm quyền/tin cậy cấp chứng chỉ



5. Chứng chỉ số

❖ Chứng chỉ số gồm các trường chính sau:

- **Issuer:** Người/tổ chức có thẩm quyền/tin cậy cấp chứng chỉ;
- **Valid-From:** Ngày bắt đầu có hiệu lực của chứng chỉ;
- **Valid-To:** Ngày hết hạn sử dụng chứng chỉ;
- **Key-Usage:** Mục đích sử dụng khóa (chữ ký số, mã hóa,...);
- **Public Key:** Khóa công khai của chủ thẻ;
- **Thumbprint Algorithm:** Giải thuật hash sử dụng để tạo chuỗi băm cho khóa công khai;
- **Thumbprint:** Chuỗi băm tạo từ khóa công khai;

5. Chứng chỉ số

❖ Nội dung của trường Subject:

- CN (Common Name): Tên chung, nhưng một tên miền được gán chứng chỉ;
- OU (Organisation Unit): Tên bộ phận/phòng ban;
- O (Organisation): Tổ chức/Cơ quan/công ty;
- L (Location): Địa điểm/Quận huyện;
- S (State/Province): Bang/Tỉnh/Thành phố;
- C (Country): Đất nước.

5. Chứng chỉ số

□ Sử dụng chứng chỉ số

- ❖ Đảm bảo an toàn cho giao dịch trên nền web:
 - Dùng chứng chỉ số cho phép website chạy trên SSL (tối thiểu máy chủ phải có chứng chỉ số): HTTP → HTTPS: toàn bộ thông tin chuyển giữa server và client được đảm bảo tính bí mật (sử dụng mã hóa khóa đối xứng), toàn vẹn và xác thực (sử dụng hàm băm có khóa MAC/HMAC);
 - Chứng chỉ số để các bên xác thực thông tin nhận dạng của nhau.
- ❖ Chứng chỉ số có thể được sử dụng cho nhiều ứng dụng:
 - Email;
 - FTP;
 - Các ứng dụng khác

6. Hàm băm mật mã Hash và MAC

Các hàm băm (Hash functions) là các thuật toán để tạo các bản tóm tắt của thông điệp được sử dụng để nhận dạng và đảm bảo tính toàn vẹn của thông điệp.

- Các hàm băm là các hàm công khai được dùng để tạo các giá trị băm hay thông điệp rút gọn (message digest);
- Chiều dài của thông điệp là bất kỳ, nhưng đầu ra có chiều dài cố định.

6. Hàm băm mật mã Hash và MAC

1. Các thuộc tính của hàm băm
2. Phân loại hàm băm mật mã
3. Một số ứng dụng thực tế của hàm băm
4. Mật khẩu người dùng

6.1. Các thuộc tính của hàm băm

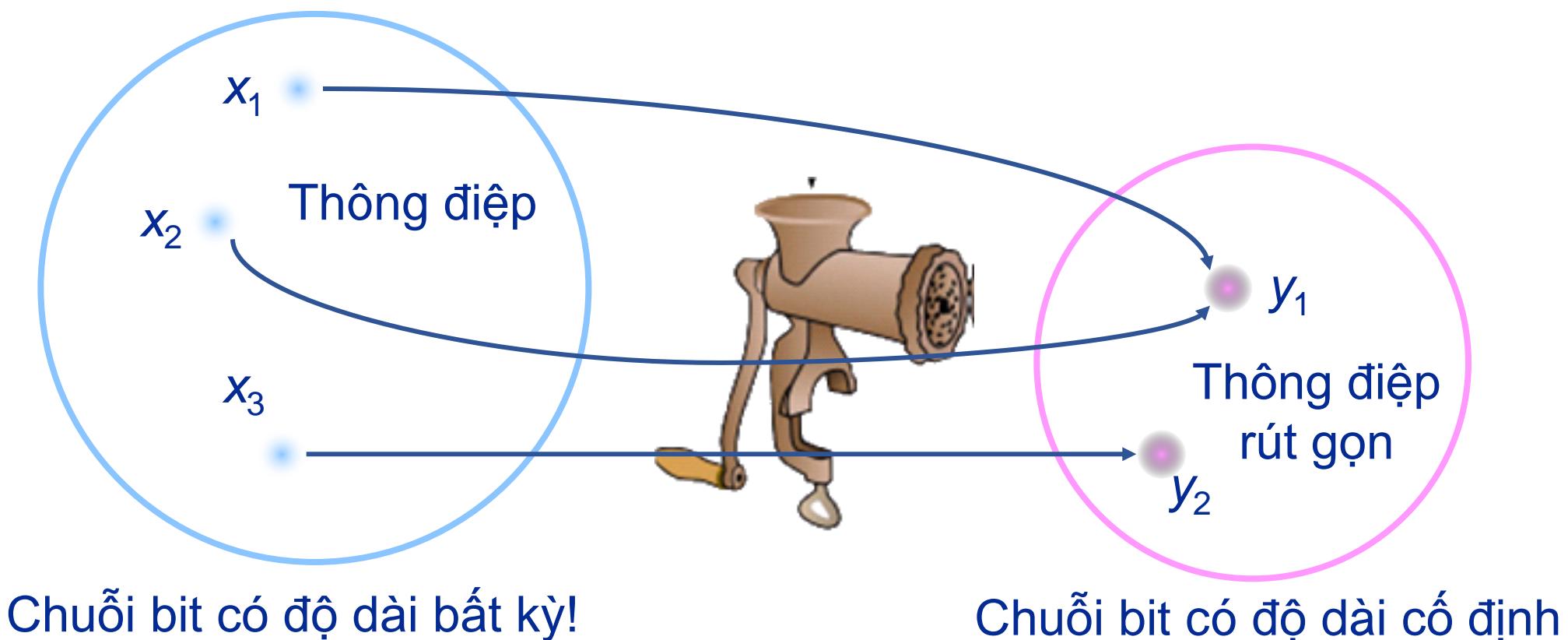
Tính toàn vẹn và tính bí mật

- **Tính toàn vẹn (Integrity):** người tấn công không thể can thiệp để sửa nội dung thông điệp.
- **Mã hóa** chỉ nhằm đảm bảo tính bí mật, không giúp đảm bảo tính toàn vẹn thông tin.
- Người tấn công có thể sửa đổi nội dung thông điệp đã được mã hóa mà không cần biết nội dung thật sự của thông điệp.
- Ví dụ: Trong đấu giá trực tuyến, có thể thay đổi giá đặt của đối thủ mà không cần biết nội dung thật sự của giá đặt.

6.1. Các thuộc tính của hàm băm

□ Ý tưởng của hàm băm

- H là hàm nén mất mát thông tin (lossy compression function)
- Hiện tượng đụng độ (Collision): $H(x)=H(x')$ với $x \neq x'$
- Kết quả của việc băm “nhìn có vẻ ngẫu nhiên”.

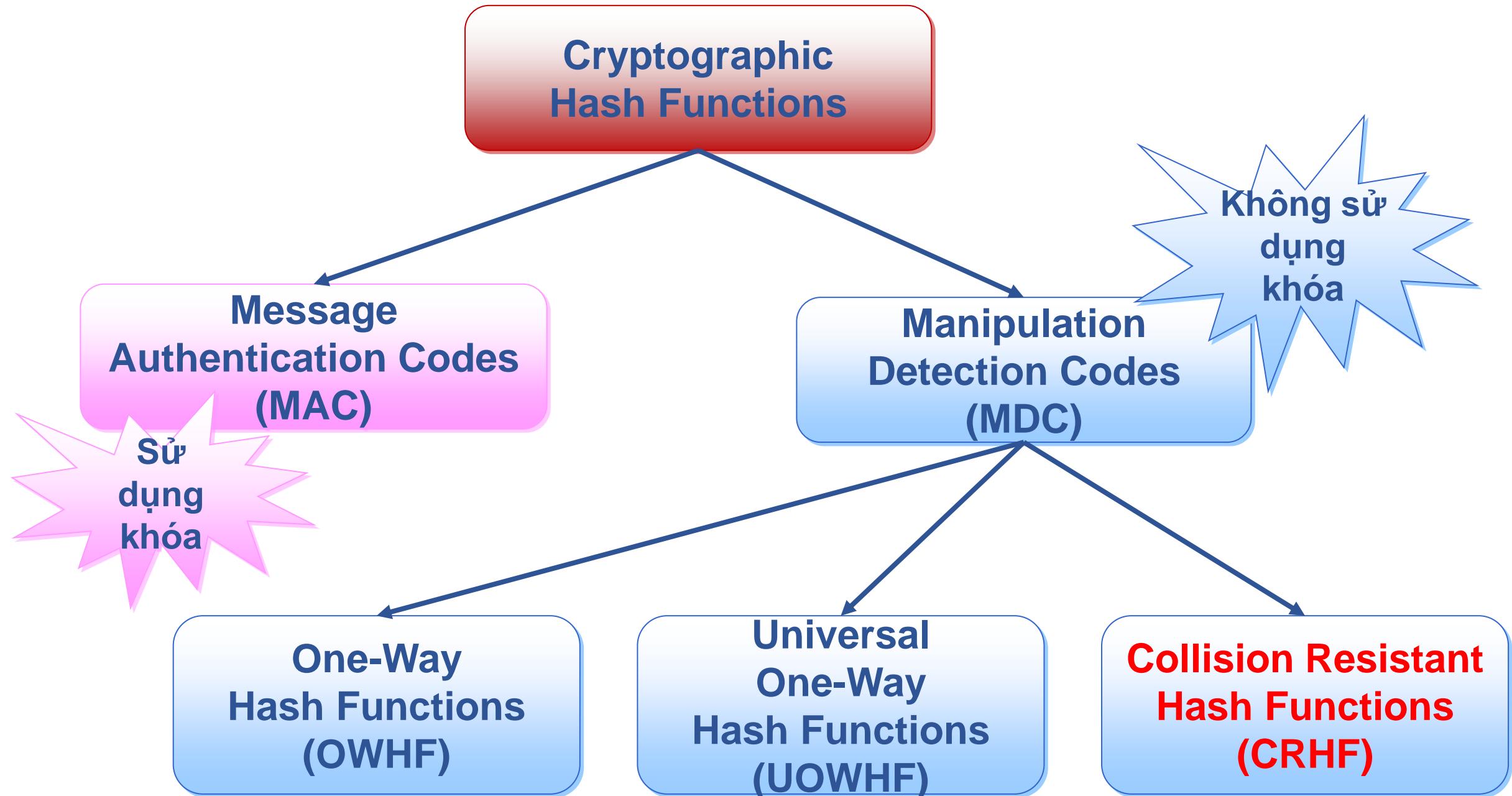


6.1. Các thuộc tính của hàm băm

❑ Tính một chiều

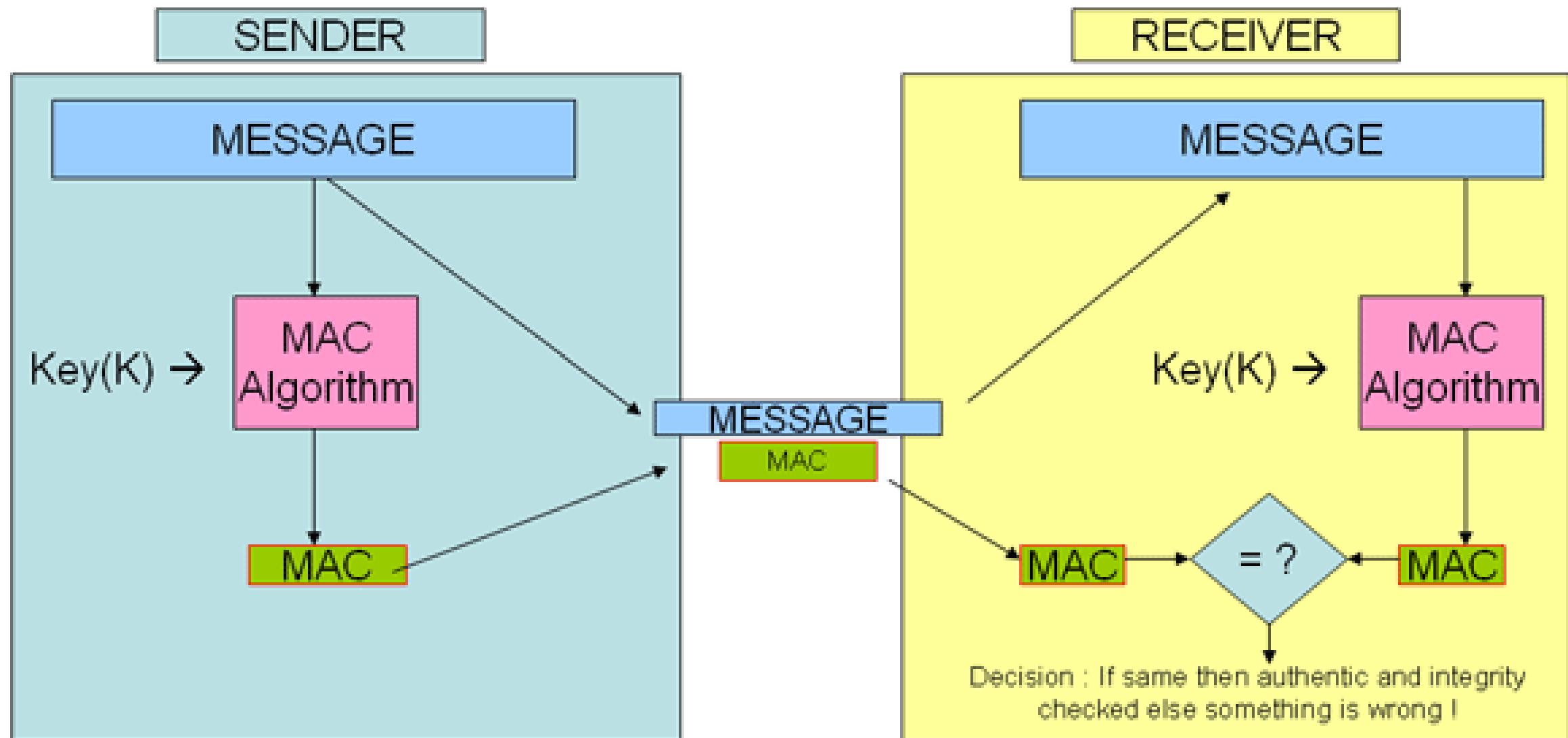
- ❖ Hàm H rất khó bị biến đổi ngược
 - Cho trước chuỗi bit ngẫu nhiên $y \in \{0,1\}^n$, rất khó tìm ra được chuỗi bit x sao cho $H(x)=y$
- ❖ Ví dụ:
 - Brute-force: Với mỗi giá trị x , kiểm tra $H(x)=y$
 - SHA-1 cho kết quả là chuỗi gồm 160-bit.
 - Giả sử phần cứng cho phép thực hiện 2^{34} phép thử trong một giây.
 - Có thể thực hiện 2^{59} phép thử trong một năm.
 - Cần 2^{101} ($\sim 10^{30}$) năm để biến đổi ngược SHA-1 với giá trị ngẫu nhiên y cho trước .

6.2. Phân loại hàm băm mật mã



6.2. Phân loại hàm băm mật mã

Message authentication code (MAC)



Mục đích: xác định nguồn gốc của thông tin

6.3. Một số ứng dụng thực tế của hàm băm

- ❖ Sử dụng trong chứng nhận (Certification).
- ❖ Sử dụng trong định danh chứng thực người dùng (authentication).
- ❖ Sử dụng trong liên lạc an toàn
- ❖ Sử dụng trong email
- ❖ Một số ứng dụng khác
 - Kiểm tra tính toàn vẹn của phần mềm/dữ liệu khi download.
 - Đối sánh CSDL (Database matching)
 - ...

6.4. Mật khẩu người dùng

❖ Lưu trong CSDL: **username + password**

- Kiểm tra: so sánh password của người dùng nhập vào và password đã lưu trong CSDL
→ An toàn? Admin biết password của người dùng!

❖ Lưu trong CSDL: **username + hash (password)**

- Kiểm tra: so sánh
- hash (password người dùng nhập) = hash (password đã lưu)?
→ An toàn hơn!

6.4. Mật khẩu người dùng

- ❖ Lưu trong CSDL:

- username + salt + H với H = hash (password, salt)

- ❖ Kiểm tra: so sánh

- hash (password người dùng nhập, salt) = H?