HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN HỌC

CƠ SỞ AN TOÀN THÔNG TIN

CHƯƠNG 4 – ĐẢM BẢO ATTT DỰA TRÊN MÃ HÓA

Giảng viên: TS. Hoàng Xuân Dậu

Điện thoại/E-mail: dauhx@ptit.edu.vn

Bộ môn: An toàn thông tin - Khoa CNTT1

NỘI DUNG CHƯƠNG 4

1. Khái quát về mã hóa thông tin và ứng dụng
2. Các phương pháp mã hóa
3. Các giải thuật mã hóa
4. Chữ ký số, chứng chỉ số và PKI
5. Các giao thức đảm bảo an toàn thông tin dựa trên mã hóa.

4.1 Khái quát về mã hóa thông tin và ứng dụng

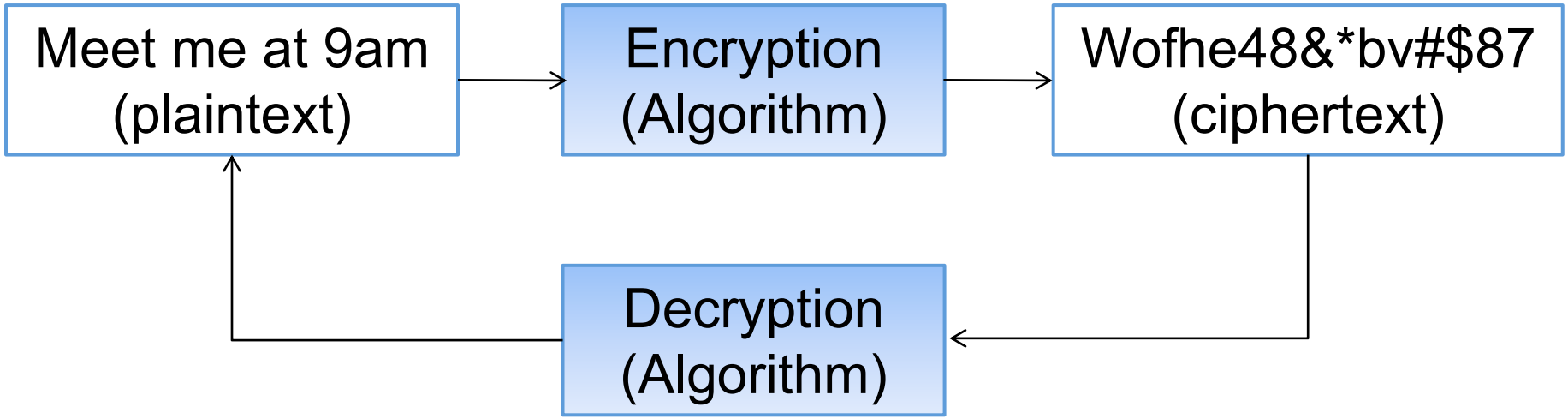
1. Mã hóa thông tin là gì?
2. Vai trò của mã hóa
3. Các thành phần của một hệ mã hóa
4. Lịch sử mã hóa
5. Mã hóa dòng và mã hóa khối
6. Các tiêu chuẩn đánh giá hệ mã hóa
7. Ứng dụng của mã hóa

4.1.1 Mã hóa thông tin là gì?

- Định nghĩa theo Webster's Revised Unabridged Dictionary: cryptography is "the act or art of writing secret characters" – mật mã là một hành động hoặc nghệ thuật viết các ký tự bí mật.
- Định nghĩa theo Free Online Dictionary of Computing: cryptography is "encoding data so that it can only be decoded by specific individuals." – mật mã là việc mã hóa dữ liệu mà nó chỉ có thể được giải mã bởi một số người chỉ định.

4.1.1 Mã hóa thông tin là gì?

- Một hệ mã hóa gồm 2 khâu:
  - Mã hóa (encryption)
  - Giải mã (decryption)



4.1.1 Mã hóa thông tin – Các thuật ngữ

- Thông tin chưa được mã hóa (Unencrypted information) là thông tin ở dạng có thể hiểu được.
  - Cũng được gọi là bản rõ (plaintext hay cleartext)
- Thông tin đã được mã hóa (Encrypted information) là thông tin ở dạng đã bị xáo trộn.
  - Cũng được gọi là bản mã (ciphertext hay encrypted text)

4.1.1 Mã hóa thông tin – Các thuật ngữ

- Mã hóa (Encryption):
  - Là hành động xáo trộn (scrambling) bản rõ để chuyển thành bản mã.
- Giải mã (Decryption):
  - Là hành động giải xáo trộn (unscrambling) bản mã để chuyển thành bản rõ.
- Mã hóa/Giải mã sử dụng một thuật toán (Algorithm) để mã hóa/giải mã thông tin;
  - Thuật toán mã hóa/giải mã có thể giống, hoặc khác nhau.
- Một bộ mã hóa (Cipher) là một giải thuật để mã hóa và giải mã thông tin.

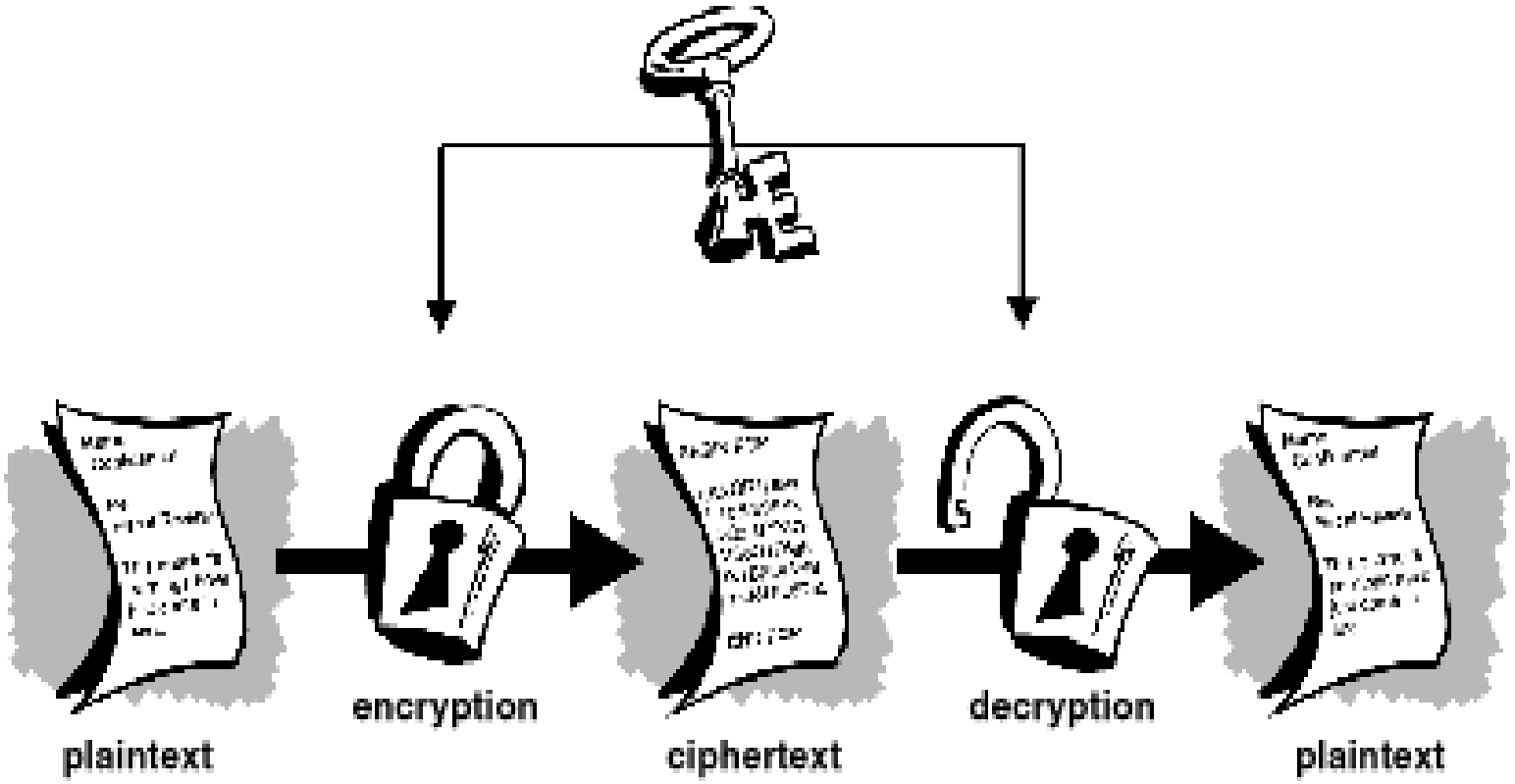
4.1.1 Mã hóa thông tin – Các thuật ngữ

- Khóa/Chìa (Key) là một chuỗi được sử dụng trong giải thuật mã hóa và giải mã.
- Không gian khóa (Keyspace) là tổng số khóa có thể có của một hệ mã hóa.
  - Ví dụ nếu sử dụng khóa kích thước 64 bít → không gian khóa là 2^{64}.

4.1.1 Mã hóa thông tin – Các thuật ngữ

- Mã hóa khóa bí mật (Secret key cryptography):
  - Một khóa được sử dụng cho cả giải thuật mã hóa và giải mã;
  - Khóa này được gọi là khóa bí mật (secret key) hay khóa chia sẻ (shared key).

4.1.1 Mã hóa thông tin – Các thuật ngữ

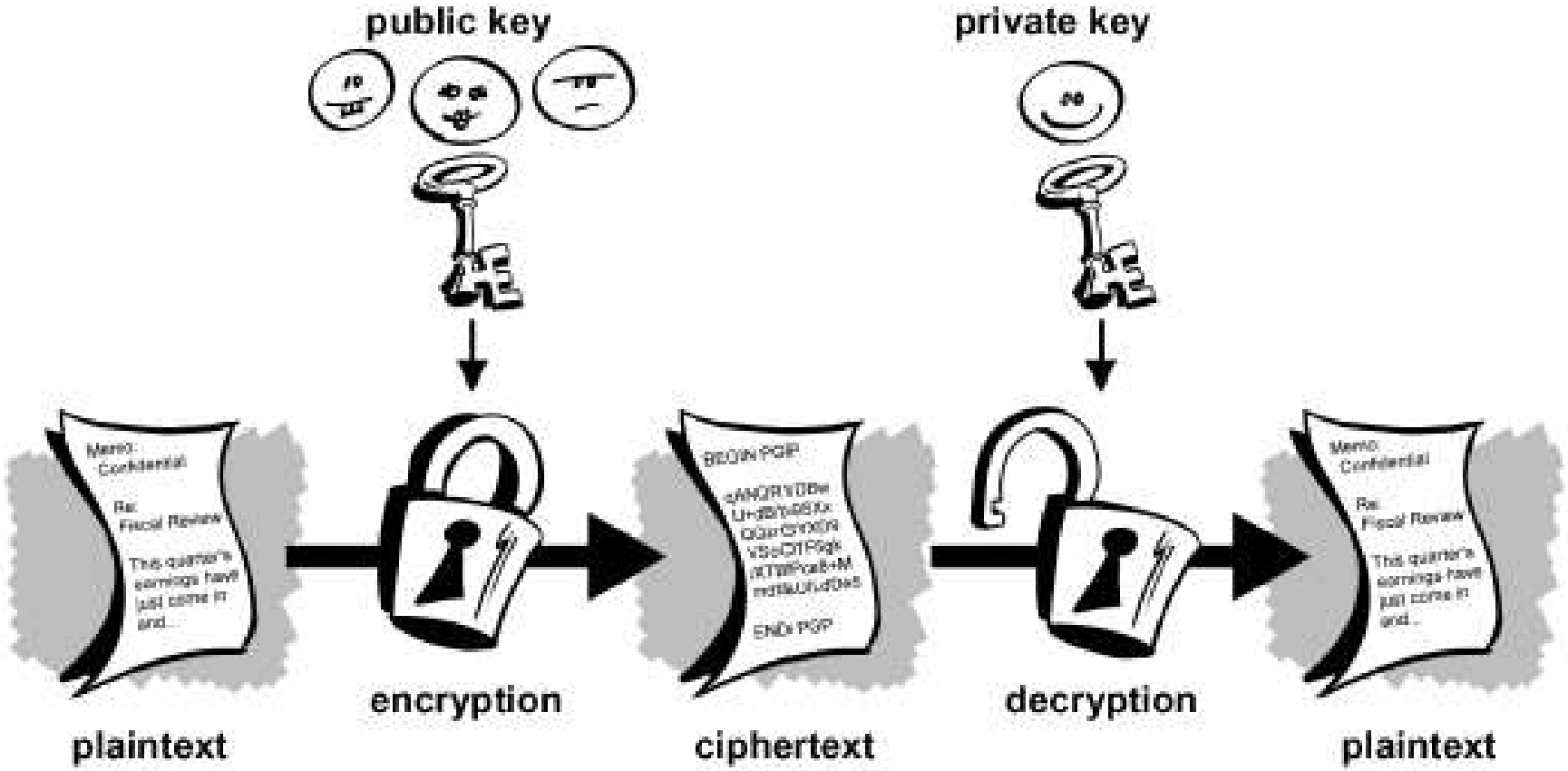


Mã hóa khóa bí mật

4.1.1 Mã hóa thông tin – Các thuật ngữ

- Mã hóa khóa công khai (Public key cryptography):
  - Một cặp khóa được sử dụng, trong đó một khóa để mã hóa, một khóa để giải mã;
  - Khóa để mã hóa được gọi là khóa công khai (public key);
  - Khóa để giải mã được gọi là khóa riêng (private key).

4.1.1 Mã hóa thông tin – Các thuật ngữ



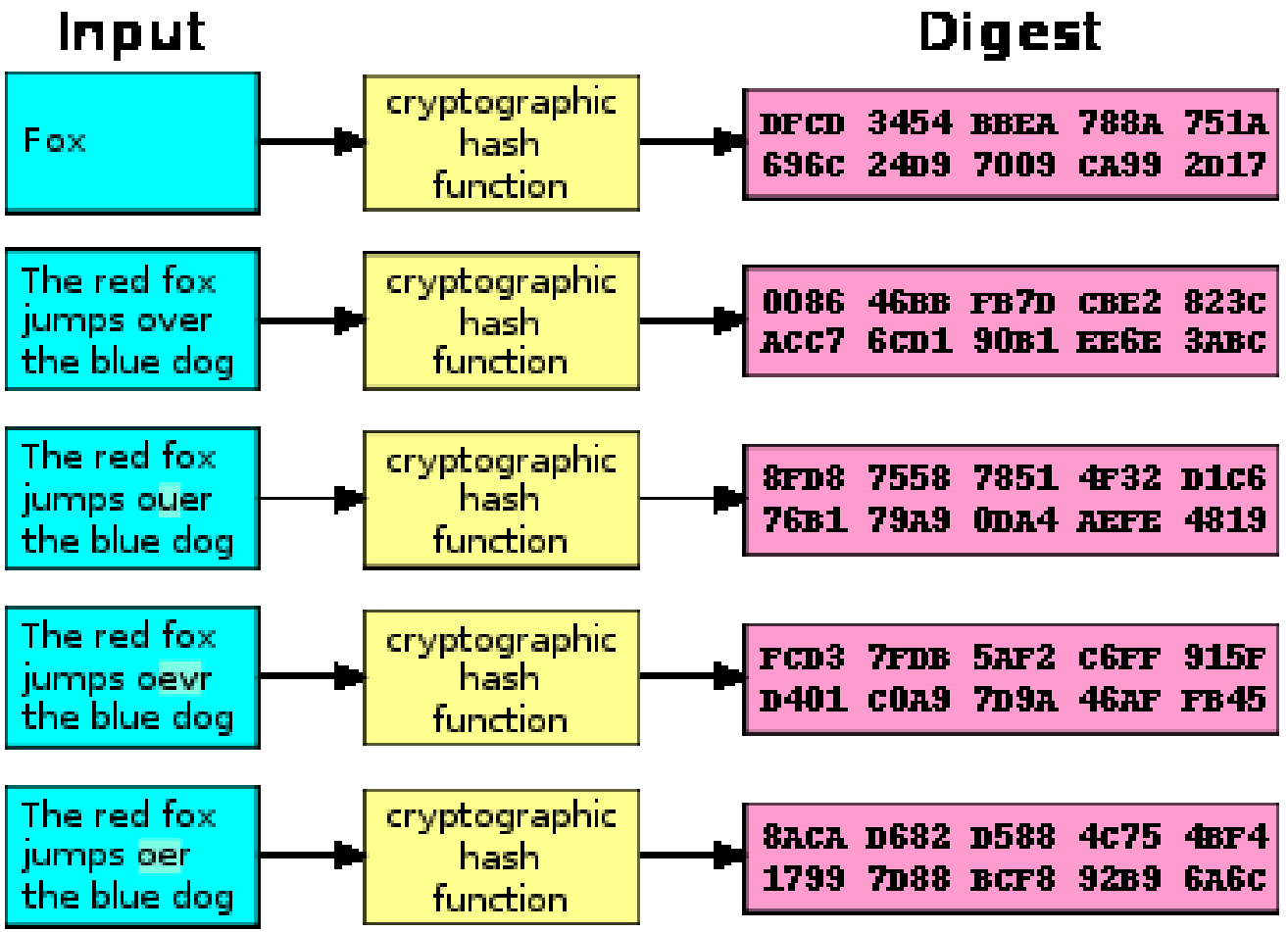
Mã hóa khóa công khai

4.1.1 Mã hóa thông tin – Các thuật ngữ

- Hàm băm (Hash function) là một ánh xạ chuyển các dữ liệu có kích thước thay đổi về dữ liệu có kích thước cố định.
  - Hàm băm 1 chiều (One-way hash function) là hàm băm trong đó việc thực hiện mã hóa tương đối đơn giản, còn việc giải mã thường có độ phức tạp rất lớn, hoặc không khả thi về mặt tính toán.

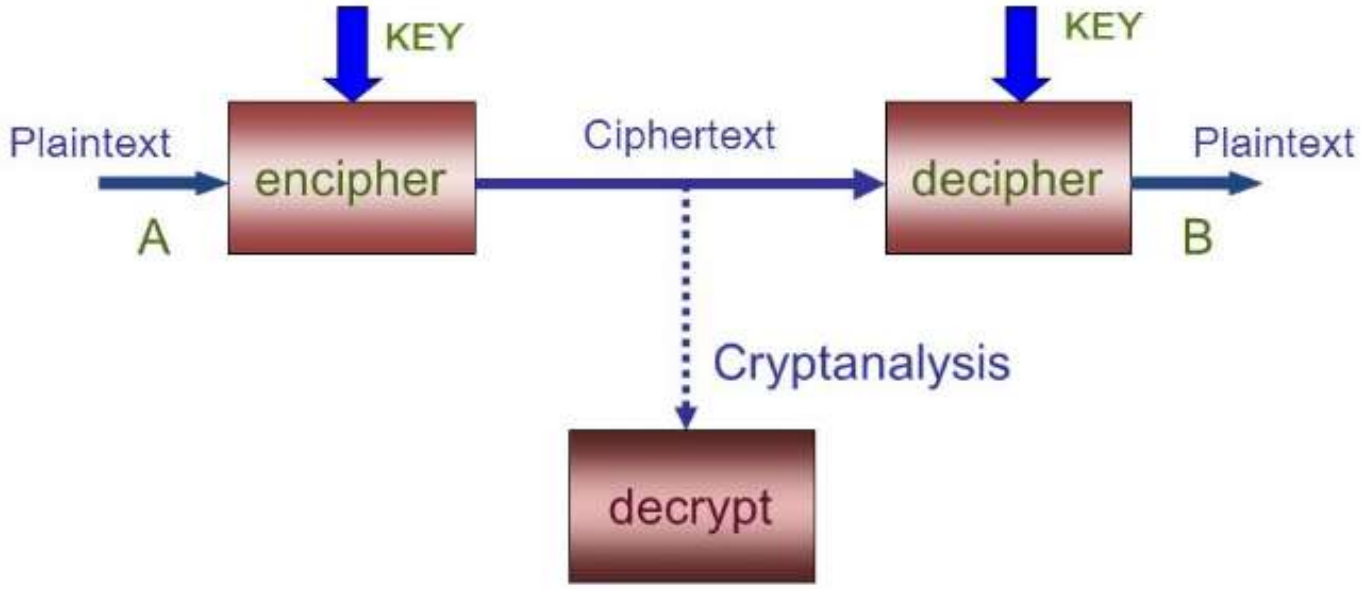
4.1.1 Mã hóa thông tin – Các thuật ngữ

Ví dụ về hàm băm (hash function)



4.1.1 Mã hóa thông tin – Các thuật ngữ

- Phá mã/Thám mã (Cryptanalysis) là quá trình giải mã thông điệp đã bị mã hóa (ciphertext) mà không cần có trước:
  - Thông tin về giải thuật mã hóa (Encryption algorithm) và
  - Thông tin về khóa mã (Key).



4.1.2 Vai trò của mã hóa trong ATTT

- Mã hoá thông tin có thể được sử dụng để đảm bảo an toàn thông tin trên đường truyền với các thuộc tính:
  - Bí mật (confidentiality): đảm bảo chỉ những người có thẩm quyền mới có khả năng truy nhập vào thông tin;
  - Toàn vẹn (integrity): đảm bảo dữ liệu không bị sửa đổi bởi các bên không có đủ thẩm quyền;
  - Xác thực (authentication): thông tin nhận dạng về các chủ thể tham gia phiên truyền thông có thể xác thực;
  - Không thể chối bỏ (non-repudiation): cho phép ngăn chặn một chủ thể chối bỏ hành vi hoặc phát ngôn đã thực hiện.

4.1.3 Các thành phần của một hệ mã hóa

- Một hệ mã hoá (cryptosystem) được cấu thành từ hai thành phần chính:
  - Phương pháp mã hoá, còn gọi là "giải thuật" (Algorithm)
  - Một tập các khoá, còn gọi là không gian khoá (Keyspace)
- Nguyên lý Kerckhoff:
  - "*tính an toàn của một hệ mã hoá không nên phục thuộc vào việc giữ bí mật giải thuật mã hoá, mã chỉ nên phục thuộc vào việc giữ bí mật khoá mã*".

4.1.4 Lịch sử mã hóa

- Các kỹ thuật mã hoá thô sơ đã được người cổ Ai cập sử dụng cách đây 4000 năm.
- Người cổ Hy lạp, Ấn độ cũng đã sử dụng mã hoá cách đây hàng ngàn năm.
- Các kỹ thuật mã hoá chỉ thực sự phát triển mạnh từ thế kỷ 1800 nhờ công cụ toán học, và phát triển vượt bậc trong thế kỷ 20 nhờ sự phát triển của máy tính và ngành CNTT.
- Trong chiến tranh thế giới thứ I và II, các kỹ thuật mã hóa được sử dụng rộng rãi trong liên lạc quân sự sử dụng sóng vô tuyến.
  - Sử dụng các công cụ phá mã/thám mã để giải mã các thông điệp của quân địch.

4.1.4 Lịch sử mã hóa

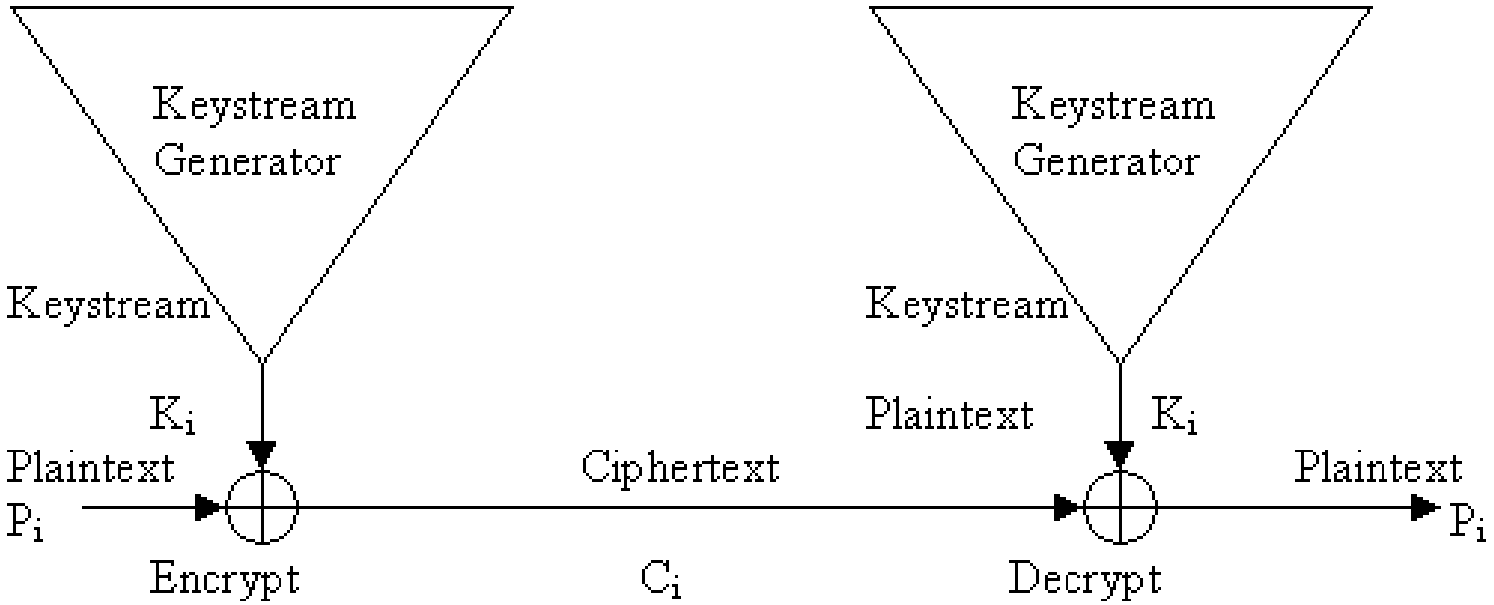
- Năm 1976 chuẩn mã hóa DES (Data Encryption Standard) được cơ quan an ninh quốc gia Mỹ (NSA – National Security Agency) thừa nhận và sử dụng rộng rãi.
- Năm 1976, hai nhà khoa học Whitman Diffie và Martin Hellman đã đưa ra khái niệm mã hóa bất đối xứng (Asymetric key cryptography) hay mã hóa khóa công khai (Public key cryptography) đưa đến những thay đổi lớn trong kỹ thuật mật mã:
  - Các hệ mã hóa khóa công khai hỗ trợ trao đổi khóa dễ dàng hơn;
  - Các hệ mã hóa khóa bí mật gặp khó khăn trong quản lý và trao đổi khóa, đặc biệt khi số lượng người dùng lớn.

4.1.4 Lịch sử mã hóa

- Năm 1977, ba nhà khoa học Ronald Rivest, Adi Shamir, và Leonard Adleman giới thiệu giải thuật mã hóa khóa công khai RSA:
  - RSA trở thành giải thuật mã hóa khóa công khai được sử dụng rộng rãi nhất.
  - RSA có thể vừa được sử dụng để mã hóa thông tin và sử dụng trong chữ ký số.
- Năm 1991, phiên bản đầu tiên của chuẩn bảo mật PGP (Pretty Good Privacy) ra đời.
- Năm 2001, chuẩn mã hóa AES (Advanced Encryption Standard) được xây dựng và sử dụng rộng rãi.

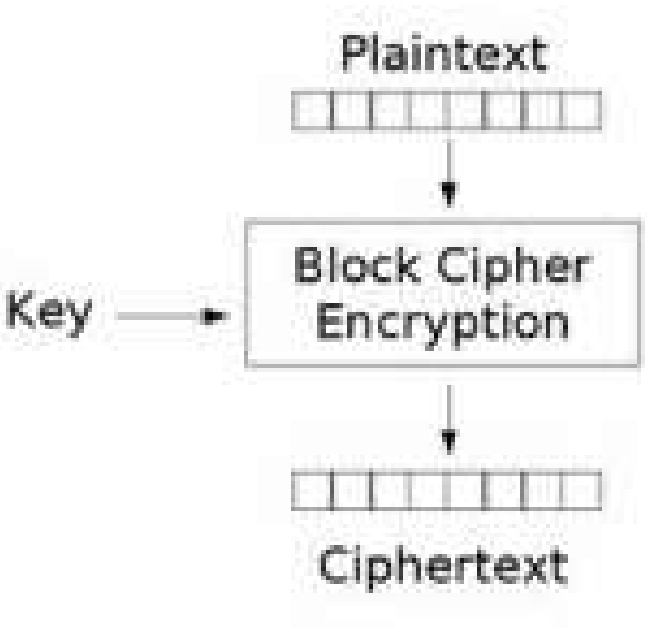
4.1.5 Mã hóa dòng và mã hóa khối

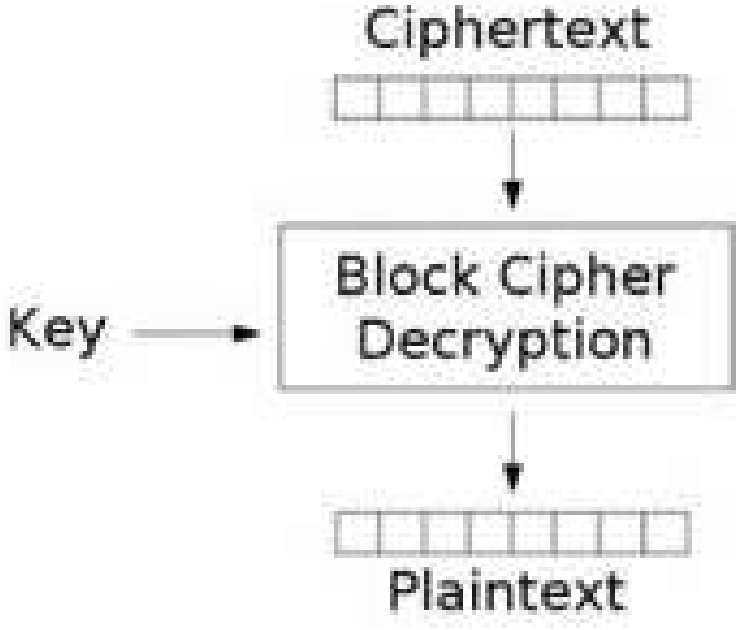
- Mã hóa dòng (Stream cipher) là kiểu mã hóa mà từng bít (hoặc ký tự) của dữ liệu được kết hợp với từng bít (hoặc ký tự) tương ứng của khóa để tạo thành bản mã.



4.1.5 Mã hóa dòng và mã hóa khối

- Mã hóa khối (Block cipher) là kiểu mã hóa mà dữ liệu được chia ra thành từng khối có kích thước cố định để mã hóa.





4.1.5 Mã hóa dòng và mã hóa khối

- Các chế độ hoạt động, hay cách chia khối (Modes of Operation) của mã hóa khối:
  - Chế độ ECB (Electronic Codebook): cùng khối bản rõ đầu vào, khối bản mã giống nhau. Các khối mã hoàn toàn độc lập nhau ($c_j = E_k(x_j)$).
  - Chế độ CBC (Cipher-Block Chaining): cùng khối bản rõ đầu vào, khối bản mã giống nhau với cùng khóa và véc tơ khởi tạo (IV). Khối mã c_j phụ thuộc vào khối rõ x_j và các khối rõ trước đó (x_1-x_{j-1}) thông qua khối mã c_{j-1} ($c_j = E_k(x_j \text{ XOR } c_{j-1})$, $c_0 = IV$).
  - Chế độ CFB (Cipher Feedback): cùng khối bản rõ đầu vào, khối bản mã khác nhau. Khối mã c_j phụ thuộc vào khối rõ x_j và các khối rõ trước đó (x_1-x_{j-1}) thông qua khối mã c_{j-1} ($c_j = E_k(c_{j-1}) \text{ XOR } x_j$, $c_0 = IV$).
  - Chế độ OFB (Output Feedback): cùng khối bản rõ đầu vào, khối bản mã khác nhau. Hệ thống sinh luồng khối khoá và XOR với các khối mã để tạo bản mã. Luồng khóa độc lập với bản rõ.

4.1.6 Các tiêu chuẩn đánh giá hệ mã hóa

- **Độ an toàn** (level of security): thường được đánh giá thông qua số lượng tính toán để có thể phá được hệ mã hoá.
- **Hiệu năng** (performance): có thể được đo bằng tốc độ mã hoá (bits/giây).
- **Tính năng** (functionality): hệ thống có thể được sử dụng cho nhiều mục đích bảo mật.
- **Chế độ hoạt động** (modes of operation): cung cấp các tính năng khác nhau theo chế độ hoạt động.
- **Độ dễ cài đặt** (ease of implementation): độ khó của việc cài đặt thuật toán trong thực tế trên phần cứng hoặc phần mềm.

4.1.7 Ứng dụng của mã hóa

- Các kỹ thuật mã hóa được ứng dụng rộng rãi trong các hệ thống/công cụ/dịch vụ bảo mật:
  - Dịch vụ xác thực (Kerberos, RADIUS,…)
  - Điều khiển truy nhập
  - Các công cụ đánh giá và phân tích logs
  - Các sản phẩm quản lý ATTT
  - Các công cụ cho đảm bảo an toàn cho truyền thông không dây
  - Các nền tảng bảo mật như PKI, PGP
  - Các giao thức bảo mật như SSL/TLS, SSH, SET, IPSec
  - Các hệ thống như VPN.

4.2 Các phương pháp mã hóa

1. Phương pháp thay thế
2. Phương pháp hoán vị
3. Phương pháp XOR
4. Phương pháp Vernam
5. Phương pháp sách hoặc khóa chạy
6. Phương pháp hàm băm

4.2.1 Phương pháp thay thế

- Là phương pháp thay thế một giá trị này bằng một giá trị khác:
  - Thay một ký tự bằng một ký tự khác;
  - Thay một bít bằng một bít khác.
  - Caesar cipher: dịch 3 chữ sang bên phải (A→D, B→E,….)

Bộ chữ gốc	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Bộ chữ mã	DEFGHIJKLMNOPQRSTUVWXYZABC

LOVE → ORYH

4.2.1 Phương pháp thay thế

- Số bộ chữ mã có thể là 1 hoặc nhiều:
  - Một 1 gốc → 1 chữ mã: dễ đoán theo sự lặp lại
  - Một 1 gốc → 1 trong n chữ mã: khó đoán do phức tạp hơn

Plaintext =	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Substitution cipher 1 =	DEFGHIJKLMNOPQRSTUVWXYZABC
Substitution cipher 2 =	GHIJKLMNOPQRSTUVWXYZABCDEF
Substitution cipher 3 =	JKLMNOPQRSTUVWXYZABCDEFGHI
Substitution cipher 4 =	MNOPQRSTUVWXYZABCDEFGHIJKL

Ký tự số 1 dùng bộ mã 1, ký tự 2 dùng bộ mã 2,…

TEXT → WKGF

4.2.2 Phương pháp đổi chỗ

- Phương pháp đổi chỗ hoặc hoán vị (permutation) thực hiện sắp xếp lại các giá trị trong một khối để tạo bản mã:
  - Có thể thực hiện với từng bít hoặc từng byte (ký tự).

Khóa đổi chỗ (khối 8 phần tử) tính từ bên phải

Key $1\rightarrow4, 2\rightarrow8, 3\rightarrow1, 4\rightarrow5, 5\rightarrow7, 6\rightarrow2, 7\rightarrow6, 8\rightarrow3$

Bit locations:	87654321 87654321 87654321 87654321
Plaintext 8-bit blocks:	00100101|01101011|10010101|01010100
Ciphertext:	00001011|10111010|01001101|01100001

4.2.2 Phương pháp đổi chỗ

- Thực hiện đổi chỗ ký tự trong khối 8 ký tự, tính từ bên phải:

Letter locations:	87654321 | 87654321 | 87654321 | 87654321 |
Plaintext:	SACKGAUL | SPARENOO | NE | |
Key:	1→4, 2→8, 3→1, 4→5, 5→7, 6→2, 7→6, 8→3
Ciphertext:	UKAGLSCA | ORPEOSAN | E N | |

4.2.2 Phương pháp XOR

- Phương pháp XOR sử dụng phép toán logic XOR để tạo bản mã:
  - Từng bít của bản rõ được XOR với bít tương ứng của khóa.

First Bit	Second Bit	Result
0	0	0
0	1	1
1	0	1
1	1	0

Bảng giá trị chân thực của XOR

4.2.3 Phương pháp XOR

- Ví dụ: mã hóa từ CAT (biểu diễn theo mã ASCII là 01000011 01000001 01010100) sử dụng khóa là "V" (01010110)

Text Value	Binary Value
CAT as bits	0 1 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 0
VVV as key	0 1 0 1 0 1 1 0 0 1 0 1 0 1 1 0 0 1 0 1 0 1 1 0
Cipher	0 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 0 1 0

4.2.4 Phương pháp Vernam

- Phương pháp Vernam sử dụng một tập ký tự để nối vào các ký tự của bản rõ để tạo bản mã.
  - Mỗi ký tự trong tập chỉ dùng 1 lần trong một tiến trình mã hóa (được gọi là one-time pad).
- Ví dụ: với bộ chữ tiếng Anh có 26 chữ
  - Các ký tự của bản rõ được chuyển thành số trong khoảng 1-26;
  - Cộng giá trị của ký tự với giá trị tương ứng trong tập nối thêm;
  - Nếu giá trị cộng lớn hơn 26→ đem trừ cho 26.
  - Đây là phép lấy modulo (phần dư).

4.2.4 Phương pháp Vernam

Plaintext:	S	A	C	K	G	A	U	L	S	P	A	R	E	N	O	O	N	E
Plaintext value:	19	01	03	11	07	01	21	12	19	16	01	18	05	14	15	15	14	05
One-time pad text:	F	P	Q	R	N	S	B	I	E	H	T	Z	L	A	C	D	G	J
One time pad value:	06	16	17	18	14	19	02	09	05	08	20	26	12	01	03	04	07	10
Sum of plaintext and pad:	25	17	20	29	21	20	23	21	24	24	21	44	17	15	18	19	21	15
After modulo Subtraction:				03								18						
Ciphertext:	Y	Q	T	C	U	T	W	U	X	X	U	R	Q	O	R	S	U	O

Tiến trình mã hóa sử dụng phương pháp Vernam

4.2.5 Phương pháp sách hoặc khóa chạy

- Phương pháp sách hoặc khóa chạy thường được dùng trong các bộ phim trinh thám, trong đó việc mã hóa và giải mã sử dụng các khóa mã chứa trong các cuốn sách.
- Ví dụ: với bản mã là 259,19,8;22,3,8;375,7,4;394,17,2 và cuốn sách được dùng là "A Fire Up on the Deep":
  - Trang 259, dòng 19, từ thứ 8 → sack
  - Trang 22, dòng 3, từ thứ 8 → island
  - Trang 375, dòng 7, từ thứ 4 → sharp
  - Trang 394, dòng 17, từ thứ 2 → path
  - Bản rõ tương ứng của bản mã "259,19,8;22,3,8;375,7,4;394,17,2 " là "sack island sharp path".

4.2.6 Phương pháp hàm băm

- 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.

4.2.6 Phương pháp hàm băm

- Một số hàm băm thông dụng:
  - MD2, MD4, MD5 (128 bit)
  - MD6 (0-512 bit)
  - SHA0, SHA1 (160 bit)
  - SHA2 (SHA256, SHA384, SHA512), SHA3
  - CRC32 (32 bit)

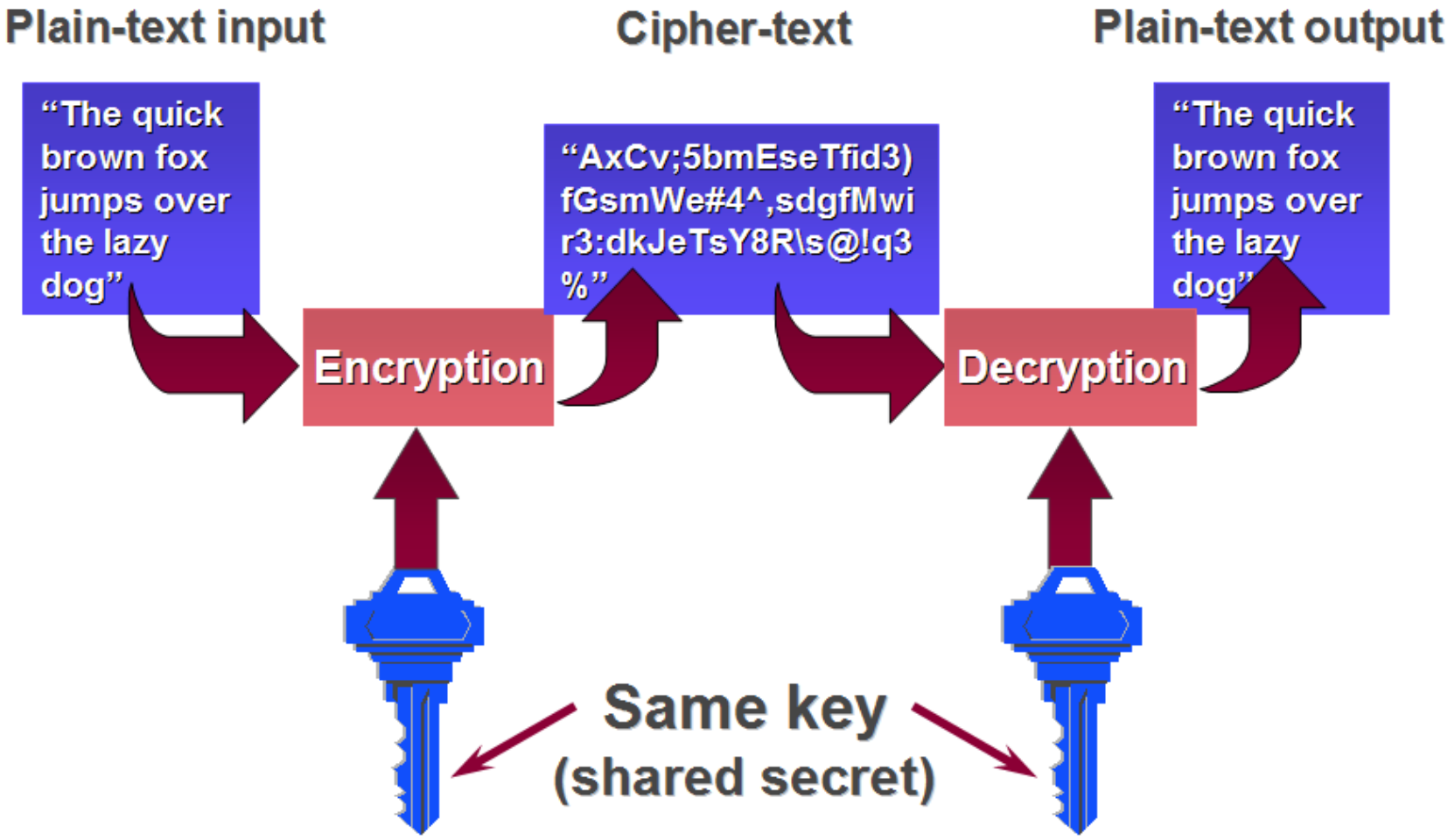
4.3 Các giải thuật mã hóa

1. Các giải thuật mã hóa khóa đối xứng
   - DES, Triple-DES
   - AES, IDEA
   - Blowfish, Twofish
   - RC4, RC5
2. Các giải thuật mã hóa khóa bất đối xứng
   - RSA
   - Rabin
   - ElGamal
3. Các hàm băm
   - MD2, MD4, MD5, MD6
   - SHA0, SHA1, SHA2, SHA3

4.3.1 Các giải thuật mã hóa khóa đối xứng

- Các giải thuật mã hóa khóa đối xứng (symetric key encryption)
  - Còn gọi là mã hóa khóa riêng hay bí mật (secret/private key encryption):
  - Sử dụng một khóa (key) duy nhất cho cả quá trình mã hóa và giải mã.
- Đặc điểm:
  - Kích thước khóa tương đối ngắn (64, 128, 192, 256 bít)
  - Tốc độ nhanh
  - Độ an toàn cao
  - Khó khăn trong quản lý và phân phối khóa.

4.3.1 Các giải thuật mã hóa khóa đối xứng



4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

- DES (Data Encryption Standard) được sử dụng phổ biến:
  - DES được phát triển tại IBM vào đầu những năm 1970;
  - Được thừa nhận là chuẩn mã hóa tại Mỹ (NSA) vào năm 1976;
  - DES được sử dụng rộng rãi trong những năm 70 và 80.
- Hiện nay DES không được coi là an toàn do:
  - Không gian khóa nhỏ (khóa 64 bít, trong đó thực sử dụng 56 bít)
  - Tốc độ tính toán của các hệ thống máy tính ngày càng nhanh.

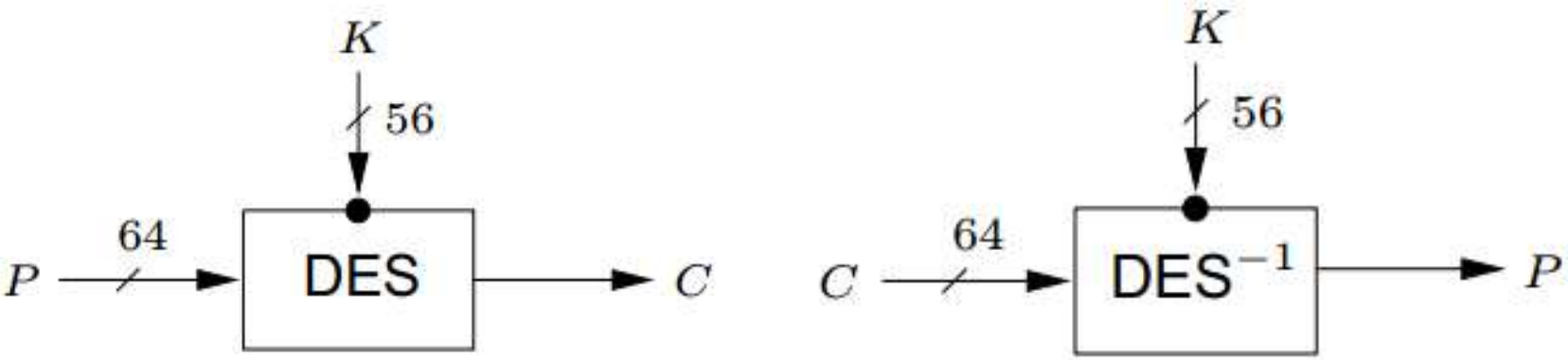
4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

❖ Đặc điểm của DES:

- Là dạng mã hóa khối, kích thước khối vào 64 bít
- Khóa 64 bít, trong đó thực sử dụng 56 bít, 8 bít dùng cho kiểm tra chẵn lẻ
- DES sử dụng chung một giải thuật cho cả hai khâu mã hóa và giải mã.

4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

❖ Mã hóa và giải mã một khối dữ liệu với DES



plaintext P

ciphertext C

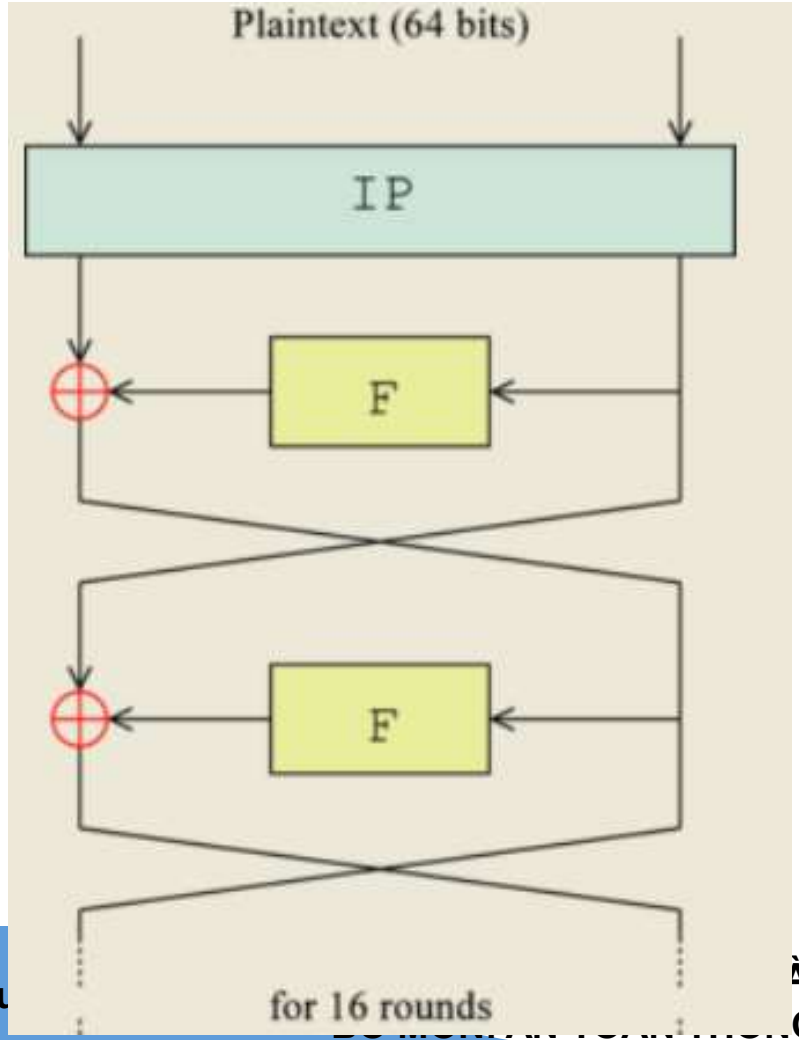
key K

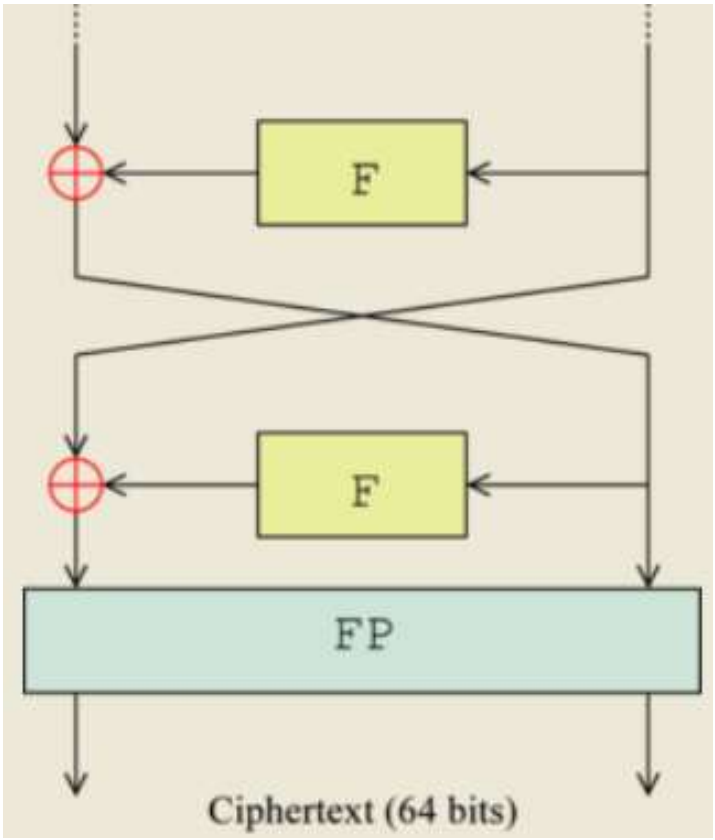
4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

- Các bước thực hiện mã hóa của DES với mỗi khối dữ liệu 64 bít:
  - Bước hoán vị khởi tạo (IP – Initial Permutation);
  - 16 vòng lặp chính thực hiện xáo trộn dữ liệu theo hàm Feistel (F);
  - Bước hoán vị kết thúc (FP – Final Permutation).
- Sử dụng phép $\oplus$ (XOR) để kết hợp trong quá trình lặp.

4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

- Tiến trình mã hóa một khối dữ liệu với DES

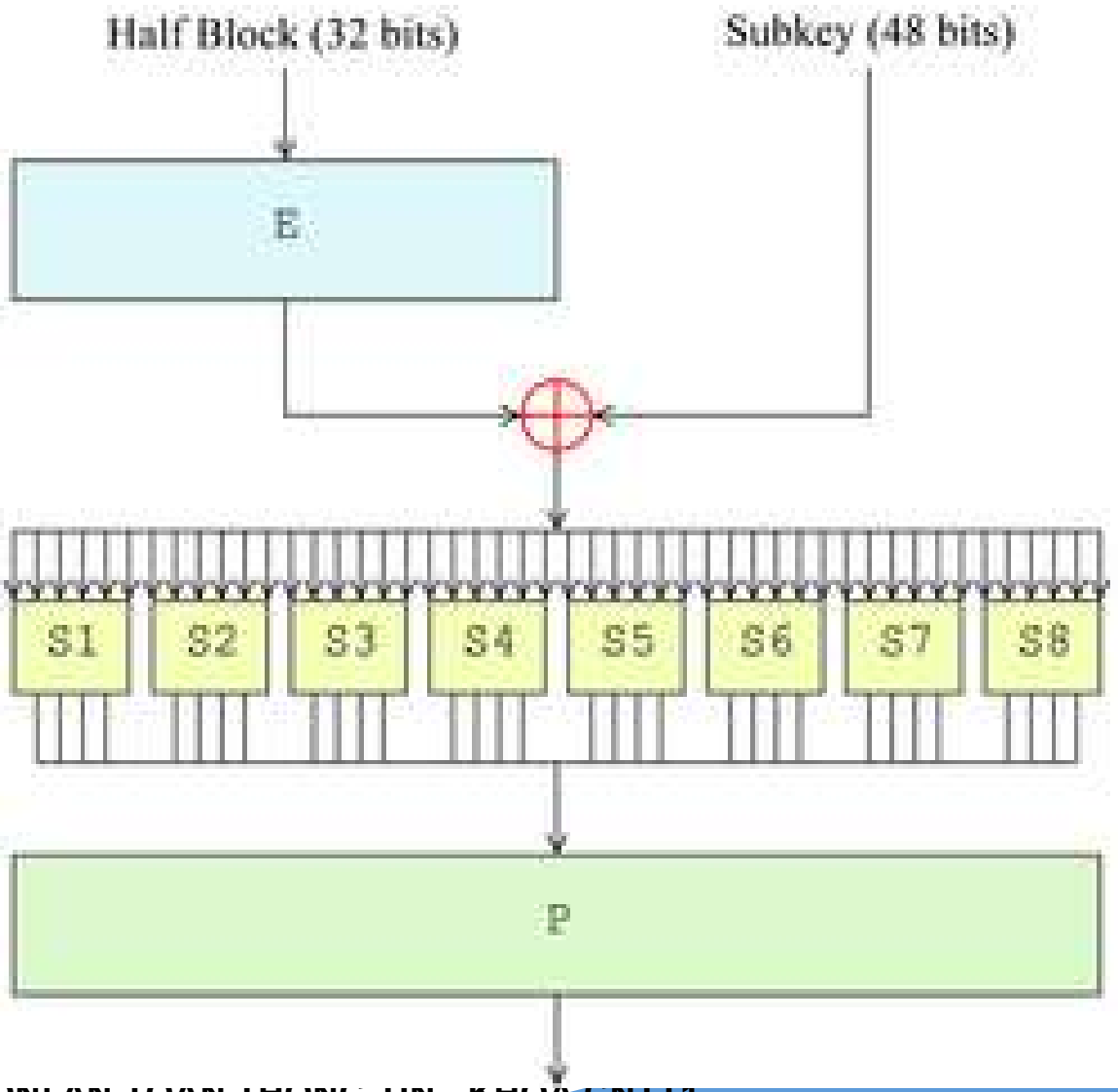




4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

Các bước thực hiện hàm F (Fiestel) của DES:

- E (Expansion) – mở rộng
- ⊕: trộn với một phần khóa
- S_i (Substitution) - thay thế
- P – Hoán vị.

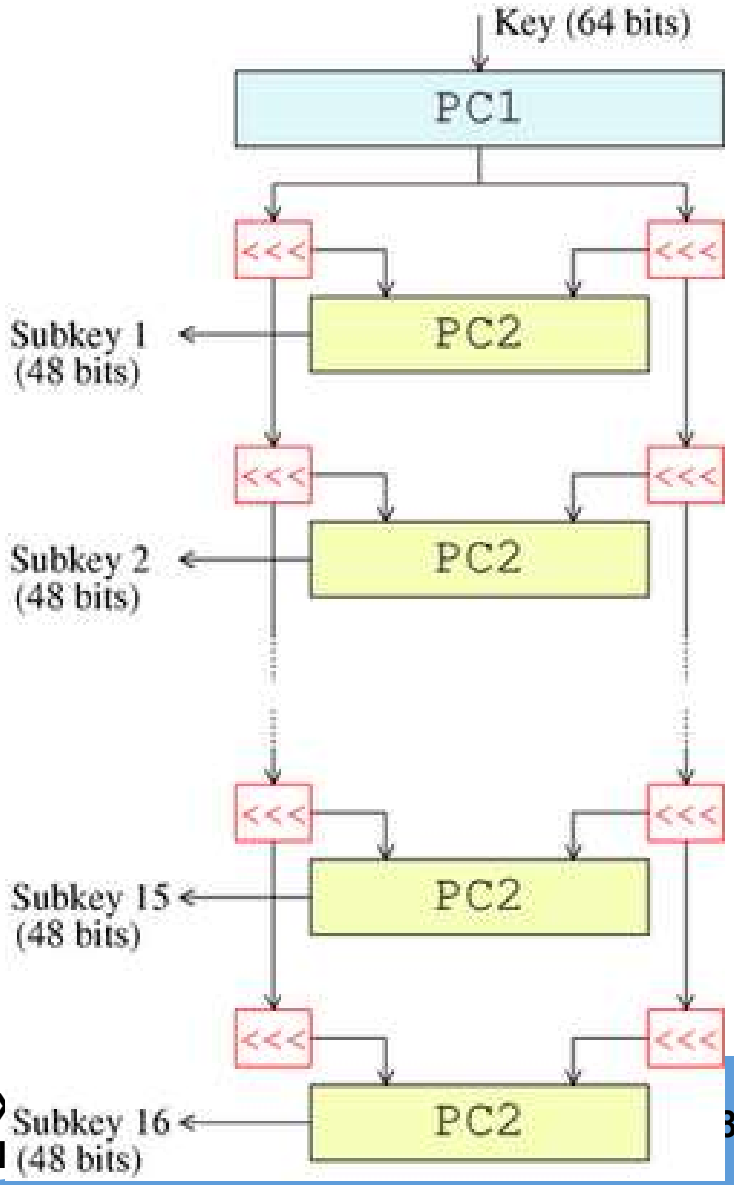


4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

- Chia khối 64 bít thành 2 khối 32 bít và xử lý lần lượt.
- Các bước thực hiện hàm F (Fiestel) với khối dữ liệu 32 bít của DES:
  - E (Expansion): thực hiện mở rộng 32 bít đầu vào thành 48 bít bằng cách nhân đôi một nửa số bit.
  - $\oplus$: Trộn 48 bit ở bước E với khóa phụ 48 bít. Có 16 khóa phụ được tạo từ khóa chính để sử dụng cho 16 vòng lặp.
  - S_i (Substitution): Khối dữ liệu 48 bit được chia thành 8 khối 6 bít và được chuyển cho các bộ thay thế (S1-S8).
    - Mỗi bộ thay thế sử dụng phép chuyển đổi phi tuyến tính để chuyển 6 bit đầu vào thành 4 bit đầu ra theo bảng tham chiếu. Các bộ thay thế là thành phần nhân an ninh (security core) của DES.
  - P: 32 bít đầu ra từ các bộ thay thế được sắp xếp bằng phép hoán vị cố định (fixed permutation) cho ra đầu ra 32 bít.

4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

- Tạo bộ khóa phụ cho 16 vòng lặp:
  - 56 bít khóa được chọn từ khóa 64 bit ban đầu bởi PC1 (Permuted Choice 1). 8 bit còn lại được hủy hoặc dùng để kiểm tra chẵn lẻ;
  - 56 bít được chia thành 2 phần 28 bit, mỗi phần được xử lý riêng;
  - Mỗi phần được quay trái 1 hoặc 2 bit.
  - Hai phần được ghép lại và 48 bit được chọn làm khóa phụ 1 bởi PC2;
  - Lặp lại bước trên để tạo 15 khóa phụ còn lại.



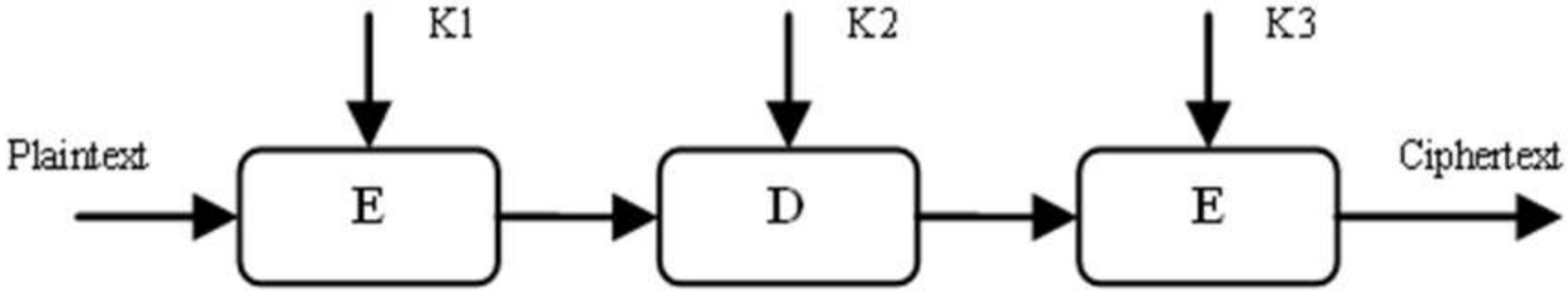
4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

- Giải mã trong DES:
  - Có thể sử dụng giải thuật mã hóa DES để giải mã;
  - Các bước thực hiện giống quá trình mã hóa;
  - Các khóa phụ sử dụng cho các vòng lặp được sử dụng theo trật tự ngược lại: Khóa phụ 16, 15,…, 2, 1 cho các vòng 1, 2,…, 15, 16 tương ứng.

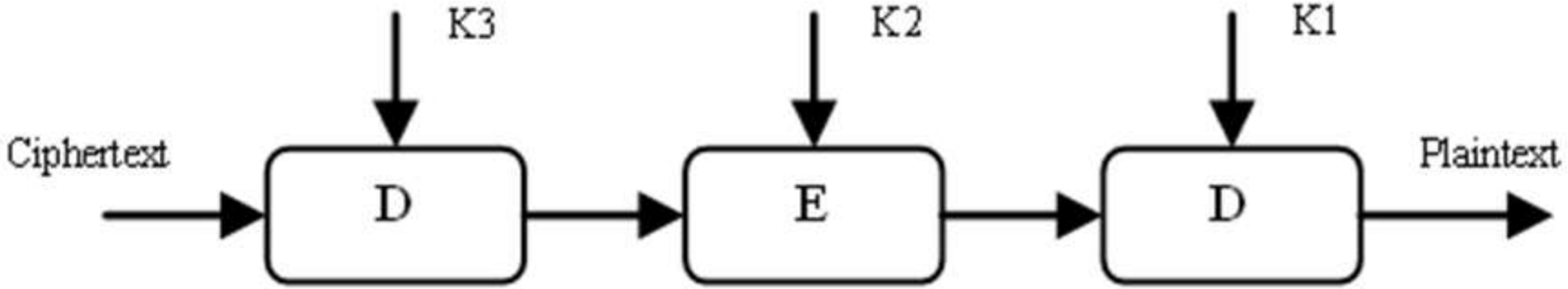
4.3.1 Các giải thuật mã hóa khóa đối xứng – Triple DES

- Triple DES (3-DES) còn được gọi là Triple Data Encryption Algorithm (TDEA hoặc Triple DEA) được phát triển từ DES bằng cách áp dụng DES 3 lần cho mỗi khối dữ liệu;
- Triple DES sử dụng bộ 3 khóa DES: K1, K2, K3, mỗi khóa kích thước hiệu dụng 56 bít;
- Các lựa chọn bộ khóa:
  - Lựa chọn 1: cả 3 khóa độc lập (168 bít)
  - Lựa chọn 2: K1 và K2 độc lập, K3 = K1 (112 bít)
  - Lựa chọn 3: 3 khóa giống nhau, K1 = K2 = K3 (56 bít).
- Kích thước khối dữ liệu vào: 64 bít.

4.3.1 Các giải thuật mã hóa khóa đối xứng – Triple DES



Encryption



Decryption

4.3.1 Các giải thuật mã hóa khóa đối xứng – Triple DES

- Giải thuật mã hóa:
  - ciphertext = $E_{K3}(D_{K2}(E_{K1}(\text{plaintext})))$
    → Mã hóa bằng khóa K1, giải mã bằng K2 và mã hóa bằng K3.
- Giải thuật giải mã:
  - plaintext = $D_{K1}(E_{K2}(D_{K3}(\text{ciphertext})))$
    → Giải mã bằng K3, mã hóa bằng K2 và giải mã bằng K1.

4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

- AES (Advanced Encryption Standard) là một chuẩn mã hóa dữ liệu được NIST công nhận năm 2001;
- AES được xây dựng dựa trên Rijndael cipher phát triển bởi 2 nhà mật mã học người Bỉ là Joan Daemen và Vincent Rijmen;
  - Rijndael cipher là bộ mã hóa được lựa chọn để xây dựng AES sau khi giành chiến thắng trong cuộc thi tuyển chọn bộ mã hóa làm chuẩn mã hóa mới thay cho DES.
  - AES về cơ bản giống Rijndael cipher.

4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

❖ Đặc điểm của AES:

- Kích thước khối dữ liệu của AES là 128 bít;
- Kích thước khóa có thể là 128, 192, hoặc 256 bit;
- AES được thiết kế dựa trên mạng hoán vị-thay thế (substitution-permutation network);
  - Có thể đạt tốc độ cao trên cả cài đặt phần mềm và phần cứng.

4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

- AES vận hành dựa trên một ma trận 4x4, được gọi là *state* (trạng thái);
- Kích thước của khóa quyết định số vòng lặp chuyển đổi cần thực hiện để chuyển bản rõ thành bản mã:
  - 10 vòng lặp với khóa 128 bít;
  - 12 vòng lặp với khóa 192 bít;
  - 14 vòng lặp với khóa 256 bít.

4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

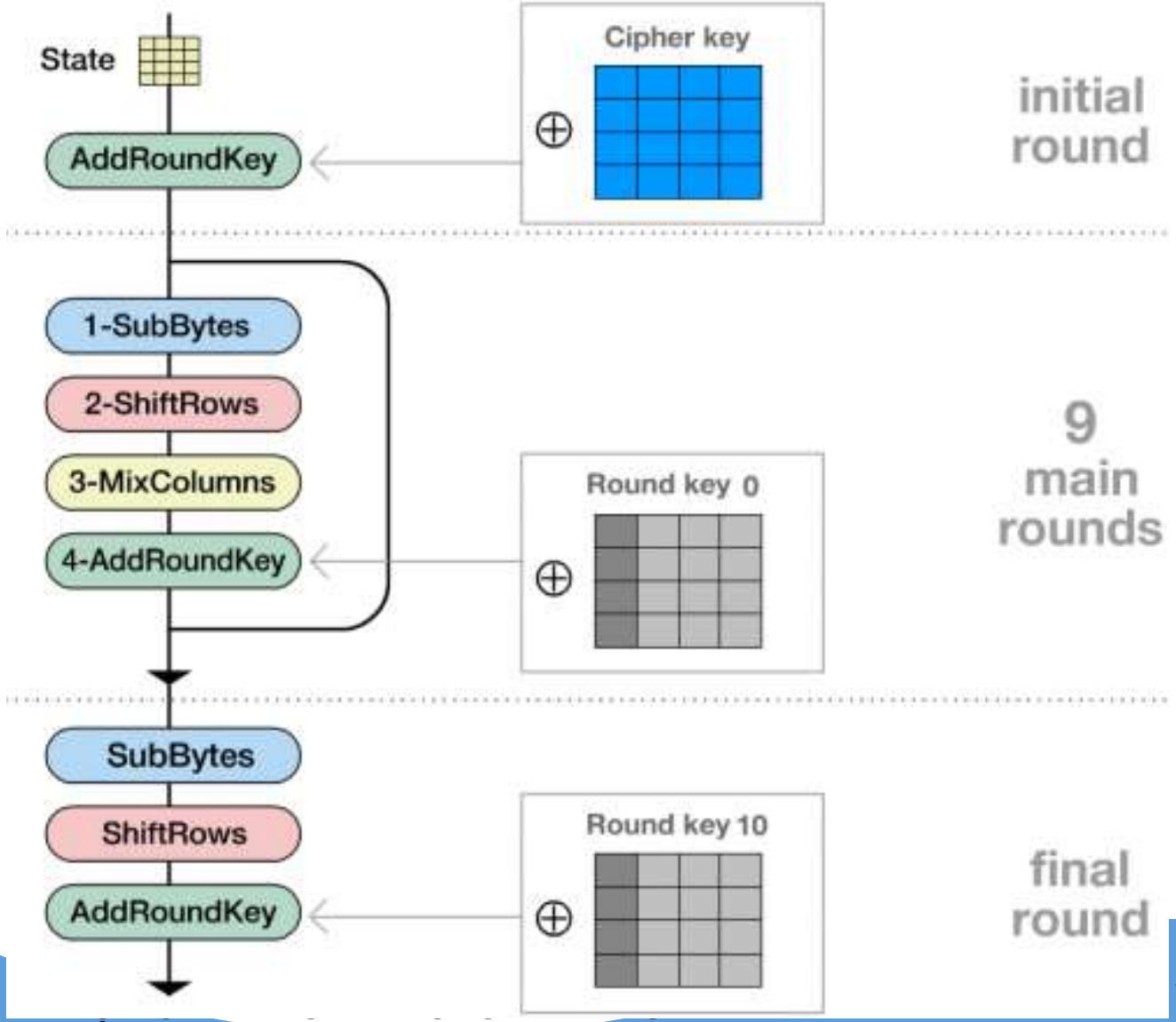
- Mô tả khái quát giải thuật AES:
  1. Mở rộng khóa (KeyExpansion): các khóa phụ dùng trong các vòng lặp được sinh ra từ khóa chính AES sử dụng thủ tục sinh khóa Rijndael.
  2. Vòng khởi tạo (InitialRound)
     a) AddRoundKey: Mỗi byte trong *state* được kết hợp với khóa phụ sử dụng XOR.

4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

- Mô tả khái quát giải thuật AES:
  3. Các vòng lặp chính (Rounds)
     a) SubBytes: bước thay thế phi tuyến tính, trong đó mỗi byte trong *state* được thay thế bằng một byte khác sử dụng bảng tham chiếu;
     b) ShiftRows: bước đổi chỗ, trong đó mỗi dòng trong *state* được dịch một số bước theo chu kỳ;
     c) MixColumns: trộn các cột trong *state*, kết hợp 4 bytes trong mỗi cột.
     d) AddRoundKey.
  4. Vòng cuối (Final Round - không MixColumns)
     a) SubBytes;
     b) ShiftRows;
     c) AddRoundKey.

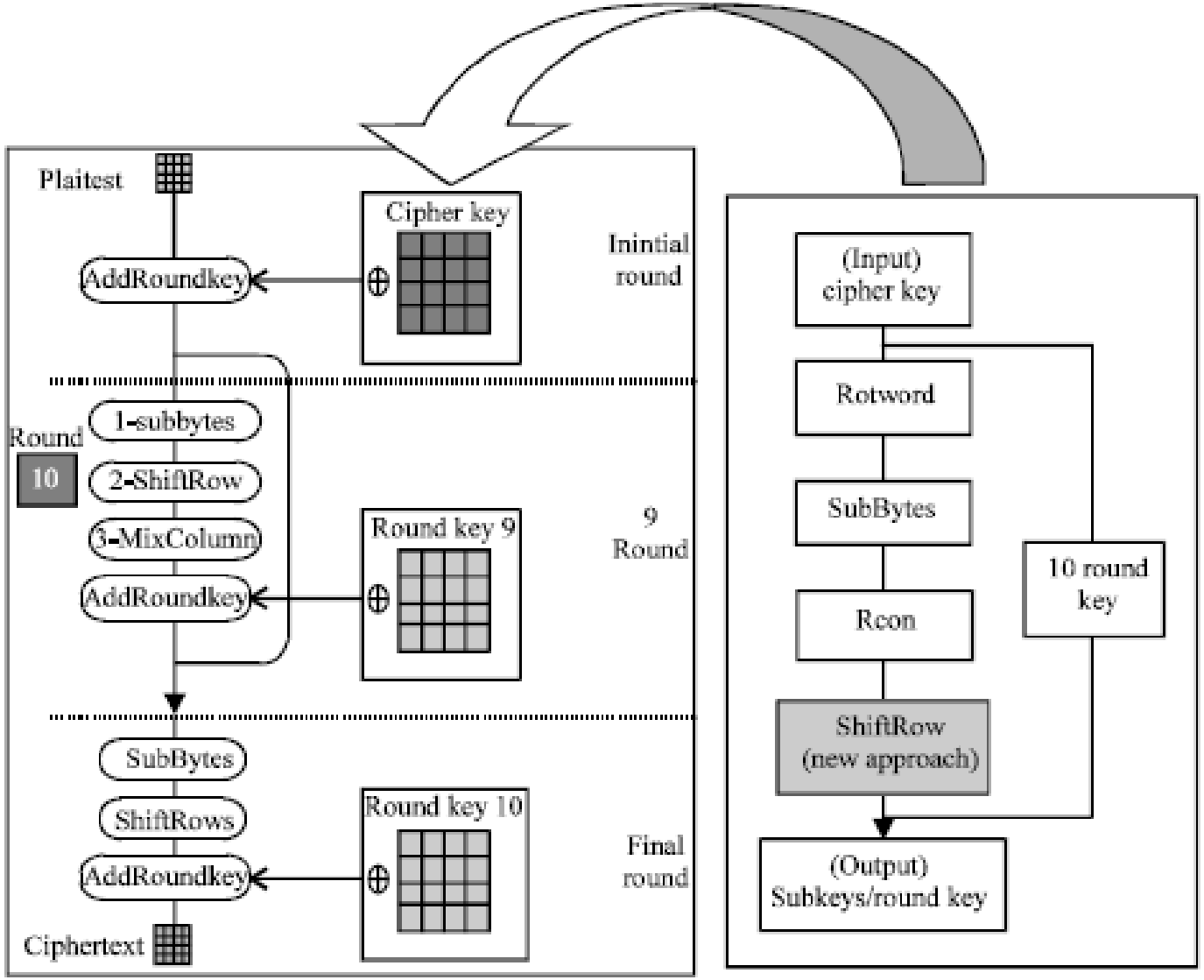
4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

Các bước xử lý chính của AES



4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

Các bước xử lý chính của AES

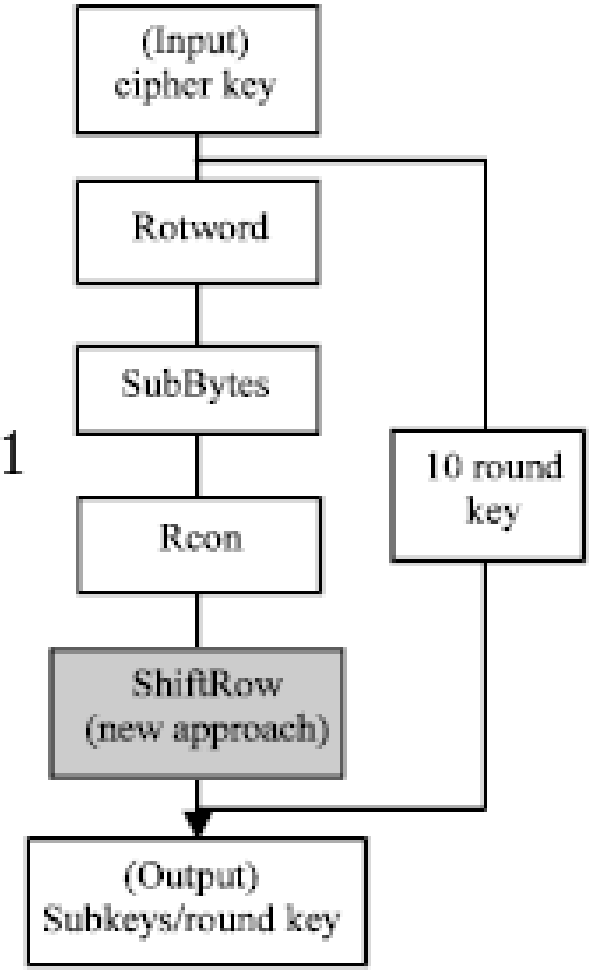


4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

- Mở rộng khóa sử dụng thủ tục sinh khóa Rijndael:
  - Rotword: quay trái 8 bít;
  - SubBytes
  - Rcon: tính toán giá trị Rcon(i)

$$\text{rcon}(i) = x^{(i-1)} \mod x^8 + x^4 + x^3 + x + 1$$

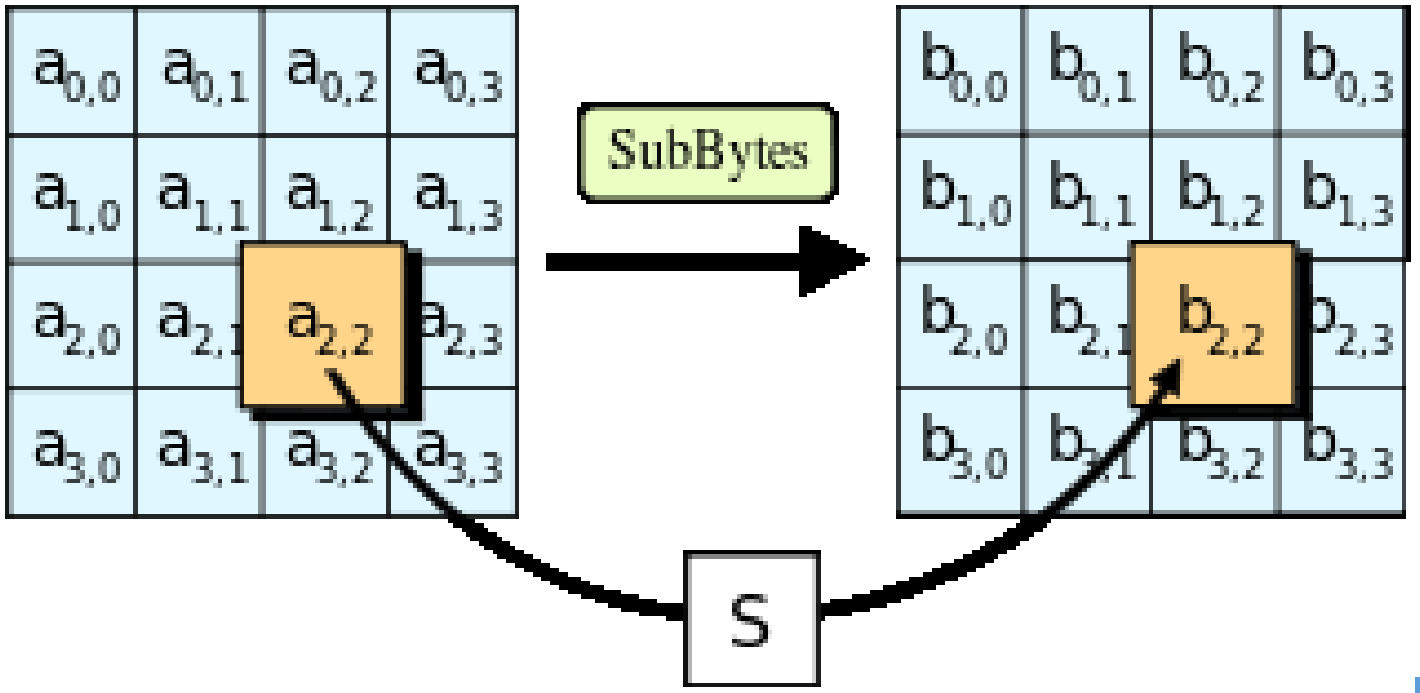
  - ShiftRow



4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

❖ Bước SubBytes:

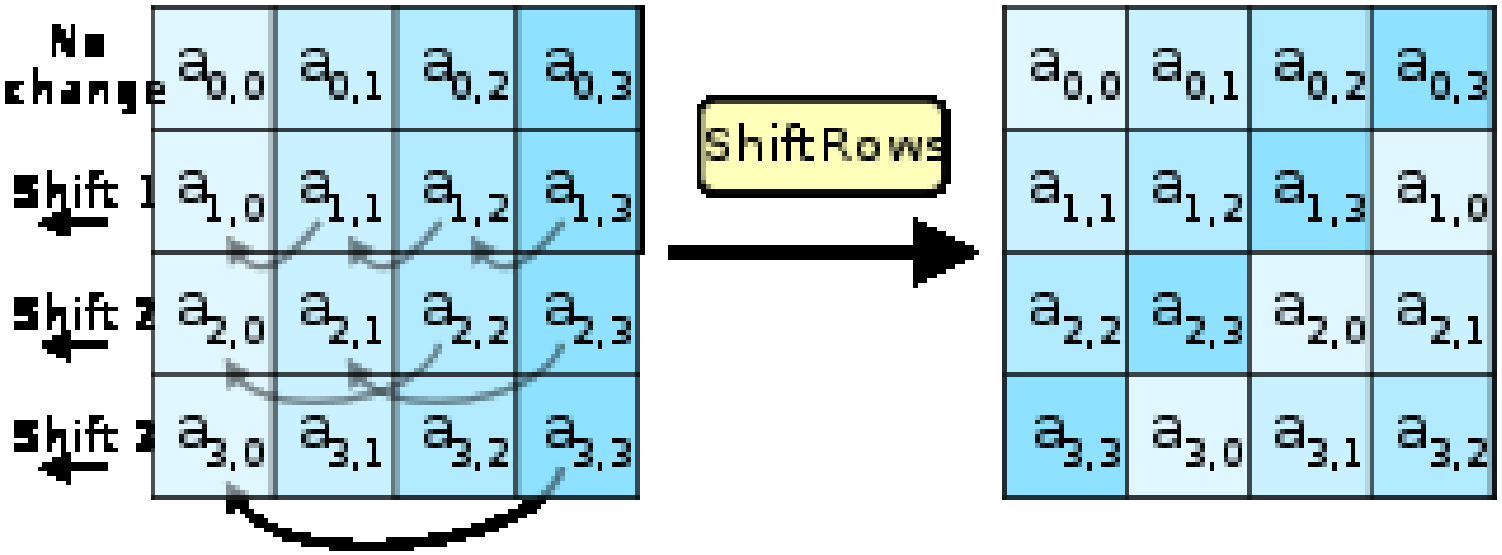
- Mỗi byte trong ma trận state được thay thế bởi 1 byte trong Rijndael S-box, hay $b_{ij} = S(a_{ij})$



4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

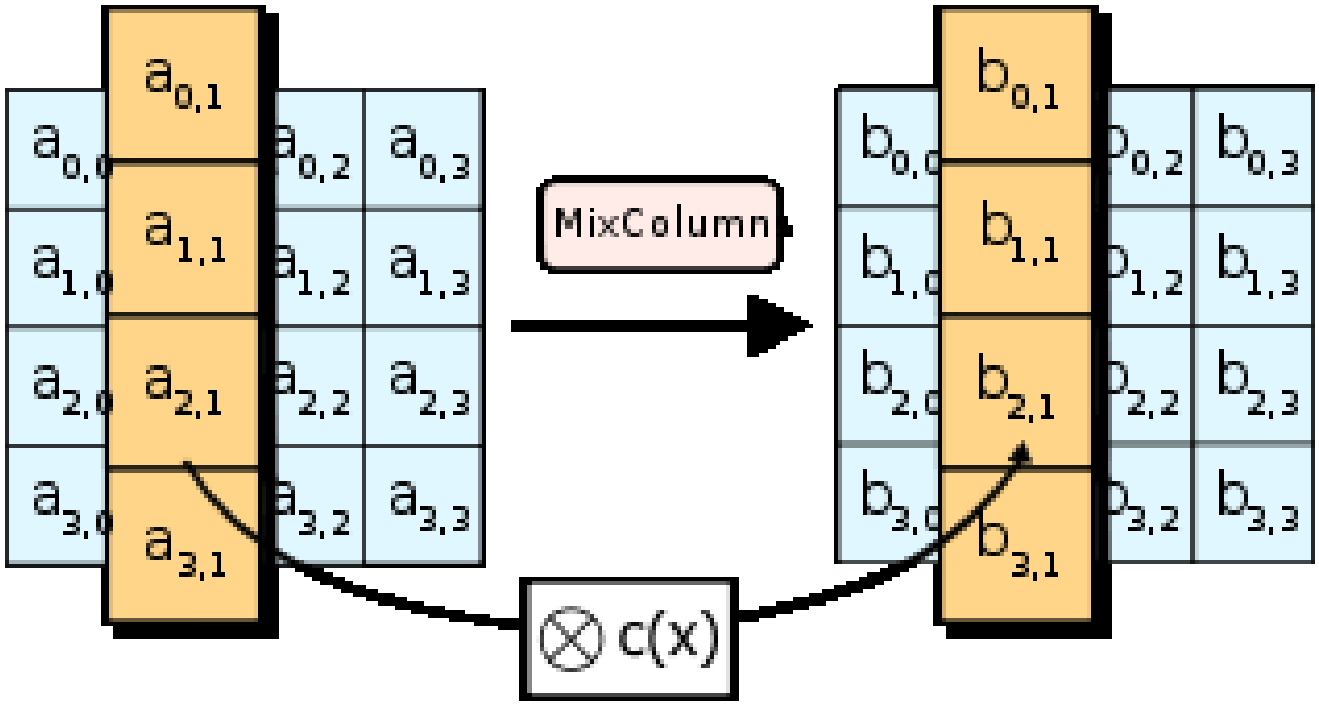
❖ Bước ShiftRows:

- Các dòng của ma trận state được dịch theo chu kỳ sang trái;
- Dòng thứ nhất giữ nguyên.



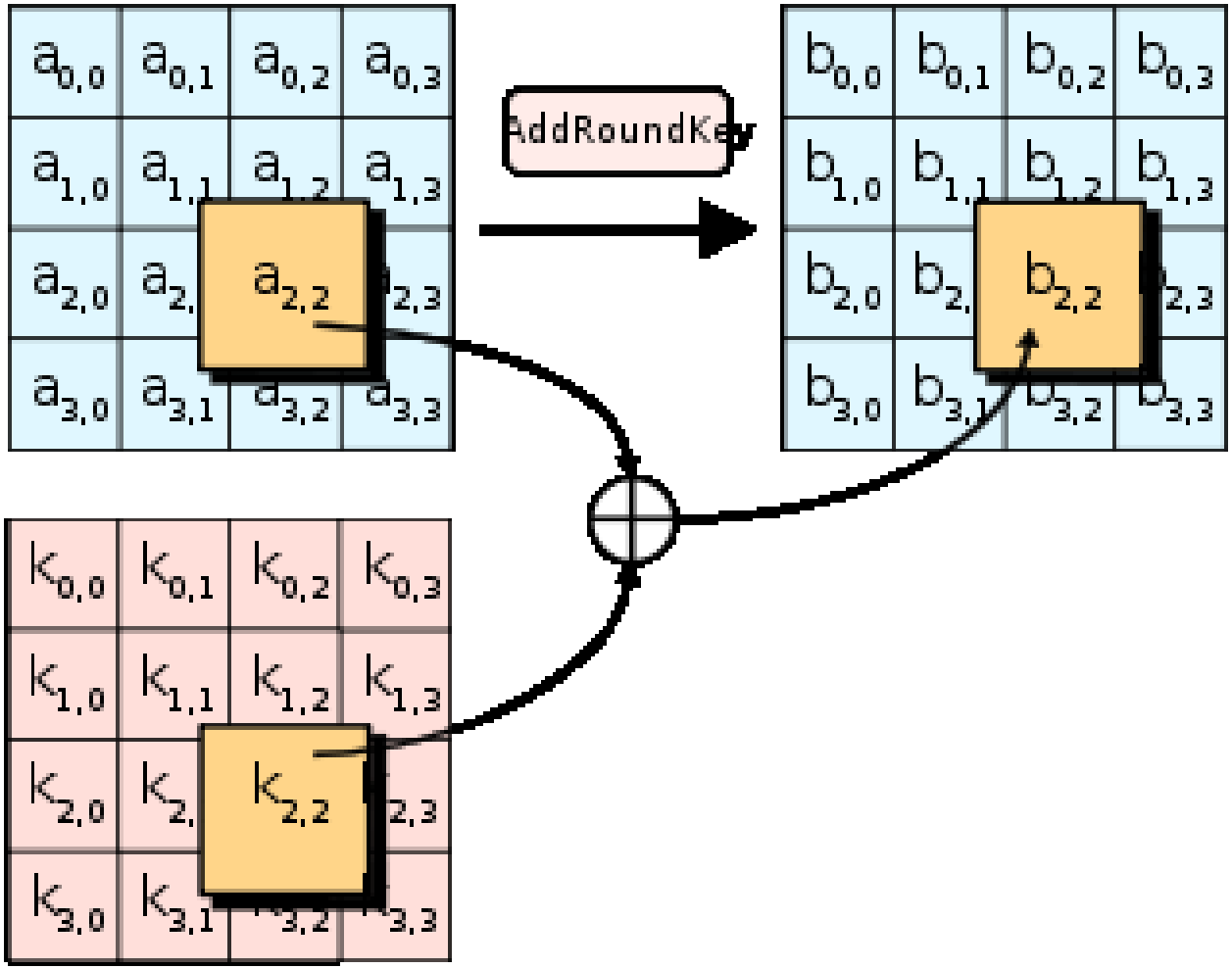
4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

- Bước MixColumns:
  - Mỗi cột của ma trận state được nhân với một đa thức c(x)



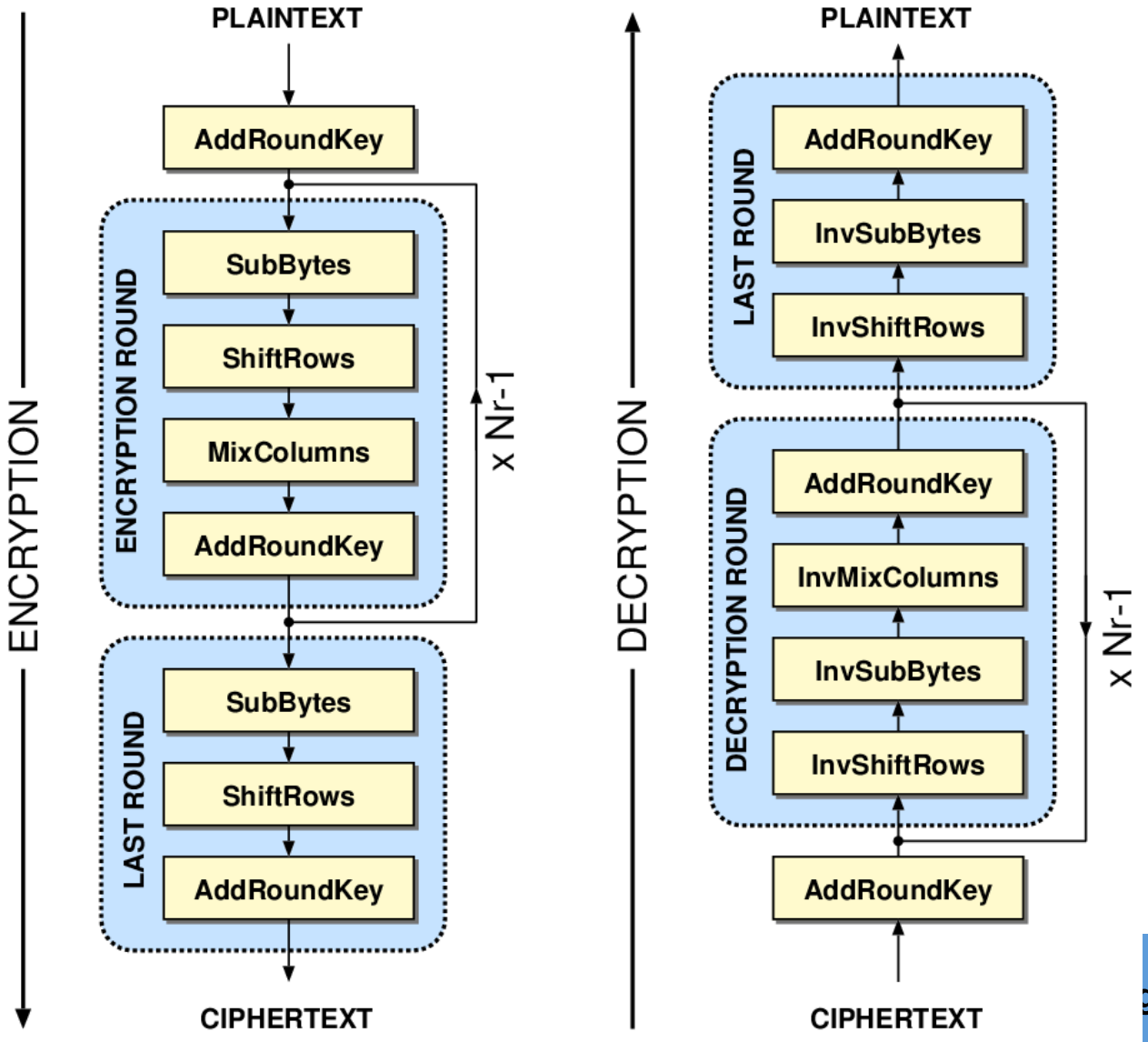
4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

- Bước AddRoundKey:
  - Mỗi byte của ma trận state được kết hợp với một byte của khóa phụ sử dụng phép ⊕ (XOR).



4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

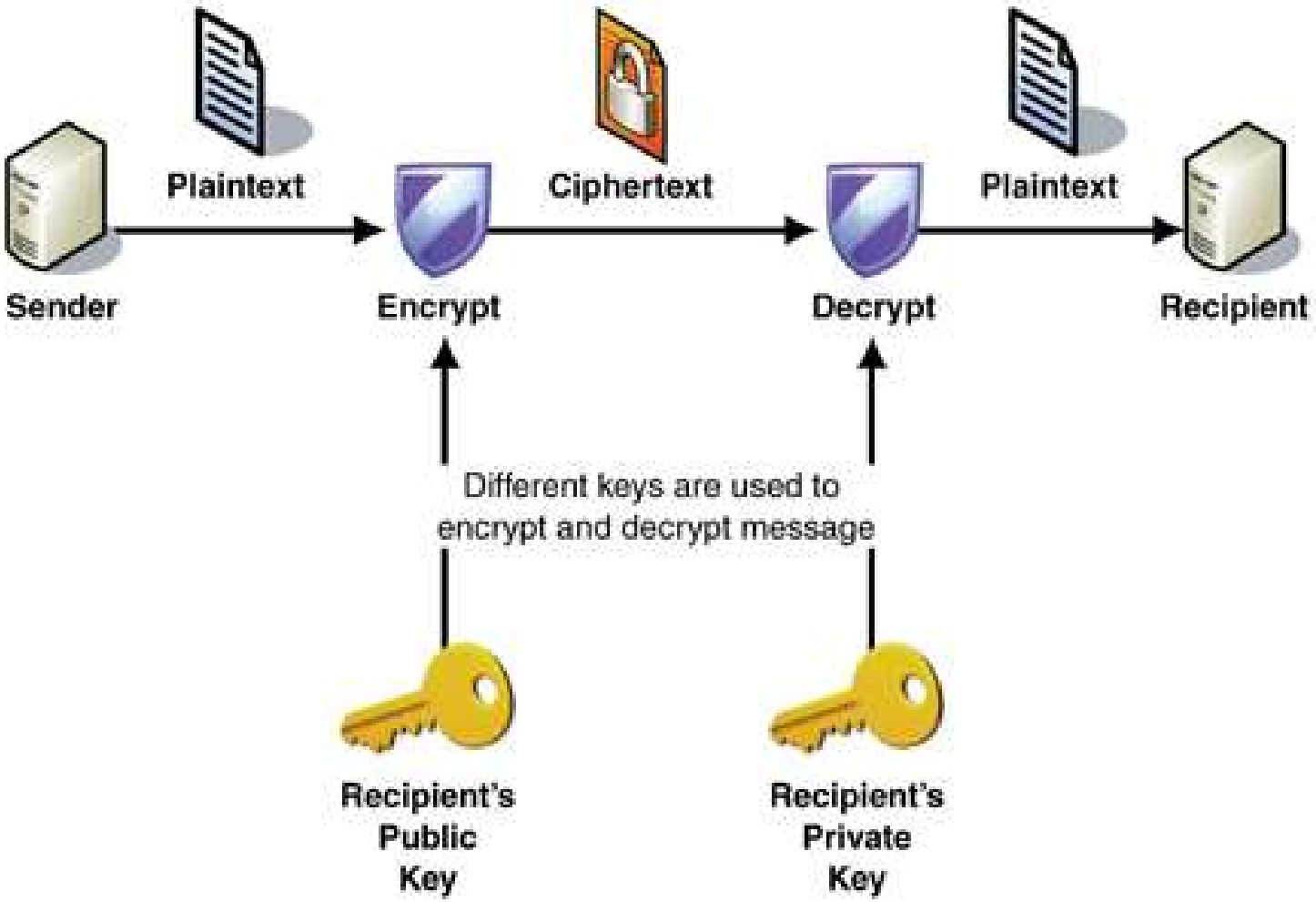
Quá trình mã hóa và giải mã của AES



4.3.1 Các giải thuật mã hóa khóa bất đối xứng

- Các giải thuật mã hóa khóa bất đối xứng (asymetric key encryption)
  - Còn gọi là mã hóa khóa công khai (public key encryption):
  - Sử dụng một cặp khóa (key pair):
    - một khóa (public key) cho mã hóa và
    - một khóa (private key) cho giải mã.

4.3.2 Các giải thuật mã hóa khóa bất đối xứng



4.3.1 Các giải thuật mã hóa khóa bất đối xứng

- Đặc điểm:
  - Kích thước khóa lớn (1024 – 3072 bít)
  - Tốc độ chậm
    - Phần lớn do khóa có kích thước lớn.
  - Độ an toàn cao
  - Thuận lợi trong quản lý và phân phối khóa:
    - Do khóa mã hóa là công khai và có thể trao đổi dễ dàng.

4.3.2 Các giải thuật mã hóa khóa bất đối xứng

- Các giải thuật mã hóa khóa bất đối xứng điển hình:
  - RSA
  - Rabin
  - ElGamal
  - McEliece
  - Knapsack

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Giải thuật mã hóa RSA được 3 nhà khoa học Ronald Rivest, Adi Shamir và Leonard Adleman phát minh năm 1977;
  - Tên giải thuật RSA lấy theo chữ cái đầu của tên 3 ông.
- Độ an toàn của RSA dựa trên tính khó của việc phân tích số nguyên rất lớn:
  - Khóa RSA là số nguyên rất lớn có hàng trăm chữ số thập phân.

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- RSA sử dụng một cặp khóa:
  - Khóa công khai (Public key) dùng để mã hóa;
  - Khóa riêng (Private key) dùng để giải mã.
  - Chỉ khóa riêng cần giữ bí mật. Khóa công khai có thể công bố rộng rãi.
- Kích thước khóa của RSA:
  - Khóa < 1024 bít không an toàn hiện nay.
  - Khuyến nghị dùng khóa >= 2048 bít với các ứng dụng mật mã dân sự hiện nay.
  - Tương lai nên dùng khóa >=3072 bít.

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Thủ tục sinh khóa RSA:
  - Tạo 2 số nguyên tố p và q;
  - Tính n = p x q
  - Tính $\Phi(n) = (p-1) \times (q-1)$
  - Chọn số nguyên tố e sao cho $0 < e < \Phi(n)$ và gcd(e, $\Phi(n)$) = 1, hay e, $\Phi(n)$ là 2 số nguyên tố cùng nhau
  - Chọn số d sao cho $d \equiv e^{-1} \bmod \Phi(n)$,

    hoặc $(d \times e) \bmod \Phi(n) = 1$

    (d là molulo nghịch đảo của e)
- Ta có (n, e) là khóa công khai, (n, d) là khóa riêng.

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Thủ tục mã hóa RSA:
  - Thông điệp m đã được chuyển thành số, m<n
  - Bản mã $c = m^e \bmod n$
- Thủ tục giải mã RSA:
  - Bản mã c, c<n
  - Bản rõ $m = c^d \bmod n$

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Ví dụ 1:
  - Chọn 2 số nguyên tố p=3 và q=11
  - Tính n = p x q = 3 x 11 = 33
  - Tính $\Phi(n) = (p-1) \times (q-1) = 2 \times 10 = 20$
  - Chọn số e sao cho $0 < e < 20$, và e và $\Phi(n)$ là số nguyên tố cùng nhau ($\Phi(n)$ không chia hết cho e). Chọn e = 7
  - Tính $(d \times e) \bmod \Phi(n) \rightarrow (d \times 7) \bmod 20 = 1$

    $d = (20*k + 1)/7 \rightarrow d = 3 \quad (k=1)$
  - Khóa công khai (33, 7)
  - Khóa bí mật (33, 3)

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Ví dụ 1:
  - Mã hóa:
    - Với m = 6,
    - $c = m^e \bmod n = 6^7 \bmod 33 = 279936 \bmod 33 = 30$
    - → c = 30
  - Giải mã:
    - $m = c^d \bmod n = 30^3 \bmod 33 = 27000 \bmod 33 = 6$
    - → m = 6

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Ví dụ 2:
  - Chọn 2 số nguyên tố p=61 và q=53
  - Tính n = p x q = 61 x 53 = 3233
  - Tính $\Phi(n) = (p-1) \times (q-1) = 60 \times 52 = 3120$
  - Chọn số e sao cho $0 < e < 3120$ và e và $\Phi(n)$ là số nguyên tố cùng nhau ($\Phi(n)$ không chia hết cho e). Chọn e = 17
  - Tính $(d \times e) \bmod \Phi(n) \rightarrow (d \times 17) \bmod 3120 = 1$

    $d = (3120*k + 1)/17 \rightarrow d = 2753$ (k=15)
  - Khóa công khai (3233, 17)
  - Khóa bí mật (3233, 2753)

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Ví dụ 2:
  - Mã hóa:
    - Với m = 65,
    - $c = m^e \bmod n = 65^{17} \bmod 3233 = 2790$
    - → c = 2790
  - Giải mã:
    - $m = c^d \bmod n = 2790^{2753} \bmod 3233$
    - → m = 65

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Một số yêu cầu với quá trình sinh khóa RSA:
  - Các số nguyên tố p và q phải được chọn sao cho việc phân tích n (n = pq) là không khả thi về mặt tính toán;
  - p và q nên có cùng độ lớn (tính bằng bit) và phải là các số đủ lớn;
    - Nếu n có kích thước 1024 bít thì p và q nên có kích thước khoảng 512 bít.
    - Nếu n có kích thước 2048 bít thì p và q nên có kích thước khoảng 1024 bít.

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Một số yêu cầu với quá trình sinh khóa RSA:
  - Hiệu số p – q không nên quá nhỏ, do nếu p – q quá nhỏ, tức p ≈ q và p ≈ $\sqrt{n}$ → chọn các số nguyên tố ở gần $\sqrt{n}$ và thử nhiều lần.
  - Khi có được p → tính q, và tìm ra d là khóa bí mật từ khóa công khai e và Φ(n).
  - Nếu p và q được chọn ngẫu nhiên và p – q đủ lớn, khả năng hai số này bị phân tích từ n giảm đi.

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Sử dụng số mũ mã hóa (e) nhỏ:
  - Khi sử dụng số mũ mã hóa (e) nhỏ, chẳng hạn e=3 có thể tăng tốc độ mã hóa;
  - Kẻ tấn công có thể nghe trộm và lấy được bản mã, từ đó phân tích bản mã để khôi phục bản rõ. Do số mũ nhỏ nên chi phí cho phân tích/vét cạn không quá lớn;
  - Phòng chống:
    - Sử dụng số mũ e lớn;
    - Thêm chuỗi ngẫu nhiên vào khối rõ trước khi mã hóa.

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Sử dụng số mũ giải mã (d) nhỏ:
  - Khi sử dụng số mũ giải mã (d) nhỏ, có thể tăng tốc độ giải mã;
  - Nếu d nhỏ và gcd(p-1, q-1) (gcd: ước số chung lớn nhất) cũng nhỏ thì d có thể tính được tương đối dễ dàng từ khóa công khai (n, e);
  - Phòng chống:
    - Sử dụng số mũ d đủ lớn.

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Cài đặt RSA trên thực tế:
  - Do kích thước cặp khóa của RSA rất lớn (n cỡ 2048 bít – khoảng hơn 600 chữ số thập phân), việc thực hiện RSA trực tiếp có chi phí tính toán và lưu trữ rất lớn:
    - Mã hóa $c = m^e \bmod n$
    - Giải mã $m = c^d \bmod n$
    - Do m, e và d thường rất lớn nên giá trị mũ m^e hoặc c^d thường rất rất lớn.
  - → cần có giải thuật hiệu quả để giảm chi phí tính toán → cài đặt trên máy tính.

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Cài đặt trong java:
  - Ngôn ngữ lập trình java định nghĩa lớp BigInteger cung cấp hầu hết các hàm dựng và các hàm số học cho phép thao tác thuận lợi với số nguyên lớn.
  - Một số hàm có thể dùng để cài đặt RSA:
    - Hàm dựng BigInteger(int bitLength, int certainty, Random rnd): sinh số nguyên tố ngẫu nhiên với số bit cho trước;
    - Hàm BigInteger add(BigInteger val): cộng hai số nguyên lớn;
    - Hàm BigInteger gcd(BigInteger val): tìm ƯSC lớn nhất của 2 số nguyên lớn;
    - Hàm BigInteger mod(BigInteger m): tính modulo (phần dư) của phép chia nguyên;
    - Hàm BigInteger modInverse(BigInteger m): tính modulo nghịch đảo ($this^{-1}$ mod m);
    - BigInteger modPow(BigInteger exponent, BigInteger m): tính ($this^{exponent}$ mod m).

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

❖ Cài đặt trên ngôn ngữ C:

- Do thư viện ngôn ngữ C không hỗ trợ số lớn nên việc cài đặt RSA trong C phải thực hiện từ các thao tác số học cơ sở;
- Có thể sử dụng 1 mảng để lưu các chữ số của số nguyên lớn và xây dựng các hàm thực hiện các phép toán số học và modulo cho số nguyên lớn;

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

- Cài đặt trên ngôn ngữ C:
  - Lựa chọn cơ số:
    - Cơ số 10: đơn giản, dễ hiểu. Tuy nhiên, tốn không gian lưu trữ và chậm do không tận dụng được khả năng thực hiện các phép toán nhân/chia với số 2 thông qua phép dịch. → Cơ số nên là số mũ của 2 và cần đủ lớn;
    - Cơ số 256: một số được lưu trong 1 phần tử mảng là 1 byte → tiết kiệm không gian lưu trữ. Tuy nhiên, số phần tử mảng vẫn có thể khá lớn → chậm trong thao tác;
    - Cơ số 2^{16} (65536): khá phù hợp do một số được lưu trong 1 phần tử mảng là 2 byte và số phần tử mảng sẽ giảm → nhanh hơn trong thao tác.

4.3.2 Các giải thuật mã hóa khóa bất đối xứng - RSA

❖ Cài đặt trên ngôn ngữ C: định nghĩa cấu trúc BigInt

```
typedef struct {
    unsigned short *digits;    // pointer to array of digits
                               // the least significant digit at index 0
    unsigned int size;         // number of digits of the big integer
    short sign;                // sign of the big integer,
                               // sign = -1 for negative number, and 1 otherwise
} BigInt ;
```

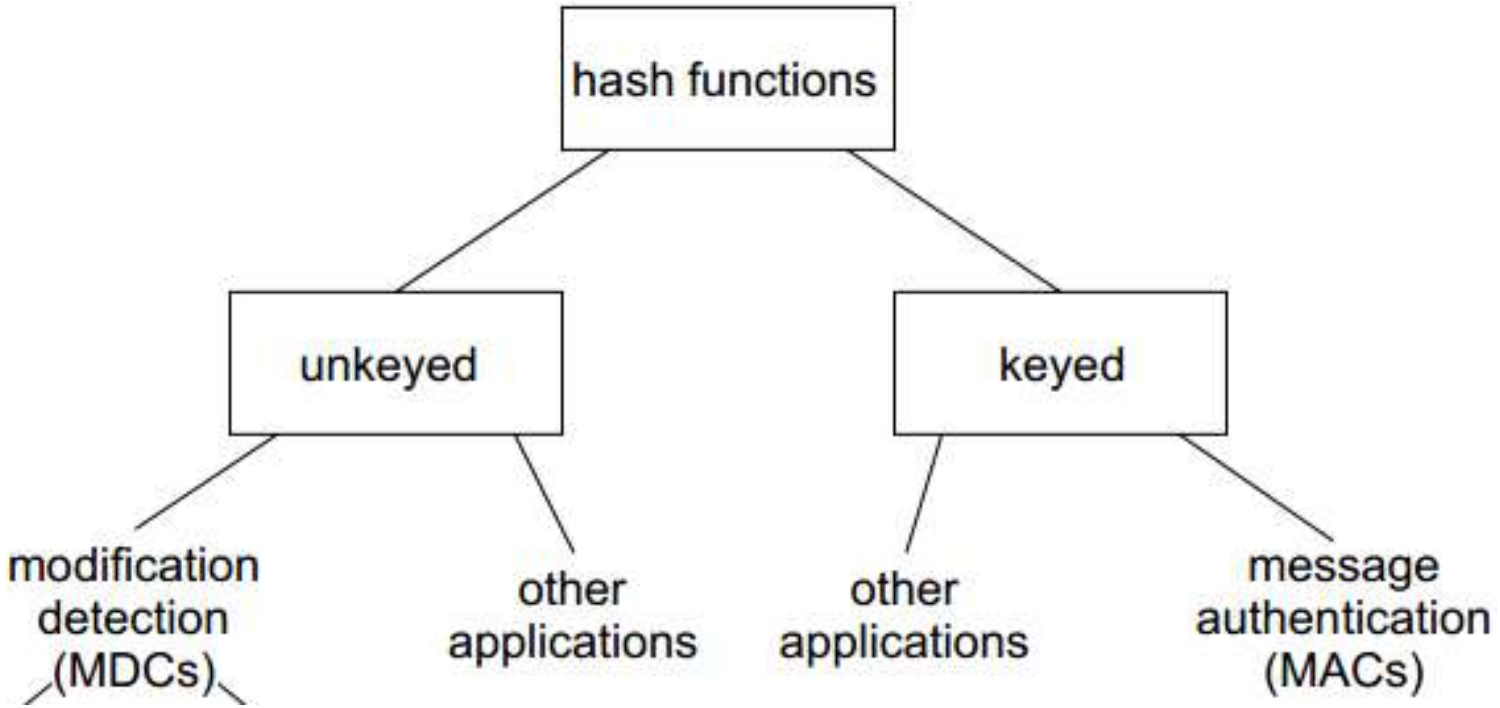
4.3.3 Các hàm băm

- Hàm băm (hash function) là một hàm toán học h có tối thiểu 2 thuộc tính cơ bản:
  - Nén (compression): h là một ánh xạ từ chuỗi đầu vào x có chiều dài bất kỳ sang một chuỗi đầu ra h(x) có chiều dài cố định n bít;
  - Dễ tính toán (ease of computation): cho trước hàm h và đầu vào x, việc tính toán h(x) là dễ dàng.

4.3.3 Các hàm băm

❖ Phân loại hàm băm theo khóa sử dụng:

- Hàm băm không khóa (unkeyed): đầu vào chỉ là thông điệp;
- Hàm băm có khóa (keyed): đầu vào gồm thông điệp và khóa.



4.3.3 Các hàm băm

- Phân loại hàm băm theo tính năng:
  - Mã phát hiện sửa đổi (MDC - Modification detection codes)
  - Mã xác thực thông điệp (MAC - Message authentication codes).

4.3.3 Các hàm băm

- Phân loại hàm băm theo tính năng:
  - Mã phát hiện sửa đổi (MDC - Modification detection codes)
    - MDC thường được sử dụng để tạo chuỗi đại diện cho thông điệp và dùng kết hợp với các biện pháp khác để đảm bảo tính toàn vẹn của thông điệp;
    - MDC thuộc loại hàm băm không khóa;
    - MDC thường được sử dụng trong các quá trình tạo và kiểm tra chữ ký số để đảm bảo tính toàn vẹn thông điệp.

4.3.3 Các hàm băm

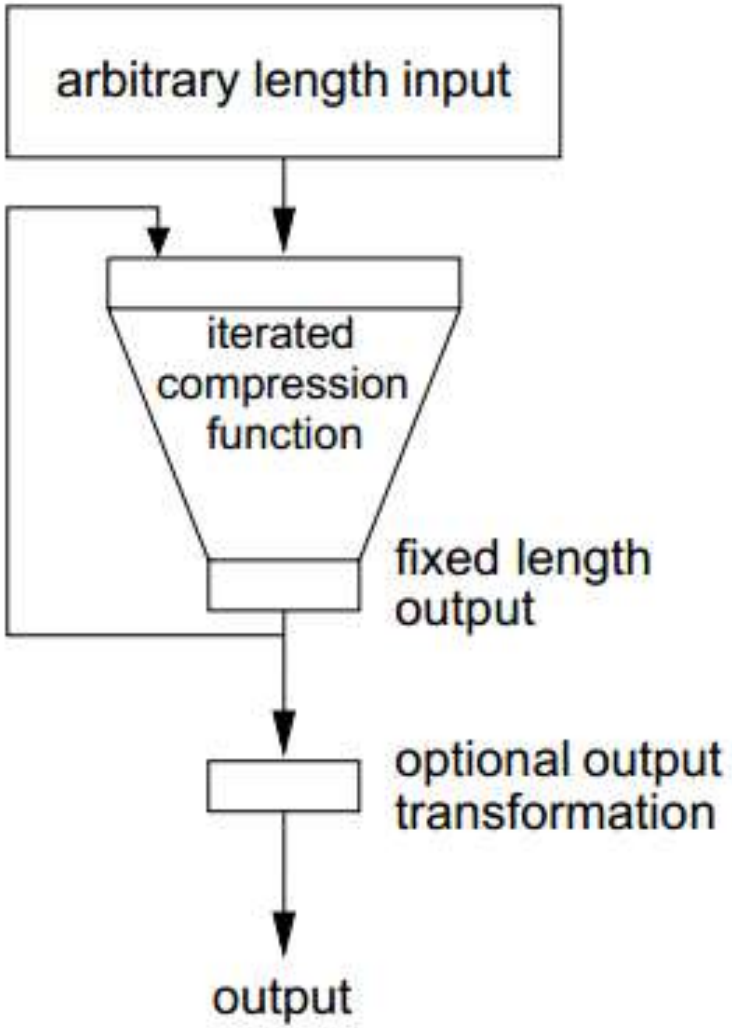
- Phân loại hàm băm theo tính năng:
  - Mã phát hiện sửa đổi (MDC - Modification detection codes)
    - Hai loại MDC:
      - Hàm băm một chiều (OWHF - One-way hash functions): dễ dàng tính giá trị băm, nhưng khôi phục thông điệp từ giá trị băm rất khó khăn;
      - Hàm băm chống đụng độ (CRHF - Collision resistant hash functions): Rất khó tìm được 2 thông điệp trùng giá trị băm.

4.3.3 Các hàm băm

- Phân loại hàm băm theo tính năng:
  - Mã xác thực thông điệp (MAC - Message authentication codes)
    - MAC cũng được dùng để đảm bảo tính toàn vẹn của thông điệp mà không cần một biện pháp bổ sung khác;
    - MAC là loại hàm băm có khóa: đầu vào là thông điệp và một khóa;
    - MAC được sử dụng trong các giao thức bảo mật SSL/TLS, IPSec,… để đảm bảo tính toàn vẹn thông điệp.

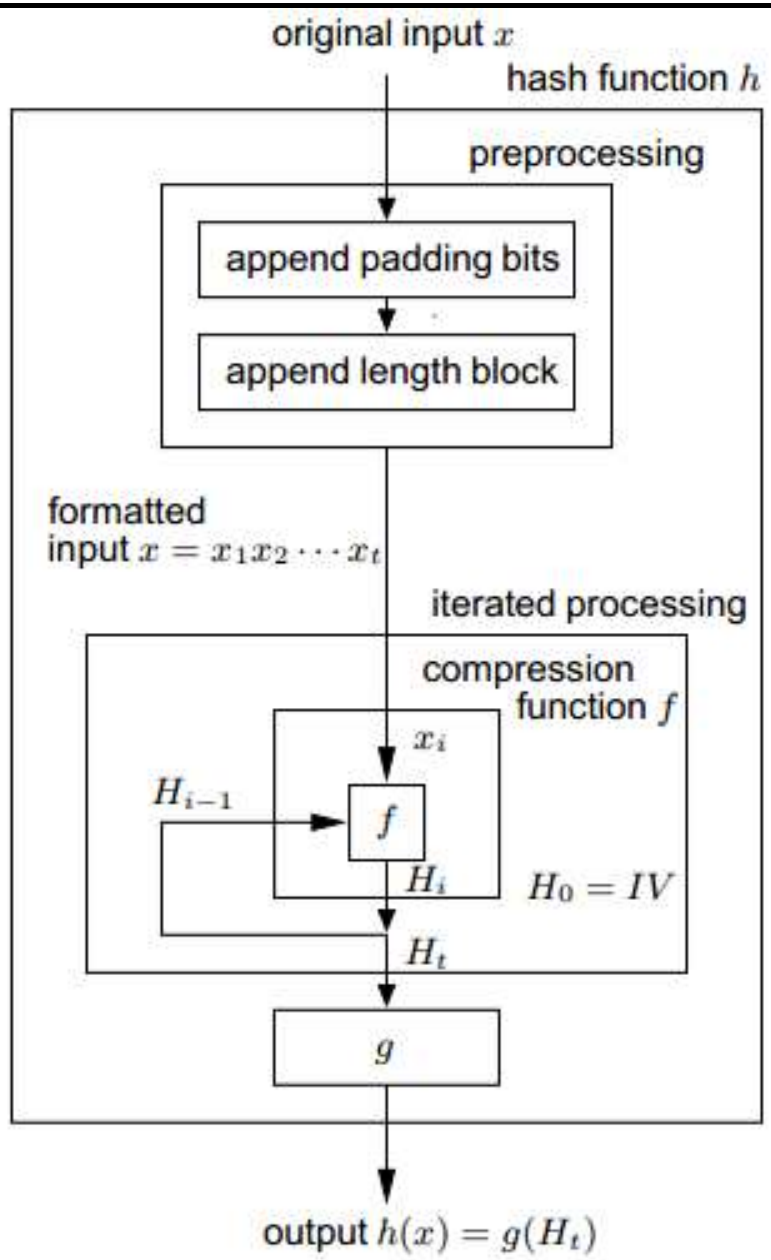
4.3.3 Các hàm băm

Mô hình lặp tổng quát tạo giá trị băm



4.3.3 Các hàm băm

Mô hình lặp chi tiết tạo giá trị băm



4.3.3 Các hàm băm

- Một số giải thuật hàm băm điển hình:
  - CRC (Cyclic redundancy checks)
  - Checksums
  - MD2, MD4, MD5
  - MD6
  - SHA0, SHA1
  - SHA2, SHA3

4.3.3 Các hàm băm – MD5

- MD5 (Message Digest) là hàm băm không khóa được Ronald Rivest thiết kế năm 1991 để thay thế MD4;
- Chuỗi đầu ra (giá trị băm) của MD5 là 128 bít (16 bytes) và thường được biểu diễn thành 32 số hexa;
- MD5 được sử dụng khá rộng rãi trong nhiều ứng dụng:
  - Chuỗi đảm bảo tính toàn vẹn thông điệp;
  - Tạo chuỗi kiểm tra lỗi – Checksum;
  - Mã hóa mật khẩu.

4.3.3 Các hàm băm – MD5

- Quá trình xử lý thông điệp của MD5:
  - Thông điệp được chia thành các khối 512 bít. Nếu kích thước thông điệp không là bội số của 512 → nối thêm số bít thiếu;
  - Phần xử lý chính của MD5 làm việc trên state 128 bít, chia thành 4 từ 32 bít (A, B, C, D);
    - Các từ A, B, C, D được khởi trị bằng một hằng cố định;
    - Từng phần 32 bít của khối đầu vào 512 bít được đưa dần vào để thay đổi state;
  - Quá trình xử lý gồm 4 vòng, mỗi vòng gồm 16 thao tác tương tự nhau;
  - Mỗi thao tác gồm:
    - Hàm F (4 hàm khác nhau cho mỗi vòng);
    - Cộng modulo;
    - Quay trái.

4.3.3 Các hàm băm – MD5

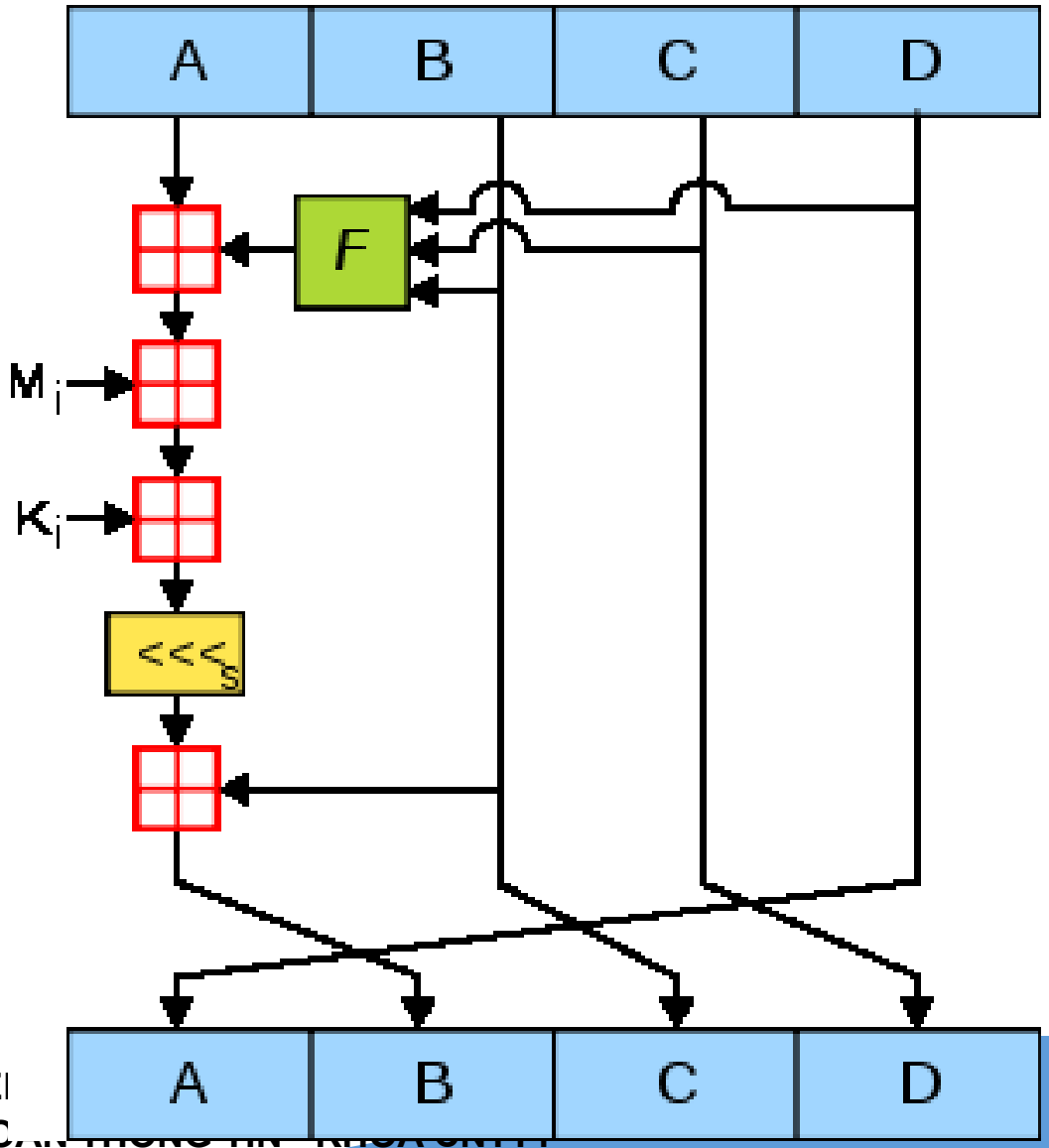
- Lưu đồ xử lý một thao tác của MD5:
  - A, B, C, D: các từ 32 bit
  - Mi: khối 32 bit thông điệp đầu vào;
  - Ki: 32 bit hằng. Mỗi thao tác sử dụng một hằng khác nhau;
  - <<<s: thao tác dịch trái s bit
  - ⊞ biểu diễn cộng modulo 32 bít;
  - F: hàm phi tuyến tính, gồm 4 loại:

$$F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$$

$$G(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$$

$$H(B,C,D) = B \oplus C \oplus D$$

$$I(B,C,D) = C \oplus (B \vee \neg D)$$



4.3.3 Các hàm băm – SHA1

- SHA1 (Secure Hash Function) được NSA (Mỹ) thiết kế năm 1995 để thay thế cho SHA0;
- Chuỗi đầu ra của SHA1 có kích thước 160 bít và thường được biểu diễn thành 40 số hexa;
- SHA1 được sử dụng rộng rãi để:
  - Đảm bảo tính xác thực và toàn vẹn thông điệp;
  - Mã hóa mật khẩu.

4.3.3 Các hàm băm – SHA1

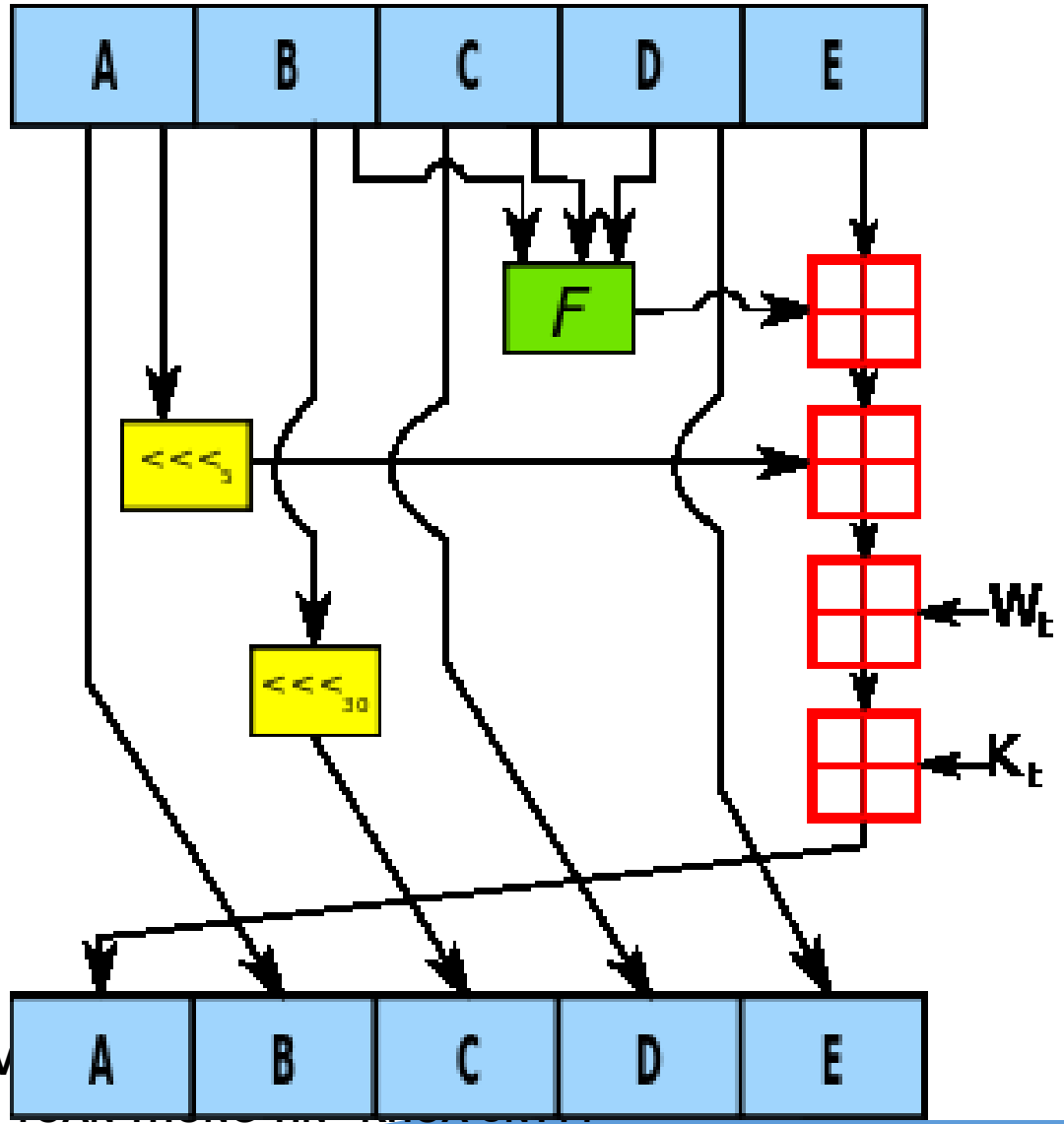
- Họ hàm băm SHA: SHA-0, SHA-1, SHA-2, SHA-3:
  - SHA0 ít được sử dụng trên thực tế;
  - SHA1 tương tự SHA0, nhưng đã khắc phục một số lỗi;
  - SHA2 ra đời năm 2001 khắc phục lỗi của SHA1 và có nhiều thay đổi. Kích thước chuỗi đầu ra có thể là 224, 256, 384 và 512 bít;
  - SHA3 ra đời năm 2012, cho phép chuỗi đầu ra có kích thước không cố định.

4.3.3 Các hàm băm – SHA1

- Quá trình xử lý thông điệp của SHA1:
  - SHA1 sử dụng thủ tục xử lý thông điệp tương tự MD5;
  - Thông điệp được chia thành các khối 512 bít. Nếu kích thước thông điệp không là bội số của 512 → nối thêm số bít thiếu;
  - Phần xử lý chính của SHA1 làm việc trên state 160 bít, chia thành 5 từ 32 bít (A, B, C, D, E);
    - Các từ A, B, C, D, E được khởi trị bằng một hằng cố định;
    - Từng phần 32 bít của khối đầu vào 512 bít được đưa dần vào để thay đổi state;
  - Quá trình xử lý gồm 80 vòng, mỗi vòng gồm các thao tác: add, and, or, xor, rotate, mod.

4.3.3 Các hàm băm – SHA1

- Lưu đồ xử lý một vòng của SHA1:
  - A, B, C, D, E: các từ 32 bit
  - Wt: khối 32 bit thông điệp đầu vào;
  - Kt: 32 bit hằng. Mỗi sử dụng một hằng khác nhau;
  - <<<n: thao tác dịch trái n bit
  - ⊞ biểu diễn cộng modulo 32 bít;
  - F: hàm phi tuyến tính.



4.4 Chữ ký số, chứng chỉ số và PKI

1. Chữ ký số
   - Khái niệm
   - Quá trình ký và kiểm tra chữ ký số
   - Thuật toán chữ ký số RSA
   - Thuật toán chữ ký số DSA
2. Chứng chỉ số
3. Hạ tầng khóa công khai - PKI – Public Key Infrastructure

4.4.1 Chữ ký số

❖ Một số khái niệm:

- 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;

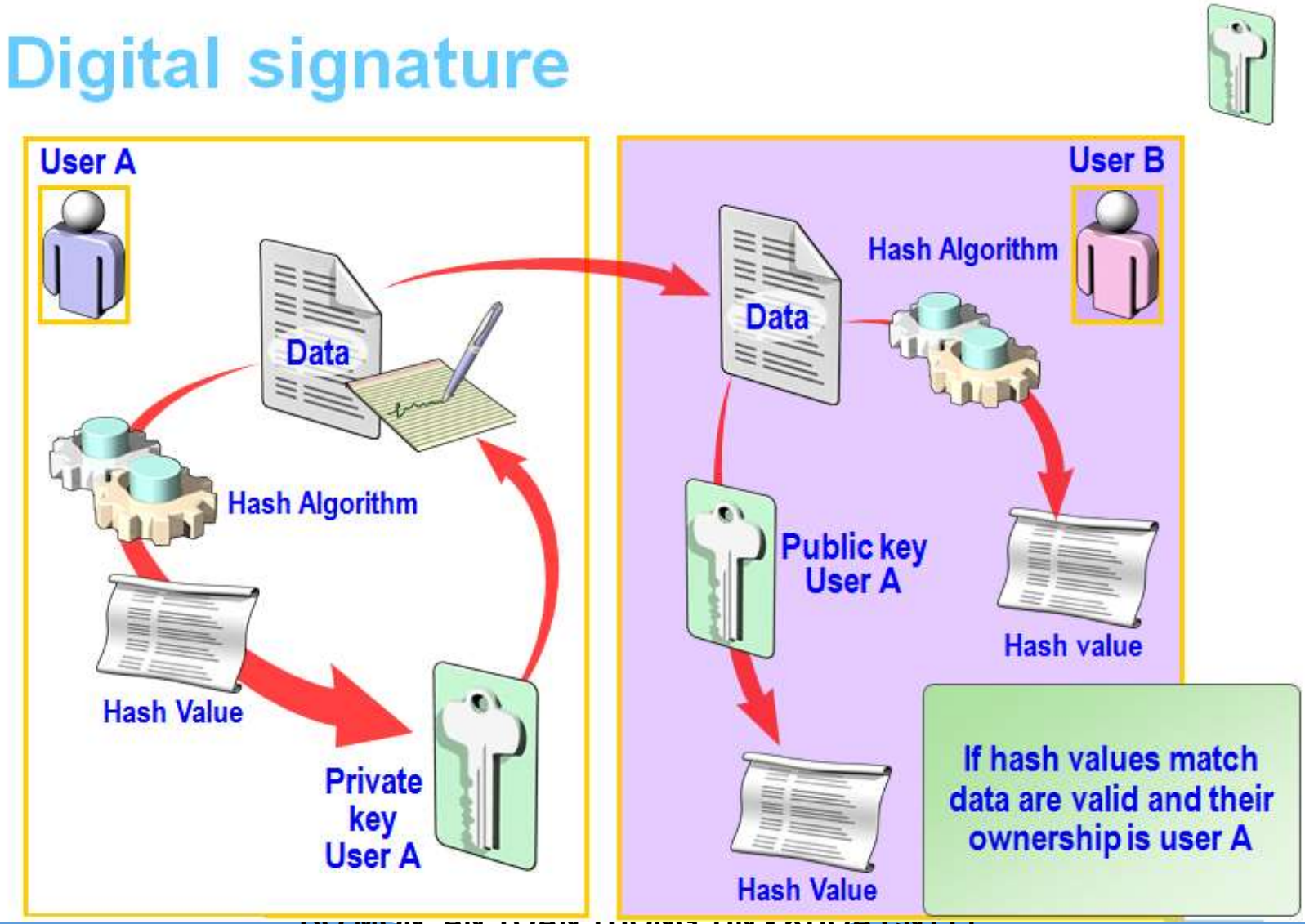
4.4.1 Chữ ký số

- Một số khái niệm:
  - 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, và
    - 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, và
    - Phương pháp khôi phục dữ liệu từ thông điệp.

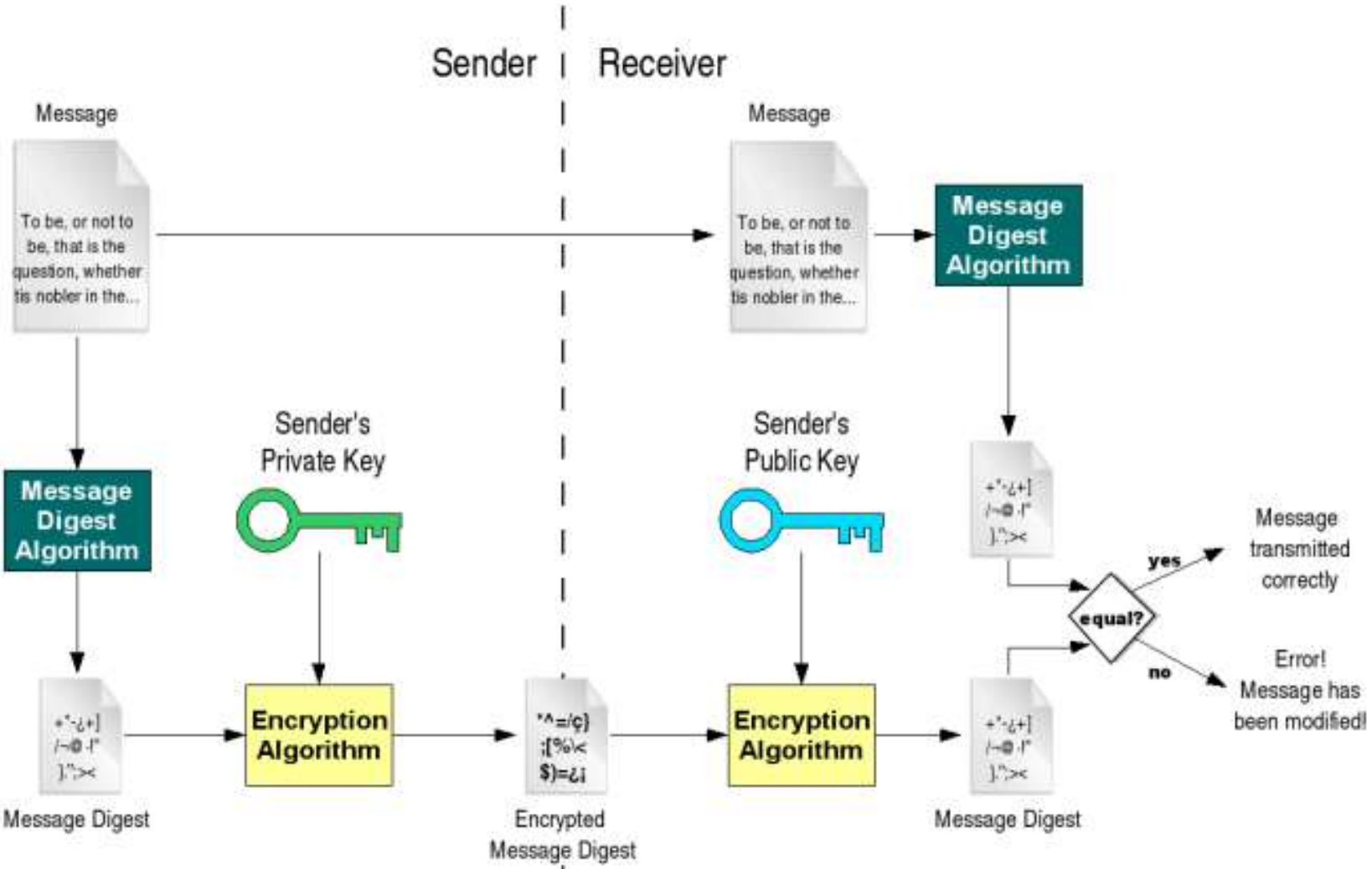
4.4.1 Chữ ký số


NGÂN HÀNG TMCP KỸ THƯƠNG VIỆT NAM
TECHCOMBANK TÂN BÌNH
GIÁM ĐỐC
CHI NHÁNH
NGÂN HÀNG
THƯƠNG MẠI CỔ PHẦN
KỸ THƯƠNG
VIỆT NAM
Ngô Quang Trường

4.4.1 Chữ ký số


Digital signature
User A
Data
Hash Algorithm
Hash Value
Private key User A
User B
Data
Hash Algorithm
Public key User A
Hash value
Hash Value
If hash values match data are valid and their ownership is user A

4.4.1 Chữ ký số - Quá trình ký và kiểm tra



4.4.1 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.

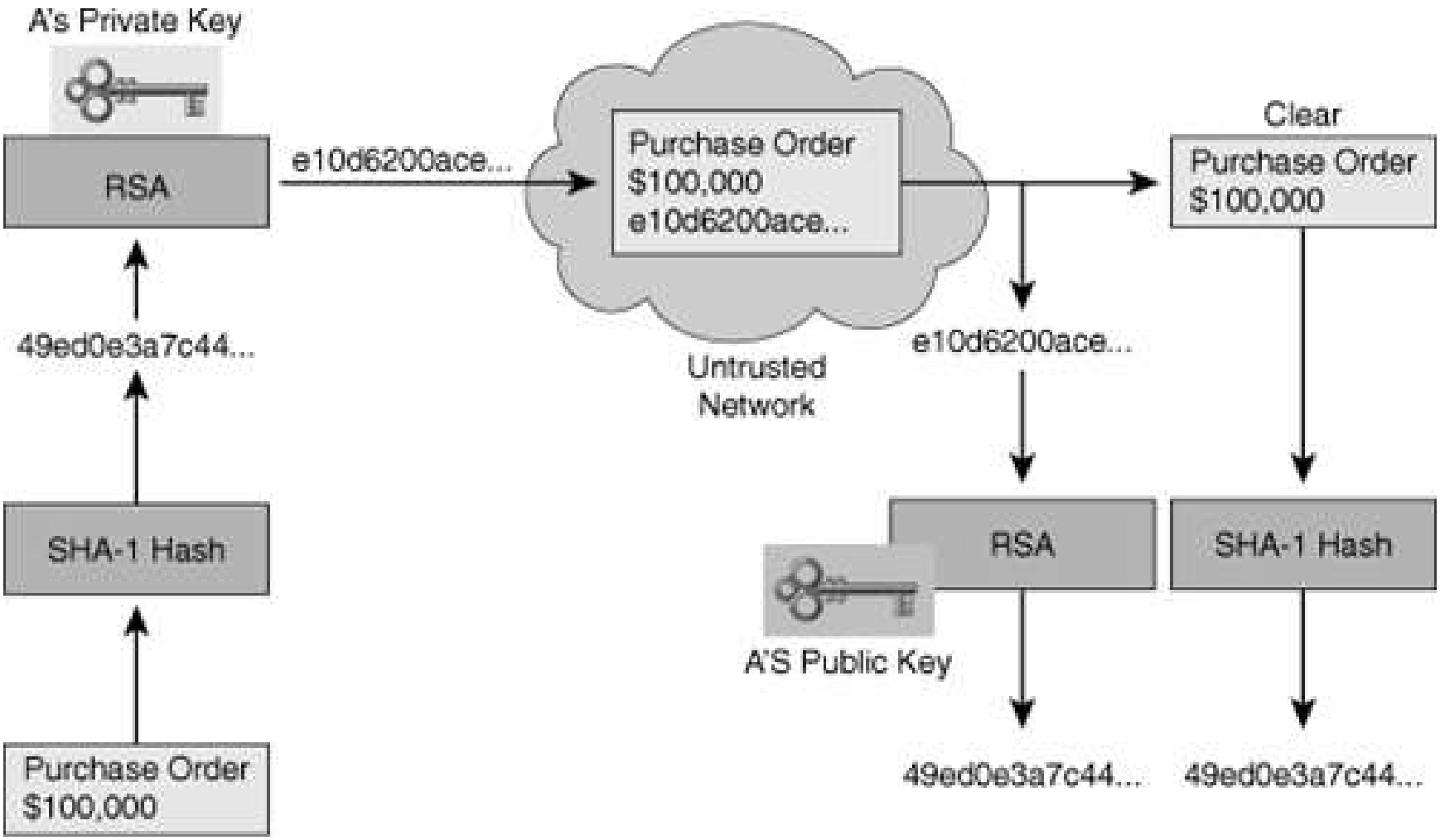
4.4.1 Chữ ký số - Quá trình kiểm tra

- 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 → 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 → chữ ký không hợp lệ. Thông điệp có thể đã bị sửa đổi hoặc không thực sự xuất phát từ người gửi.

4.4.1 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.

4.4.1 Chữ ký số - Giải thuật chữ ký số RSA



4.4.1 Chữ ký số - Giải thuật chữ ký số DSA

- DSA (Digital Signature Algorithm) là chuẩn chữ ký số được phát triển bởi NIST (Mỹ) năm 1991;
- DSA được phát triển từ giải thuật Digital Signature Standard (DSS);
- Các thành phần của DSA:
  - Sinh khóa: sinh cặp khóa. Gồm 2 giai đoạn:
    - Lựa chọn tham số của giải thuật;
    - Sinh cặp khóa cho người dùng.
  - Quá trình ký: ký thông điệp
  - Quá trình kiểm tra chữ ký: kiểm tra chữ ký.

4.4.1 Chữ ký số - Giải thuật chữ ký số DSA

- Sinh khóa:
  - Lựa chọn tham số:
    - Lựa chọn giải thuật băm chuẩn H. Giải thuật băm có thể được lựa chọn là SHA-1 hoặc SHA-2;
    - Chọn kích thước cho các khóa L và N.
      - L có thể là 1024, 2048, 3072;
      - N có thể là 160, 224, 256. N phải nhỏ hơn hoặc bằng kích thước chuỗi băm đầu ra của hàm H đã chọn;
      - Chọn số nguyên tố q N bít;
      - Chọn modulo p L bít sao cho p-1 là bội số của q;
      - Chọn g là hệ số nhân sao cho (g*q) mod p = 1;
      - Các tham số (q, p và g) được chia sẻ giữa các người dùng.

4.4.1 Chữ ký số - Giải thuật chữ ký số DSA

- Sinh khóa:
  - Sinh khóa cho một người dùng:
    - Chọn số ngẫu nhiên x sao cho $0 < x < q$;
    - Tính $y = g^x \bmod p$;
    - Khóa công khai là (q, p, g, y);
    - Khóa riêng là x.

4.4.1 Chữ ký số - Giải thuật chữ ký số DSA

- Ký thông điệp:
  - H là hàm băm sử dụng và m là thông điệp gốc;
  - Tính H(m) từ thông điệp gốc;
  - Tạo số ngẫu nhiên k cho mỗi thông điệp, $0 < k < q$;
  - Tính $r = (g^k \bmod p) \bmod q$;
  - Nếu r = 0, chọn một k mới và tính lại r;
  - Tính $s = k^{-1}(H(m) + xr) \bmod q$;
  - Nếu s = 0, chọn một k mới và tính lại r và s;
  - Chữ ký là cặp (r, s).

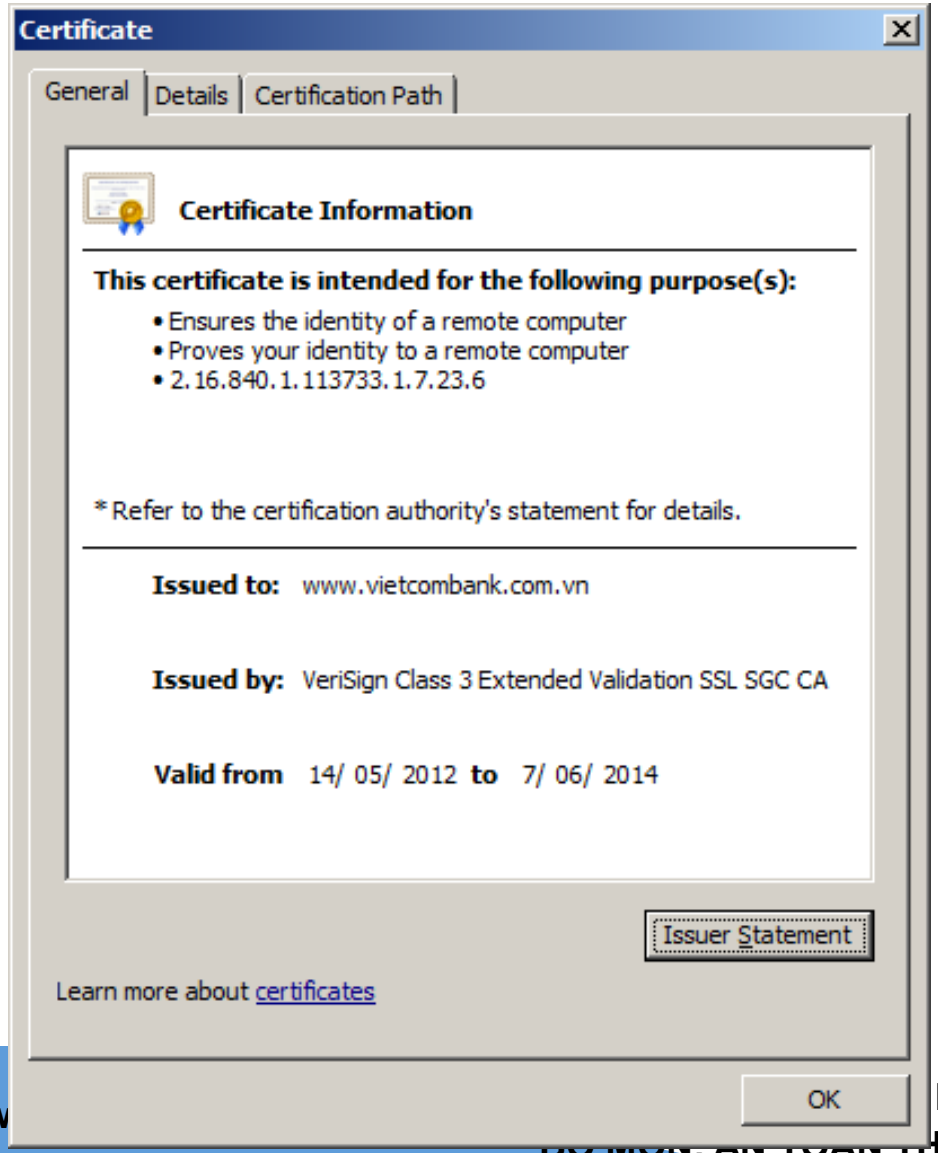
4.4.1 Chữ ký số - Giải thuật chữ ký số DSA

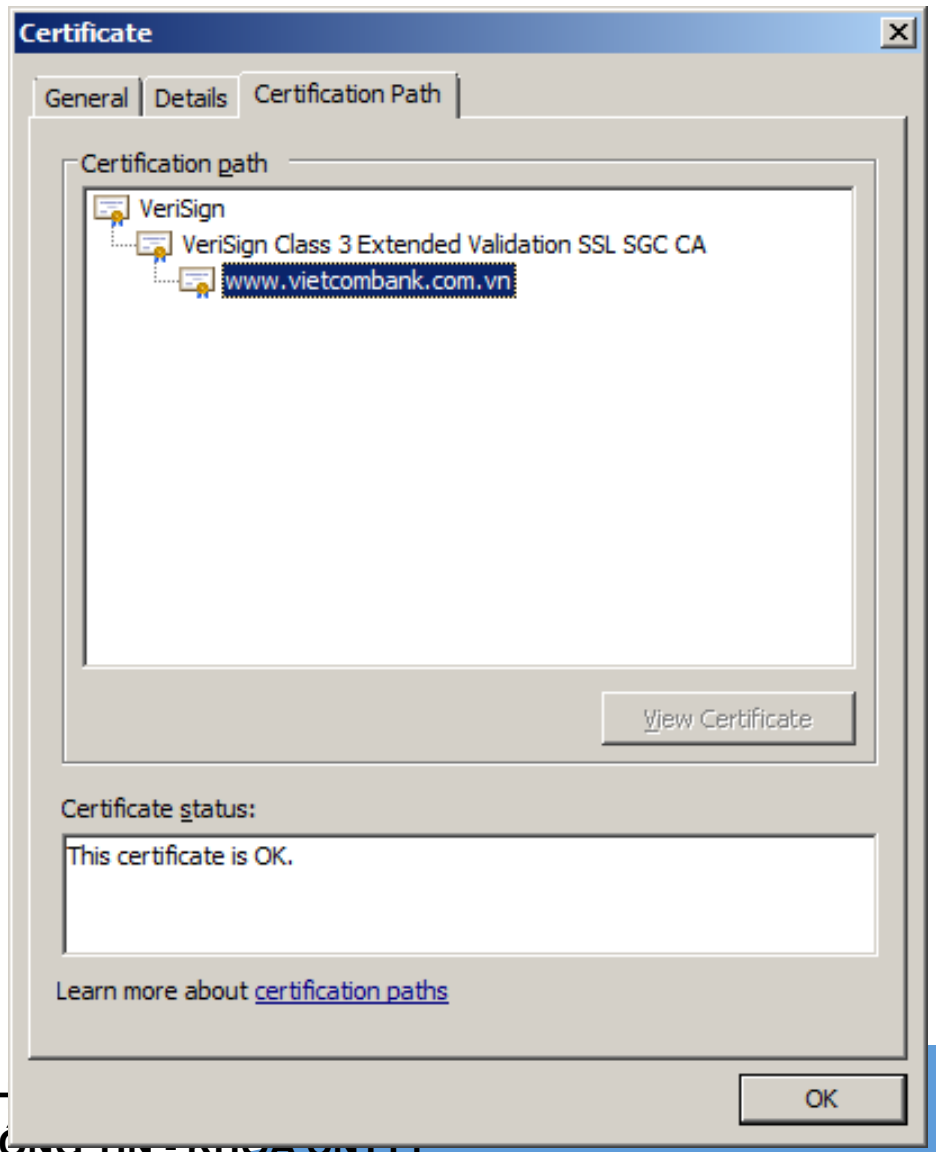
- Kiểm tra chữ ký của thông điệp:
  - Loại bỏ chữ ký nếu r và s không thỏa mãn $0 < r , s < q$;
  - Tính H(m) từ thông điệp nhận được;
  - Tính $w = s^{-1} \bmod q$;
  - Tính $u_1 = H(m) * w \bmod q$;
  - Tính $u_2 = r * w \bmod q$;
  - Tính $v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$;
  - Chữ ký là xác thực nếu v = r.

4.4.2 Chứng chỉ số - Giới thiệu

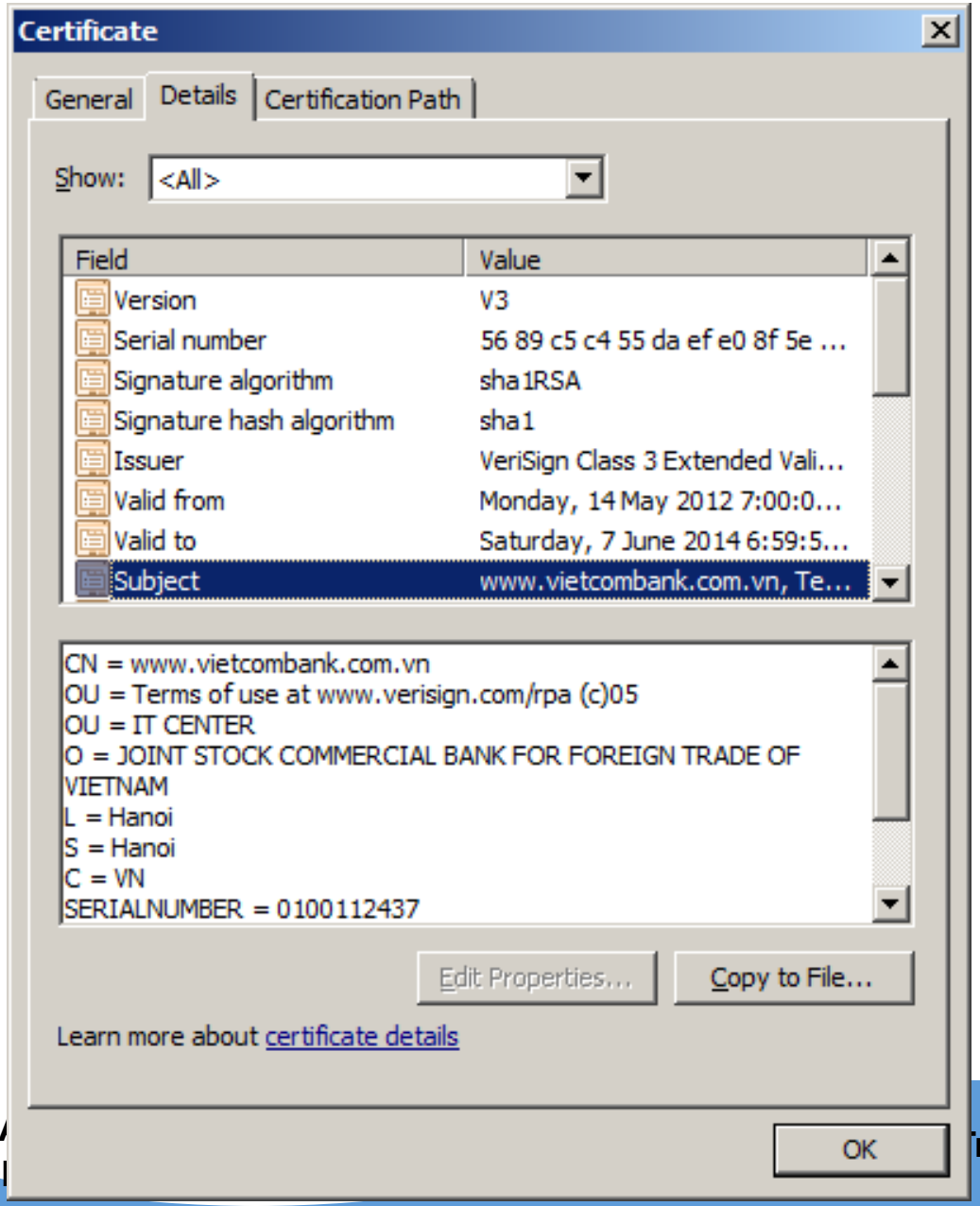
- Chứng chỉ số (Digital certificate), còn gọi là chứng chỉ khóa công khai (Public key certificate), hay chứng chỉ nhận dạng (Identity certificate) là một tài liệu điện tử sử dụng một ***chữ ký số*** để liên kết một ***khóa công khai*** và ***thông tin nhận dạng*** của một thực thể:
  - Chữ ký số: là 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 thể thực sự của một khóa công khai.

4.4.2 Chứng chỉ số - Nội dung





4.4.2 Chứng chỉ số - Nội dung



4.4.2 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;

4.4.2 Chứng chỉ số - Nội dung

- 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 thể;
  - Thumbprint Algorithm: Giải thuật hash sử dụng để tạo chuỗi băm cho khóa công khai;
  - Thumbprint: Chuỗi băm tạo từ khóa công khai;

4.4.2 Chứng chỉ số - Nội dung

- 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.

4.4.2 Chứng chỉ số - Sử dụng

- 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.

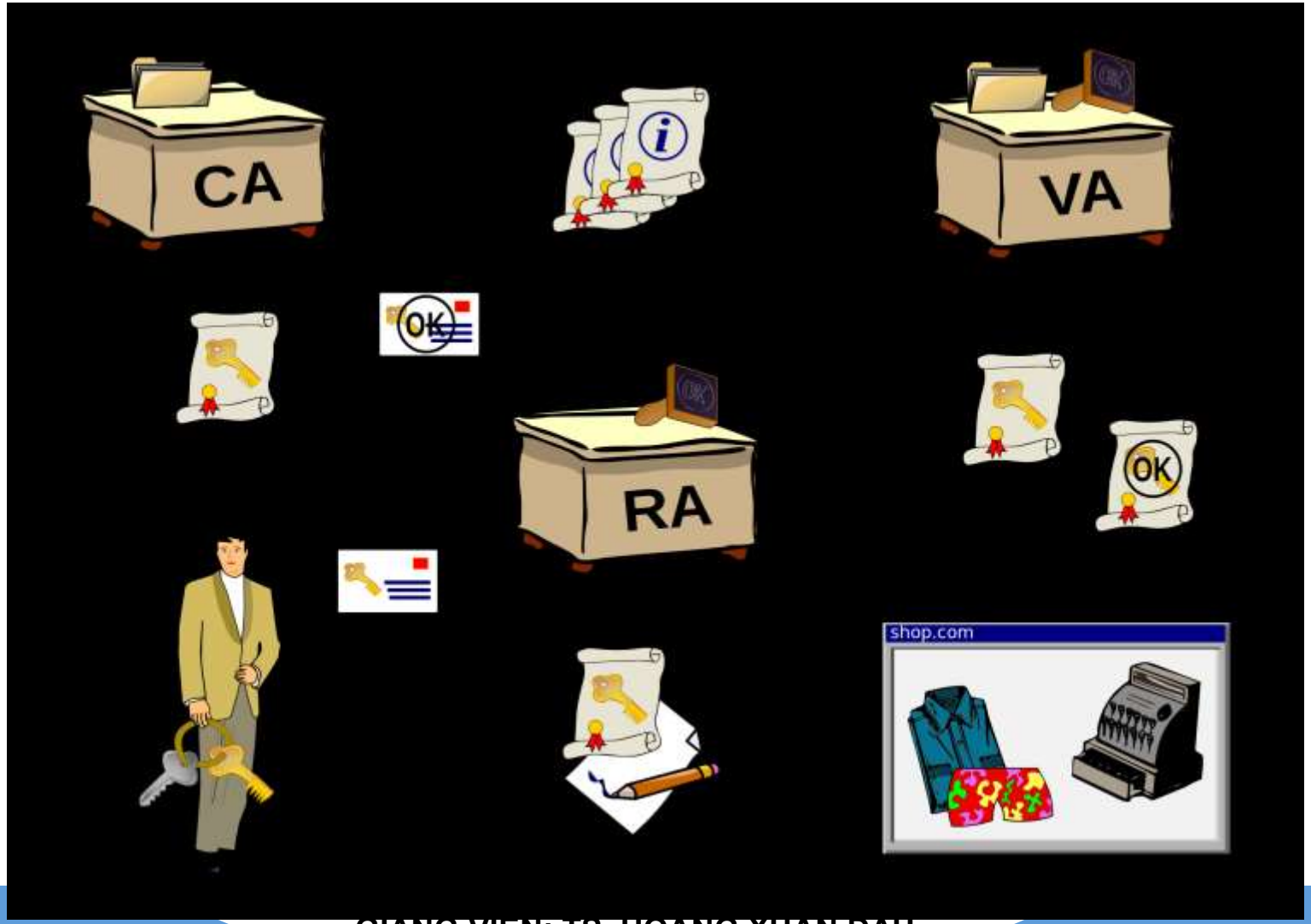
4.4.3 Hạ tầng khóa công khai - PKI

- Hạ tầng khóa công khai (Public-key infrastructure - PKI) là một tập các phần cứng, phần mềm, nhân lực, chính sách và các thủ tục để tạo, quản lý, phân phối, sử dụng, lưu trữ và thu hồi các chứng chỉ số;

4.4.3 Hạ tầng khóa công khai - PKI

- Một PKI gồm:
  - Certificate Authority (CA): Cơ quan cấp và kiểm tra chứng chỉ số;
  - Registration Authority (RA): Bộ phận kiểm tra thông tin nhận dạng của người dùng theo yêu cầu của CA;
  - Validation Authority (VA): Cơ quan xác nhận thông tin nhận dạng của người dùng thay mặt CA;
  - Central Directory (CD): Là nơi lưu danh mục và lập chỉ số các khóa;
  - Certificate Management System: Hệ thống quản lý chứng chỉ;
  - Certificate Policy: Chính sách về chứng chỉ;

4.4.3 Hạ tầng khóa công khai – Lưu đồ cấp và sử dụng



4.5 Các giao thức đảm bảo ATTT dựa trên mã hóa

- Các giao thức phổ biến đảm bảo an toàn thông tin dựa trên mã hóa gồm:
  - SSL/TLS (Secure Socket Layer/Transport Layer Security)
  - SET (Secure Electronic Transactions)
  - PGP (Pretty Good Privacy)
  - IPSec (IP Security)
  - SSH (Secure Shell)

4.5.1 Các giao thức đảm bảo ATTT – SSL/TLS

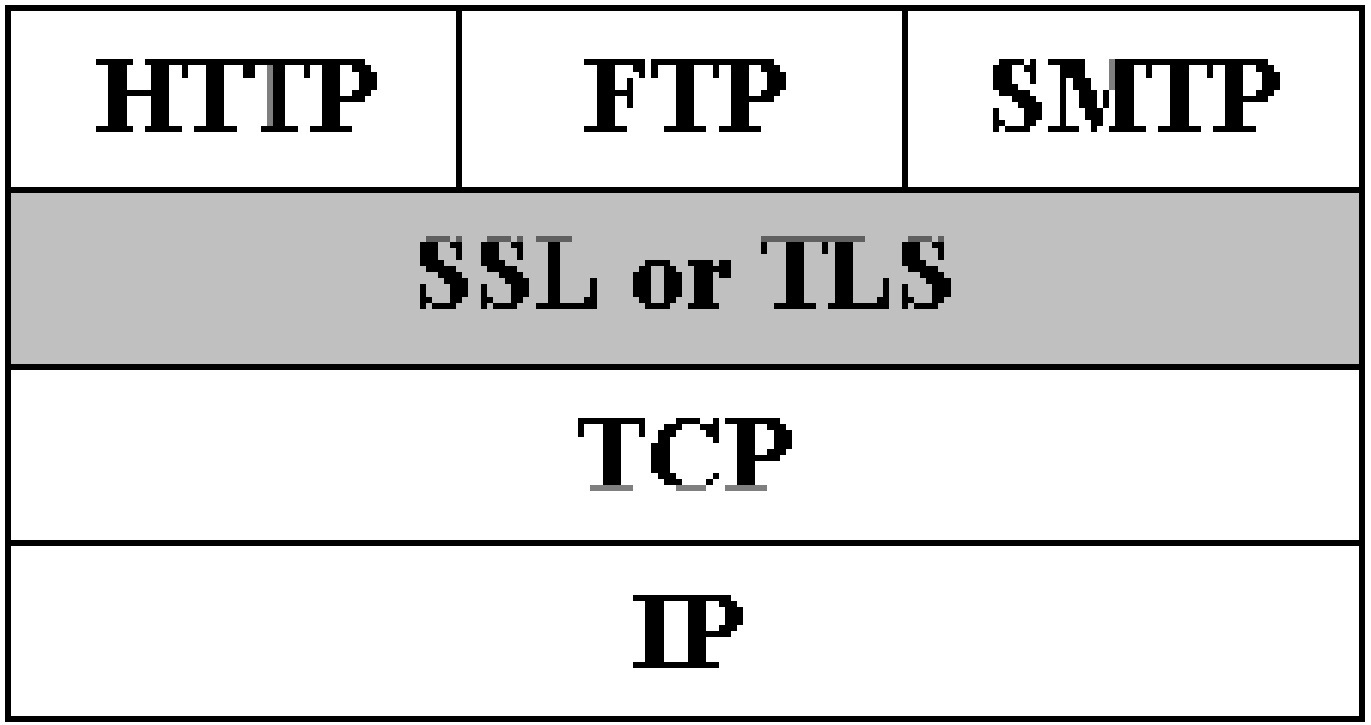
- SSL do công ty Netscape phát minh năm 1993;
  - Các phiên bản 1.0 (1993), 2.0 (1995) và 3.0 (1996);
  - SSL hiện ít được sử dụng do có nhiều lỗi và không được cập nhật.
- TLS được xây dựng vào năm 1999 dựa trên SSL 3.0 và do IETF phê chuẩn.
  - Các phiên bản của TLS: 1.0 (1999), 1.1 (2005), 1.2 (2008), 1.3 (2015 –draft).

4.5.1 Các giao thức đảm bảo ATTT – SSL/TLS

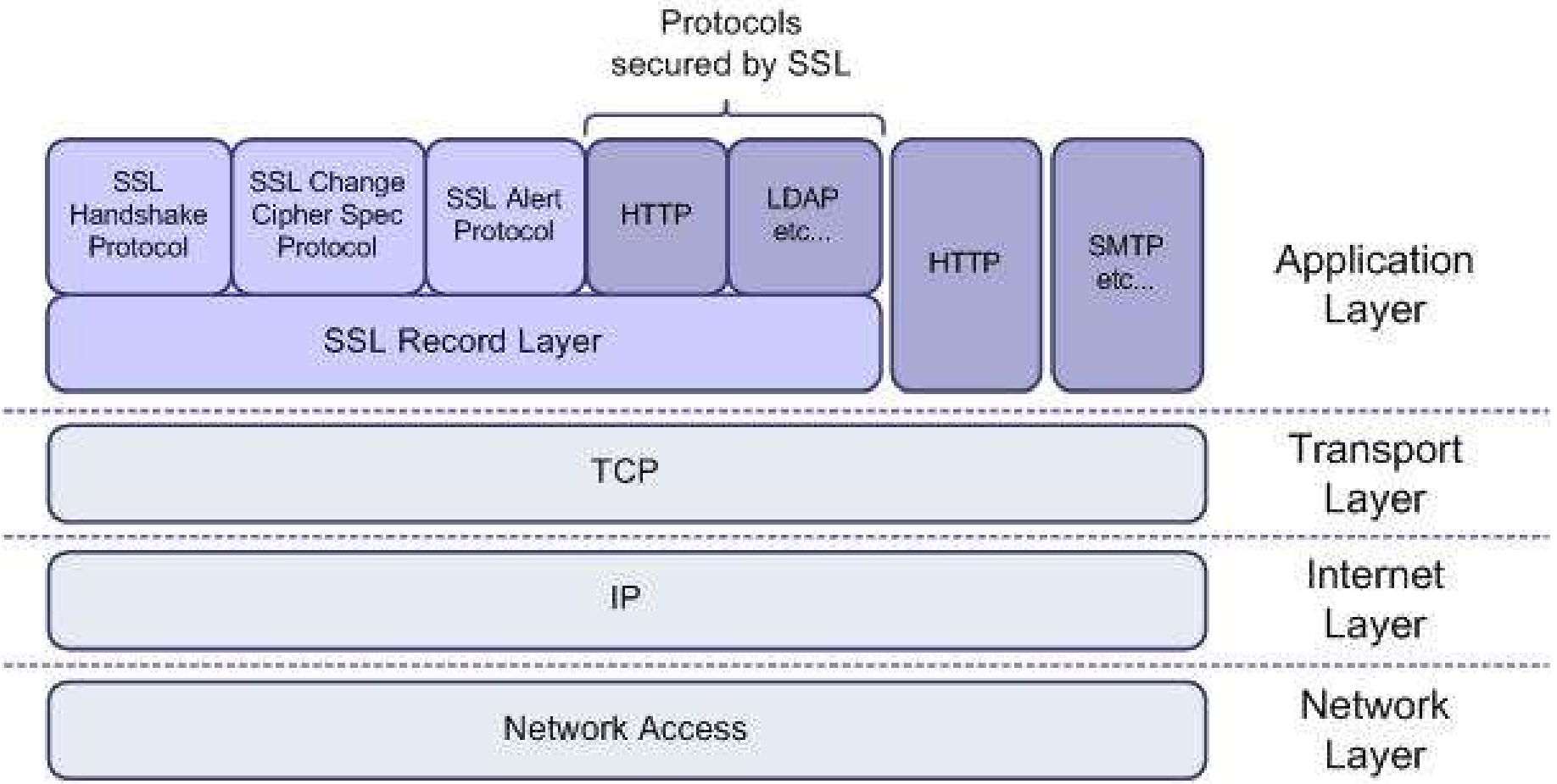
❖ Đặc điểm của SSL/TLS:

- Sử dụng mã hoá khoá công khai để trao đổi khoá phiên. Mỗi khoá phiên chỉ được sử dụng trong 1 phiên làm việc.
- Sử dụng khoá phiên và mã hoá khoá bí mật để mã hoá toàn bộ dữ liệu trao đổi.
- Sử dụng hàm băm có khóa (MAC) để đảm bảo tính toàn vẹn và xác thực thông điệp.
- Ít nhất một thực thể (thường là server) phải có chứng chỉ số cho khoá công khai (Public key certificate).

4.5.1 Các giao thức đảm bảo ATTT – SSL/TLS



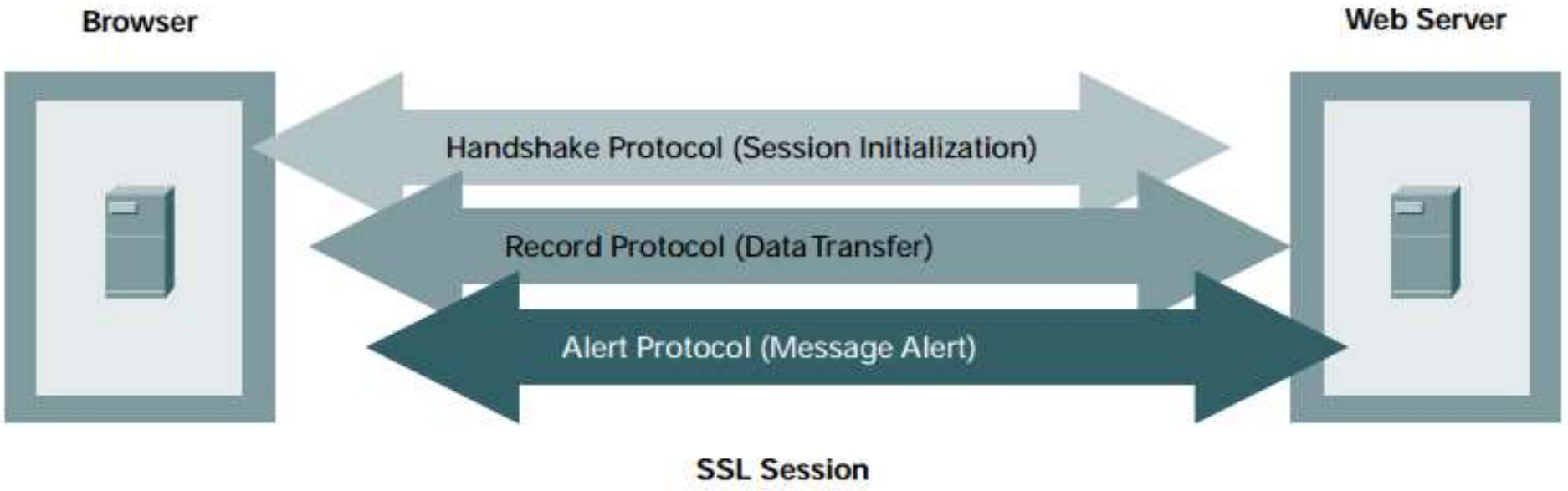
4.5.1 Các giao thức đảm bảo ATTT – SSL/TLS



4.5.1 Các giao thức đảm bảo ATTT – SSL/TLS

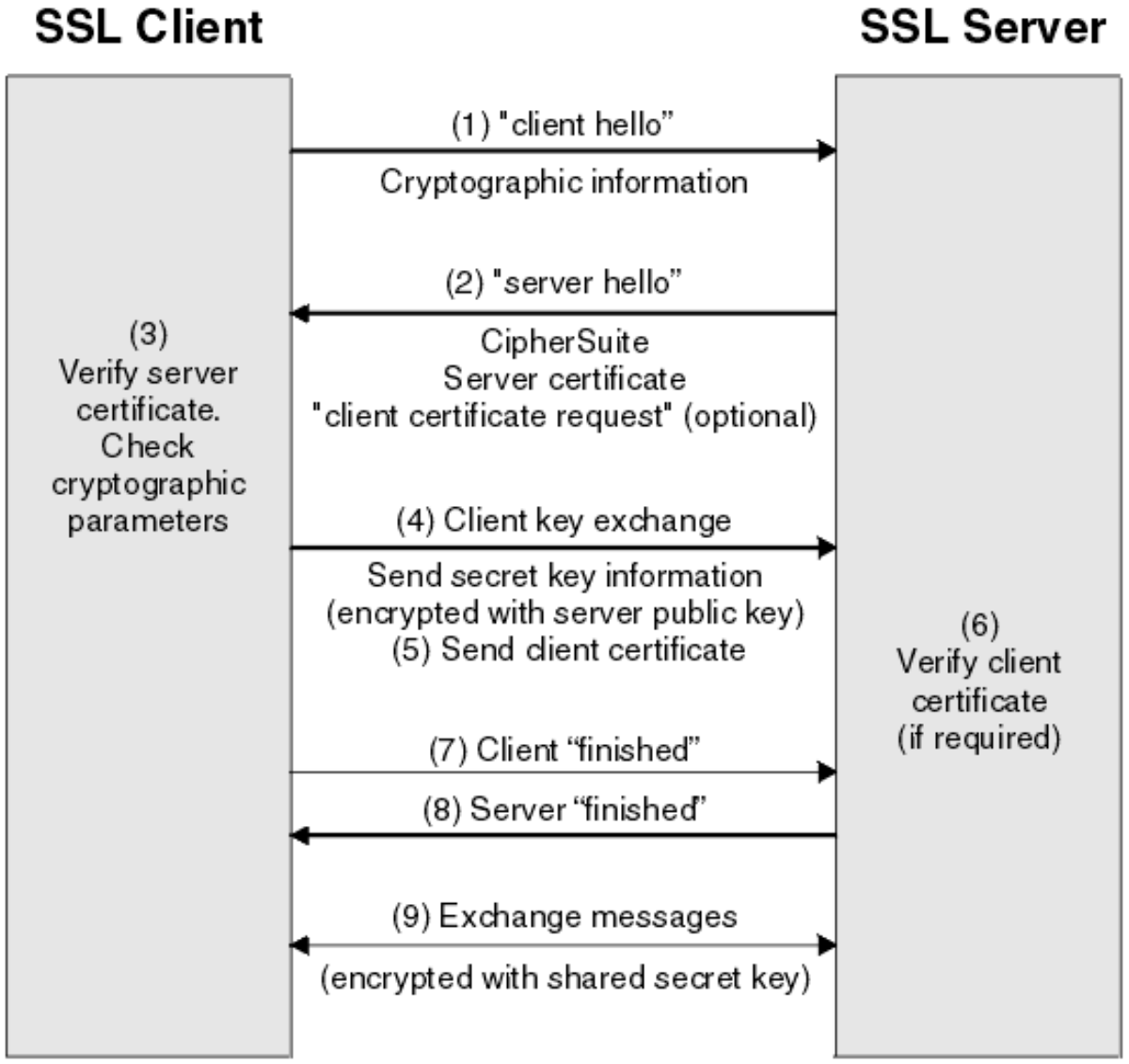
- Các giao thức con của SSL:
  - SSL Handshake Protocol: Giao thức bắt tay của SSL. Có nhiệm vụ trao đổi các thông điệp xác thực thực thể và thiết lập các thông số cho phiên làm việc;
  - SSL Change Cipher Spec Protocol: Giao thức thiết lập việc sử dụng các bộ mã hóa được hỗ trợ bởi cả 2 bên truyền thông;
  - SSL Alert Protocol: Giao thức cảnh bảo của SSL
  - SSL Record Protocol: Giao thức truyền các bản ghi của SSL có nhiệm vụ tạo đường hầm an toàn để chuyển thông tin đảm bảo tín bí mật, toàn vẹn và xác thực.

4.5.1 Các giao thức đảm bảo ATTT – SSL/TLS



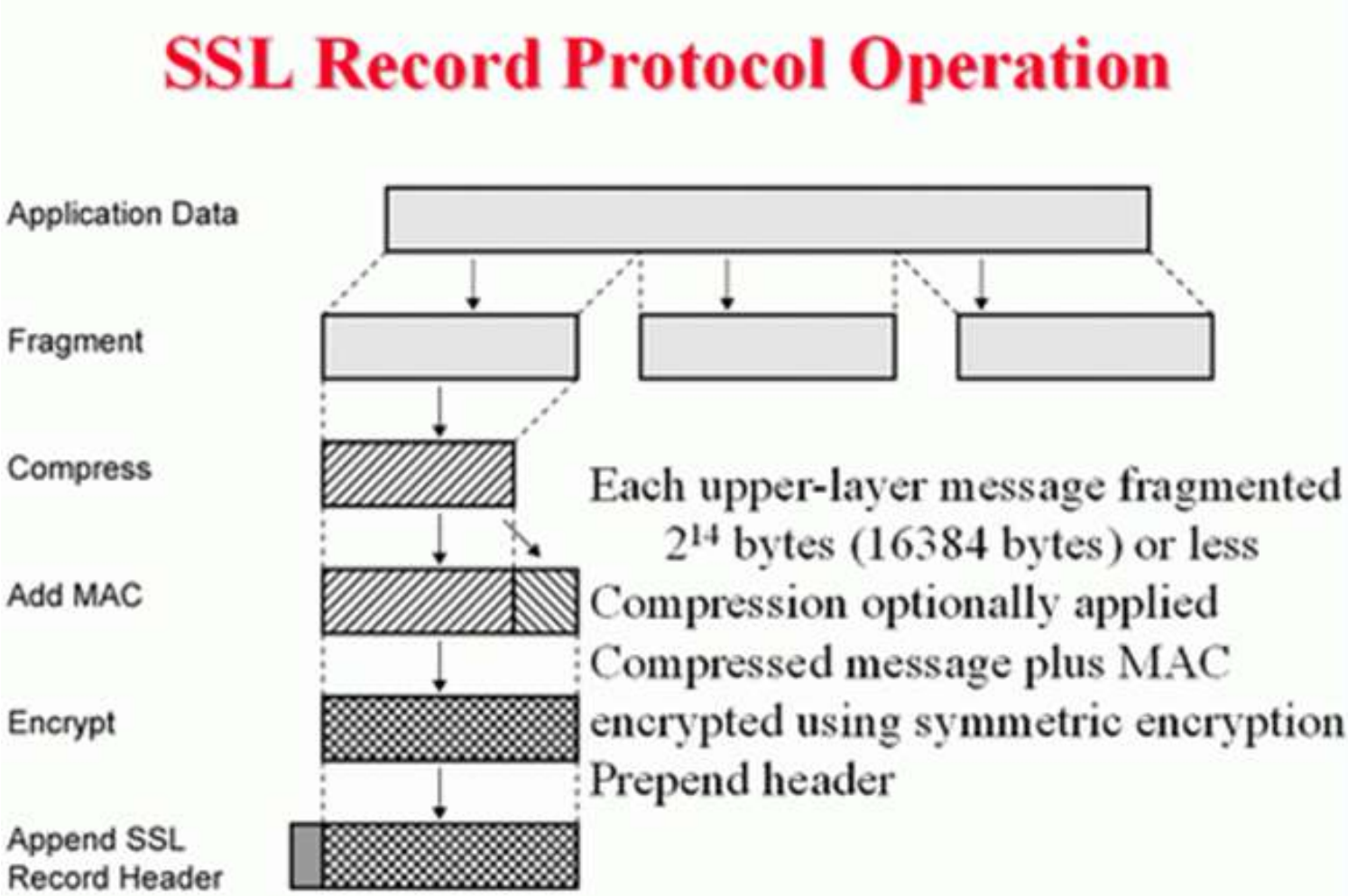
4.5.1 Các giao thức đảm bảo ATTT – SSL/TLS

- SSL Handshake Protocol:



4.5.1 Các giao thức đảm bảo ATTT – SSL/TLS

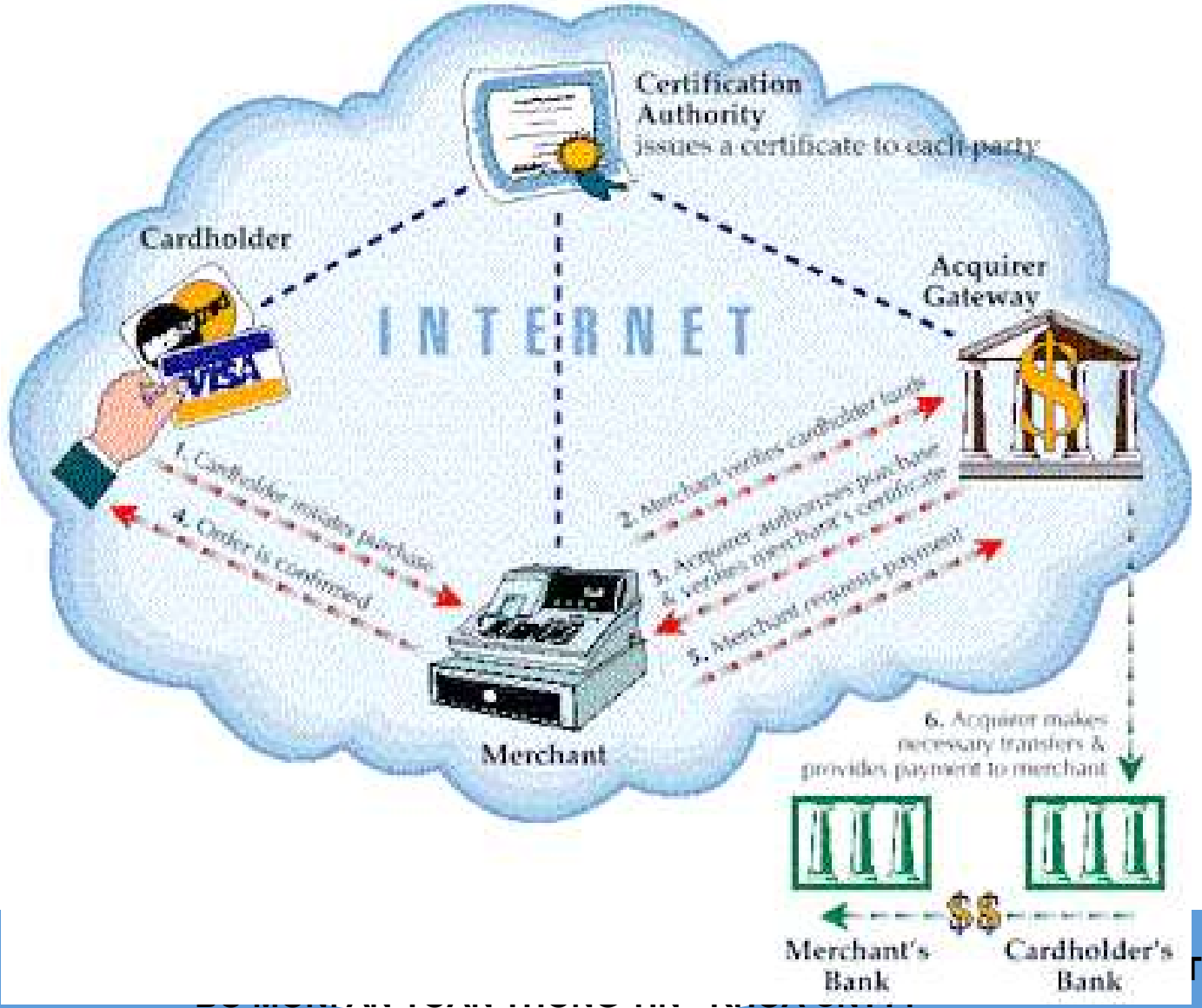
- Hoạt động của SSL:



4.5.2 Các giao thức đảm bảo ATTT – SET

- SET là giao thức cho phép thanh toán điện tử an toàn sử dụng thẻ tín dụng do 2 công ty Visa International và MasterCard phát triển;
- SET có khả năng đảm bảo các thuộc tính sau của thông tin truyền:
  - Bí mật thông tin
  - Toàn vẹn thông tin
  - Xác thực tài khoản chủ thẻ
  - Xác thực nhà cung cấp

4.5.2 Các giao thức đảm bảo ATTT – SET

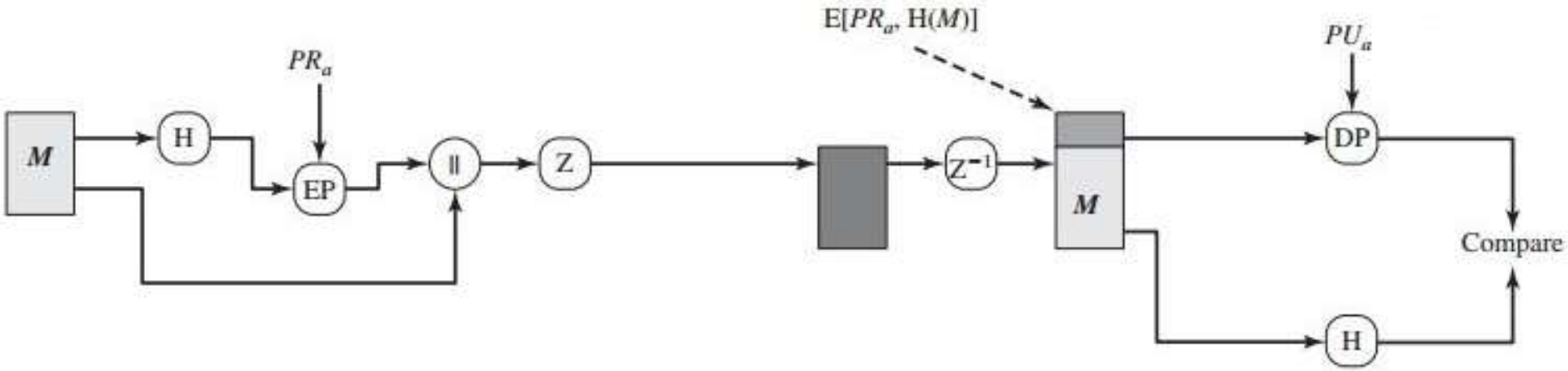


4.5.3 Các giao thức đảm bảo ATTT – PGP

- PGP do Philip Zimmermann phát triển năm 1991:
  - Cung cấp tính riêng tư
  - Cung cấp tính xác thực
- PGP được sử dụng rộng rãi và đã được thừa nhận thành chuẩn (RFC 3156).
- PGP cho phép:
  - Mã hoá dữ liệu sử dụng mã hoá khoá bí mật và khoá công khai
  - Tạo và kiểm tra chữ ký điện tử.

4.5.3 Các giao thức đảm bảo ATTT – PGP

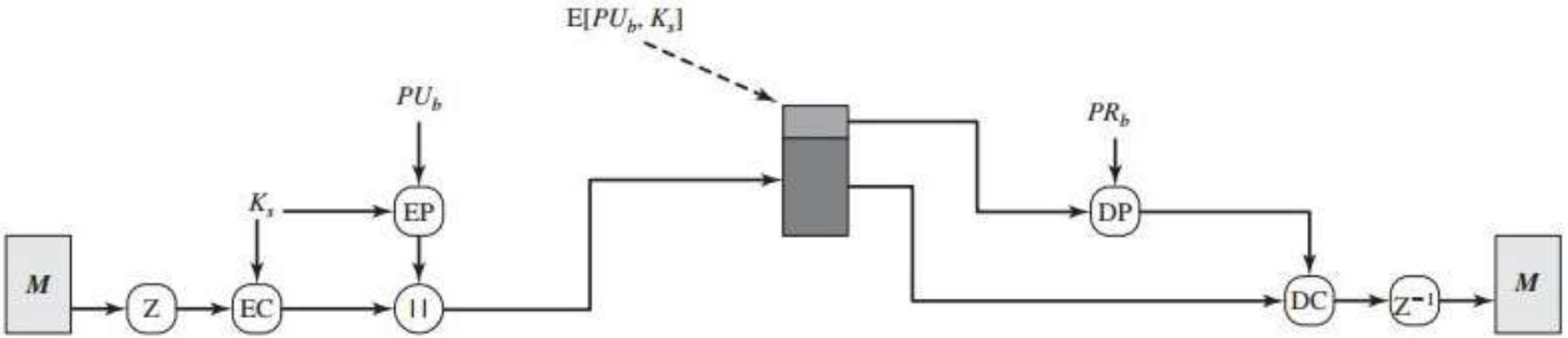
Bên gửi =====> Bên nhận



Mô hình PGP chỉ đảm bảo tính xác thực thông điệp

4.5.3 Các giao thức đảm bảo ATTT – PGP

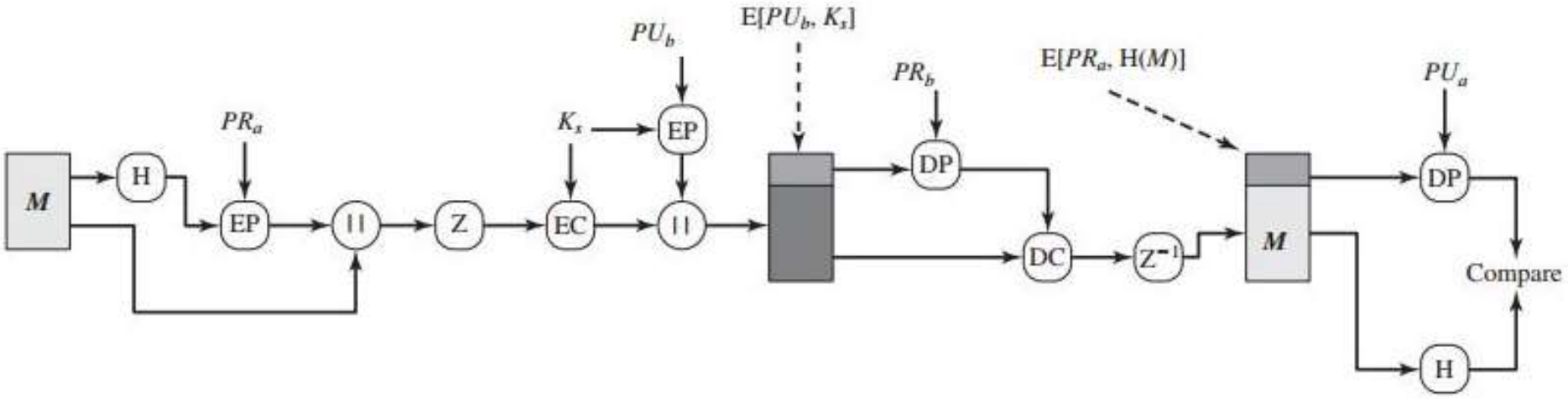
Bên gửi =====> Bên nhận



Mô hình PGP chỉ đảm bảo tính bí mật thông điệp

4.5.3 Các giao thức đảm bảo ATTT – PGP

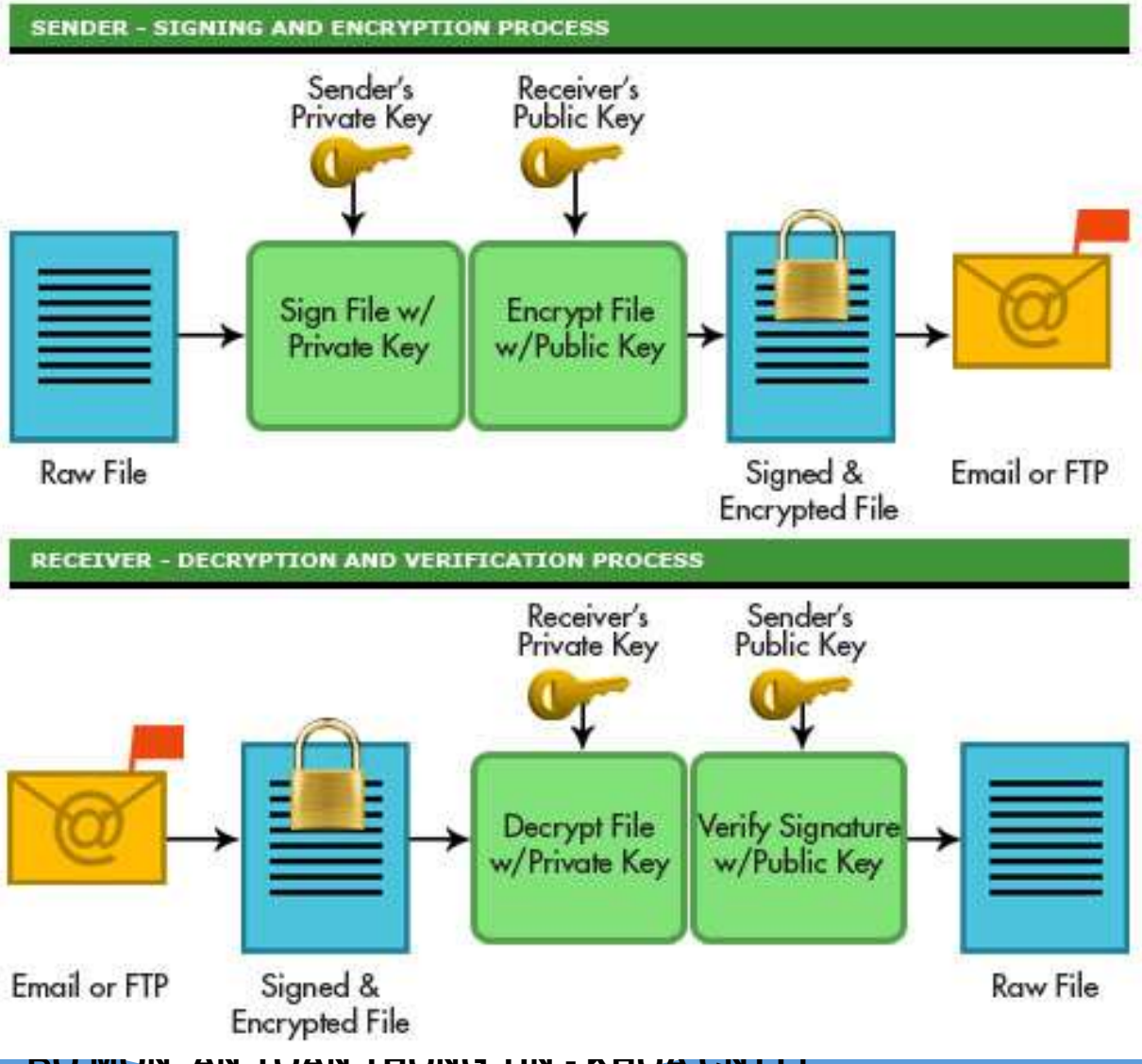
Bên gửi =====> Bên nhận



Mô hình PGP đảm bảo tính bí mật và xác thực thông điệp

4.5.3 Các giao thức đảm bảo ATTT – PGP

Mô hình trao đổi file đảm bảo tính bí mật và toàn vẹn sử dụng mã hóa khóa công khai và chữ ký số



Tổng kết các PP đảm bảo ATTT dựa trên mã hóa