HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

# AN TOÀN BẢO MẬT HỆ THỐNG THÔNG TIN

## CHƯƠNG 4 – CÁC KỸ THUẬT MÃ HÓA THÔNG TIN


**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. Quản lý khóa và phân phối khóa
5. Chữ ký số, chứng chỉ số và PKI
6. Các giao thức đảm bảo an toàn thông tin dựa trên mã hóa.

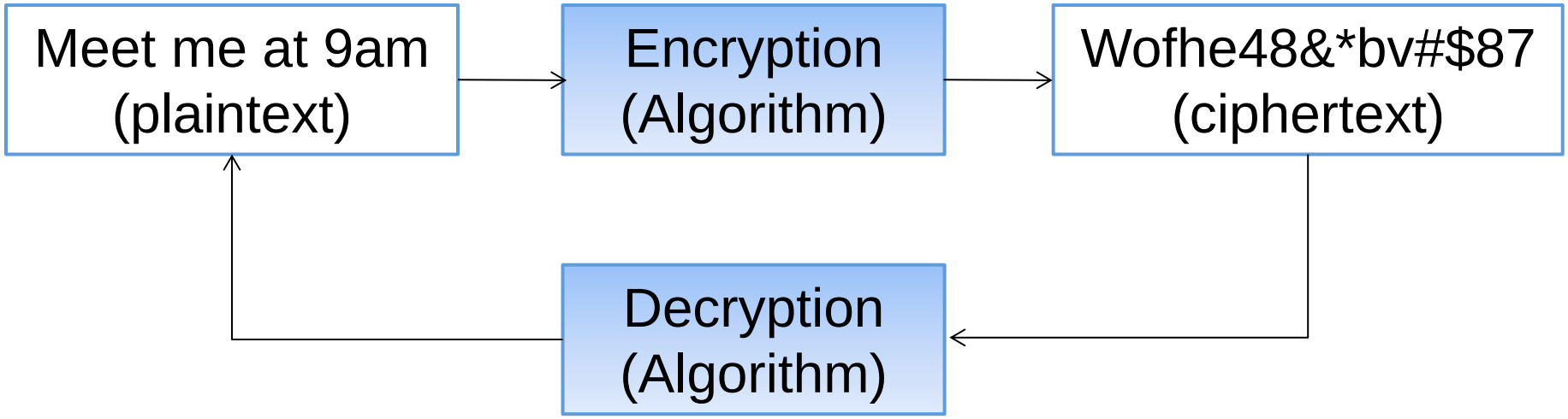
# 4.1 Khái quát về mã hóa thông tin và ứng dụng

- Mã hóa thông tin là gì?
- Vai trò của mã hóa
- Các thành phần của một hệ mã hóa
- Lịch sử mã hóa
- Mã hóa dòng và mã hóa khối
- Các tiêu chuẩn đánh giá hệ mã hóa
- 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.
- Một hệ mã hóa gồm 2 khâu:
  - Mã hóa (encryption)
  - Giải mã (decryption)

# 4.1.1 Mã hóa thông tin là gì?




Mã hóa và giải mã

# 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)
- 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.

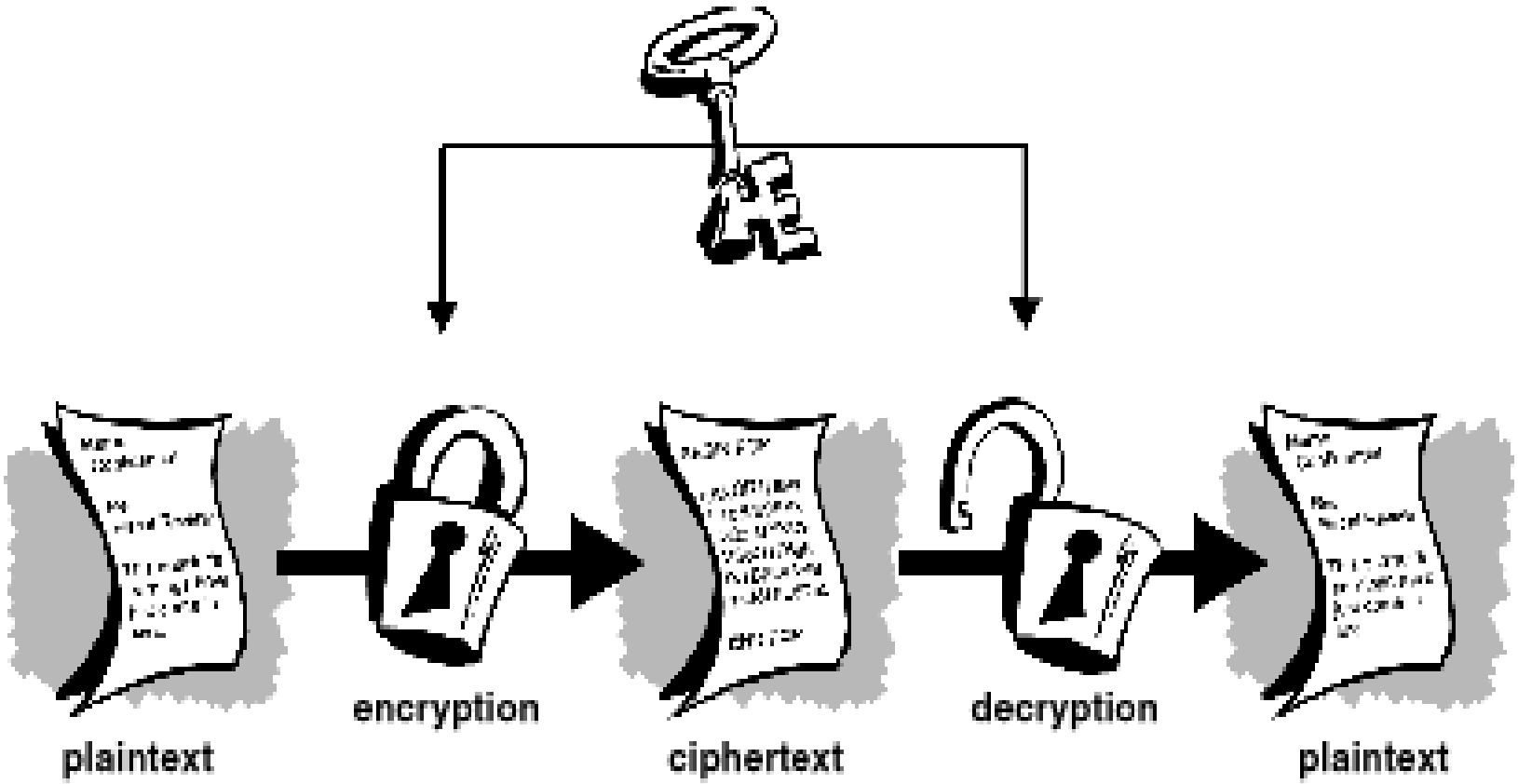
# 4.1.1 Mã hóa thông tin – Các thuật ngữ

- Mã hóa sử dụng một thuật toán (Algorithm) để mã hóa thông tin.
- Một bộ mã hóa (Cipher) là một giải thuật để mã hóa và giải mã thông tin.
- Khóa/Chìa (Key) là một chuỗi được sử dụng trong giải thuật mã hóa và giải mã.
- Mã hóa khóa bí mật (Secret key cryptography): một khóa được sử dụng cho cả giải thuật mã hóa và giải mã.
- Mã hóa khóa công khai (Public key cryptography): một cặp khóa được sử dụng, trong đó khóa công khai để mã hóa, khóa bí mật để giải mã.

# 4.1.1 Mã hóa thông tin – Các thuật ngữ

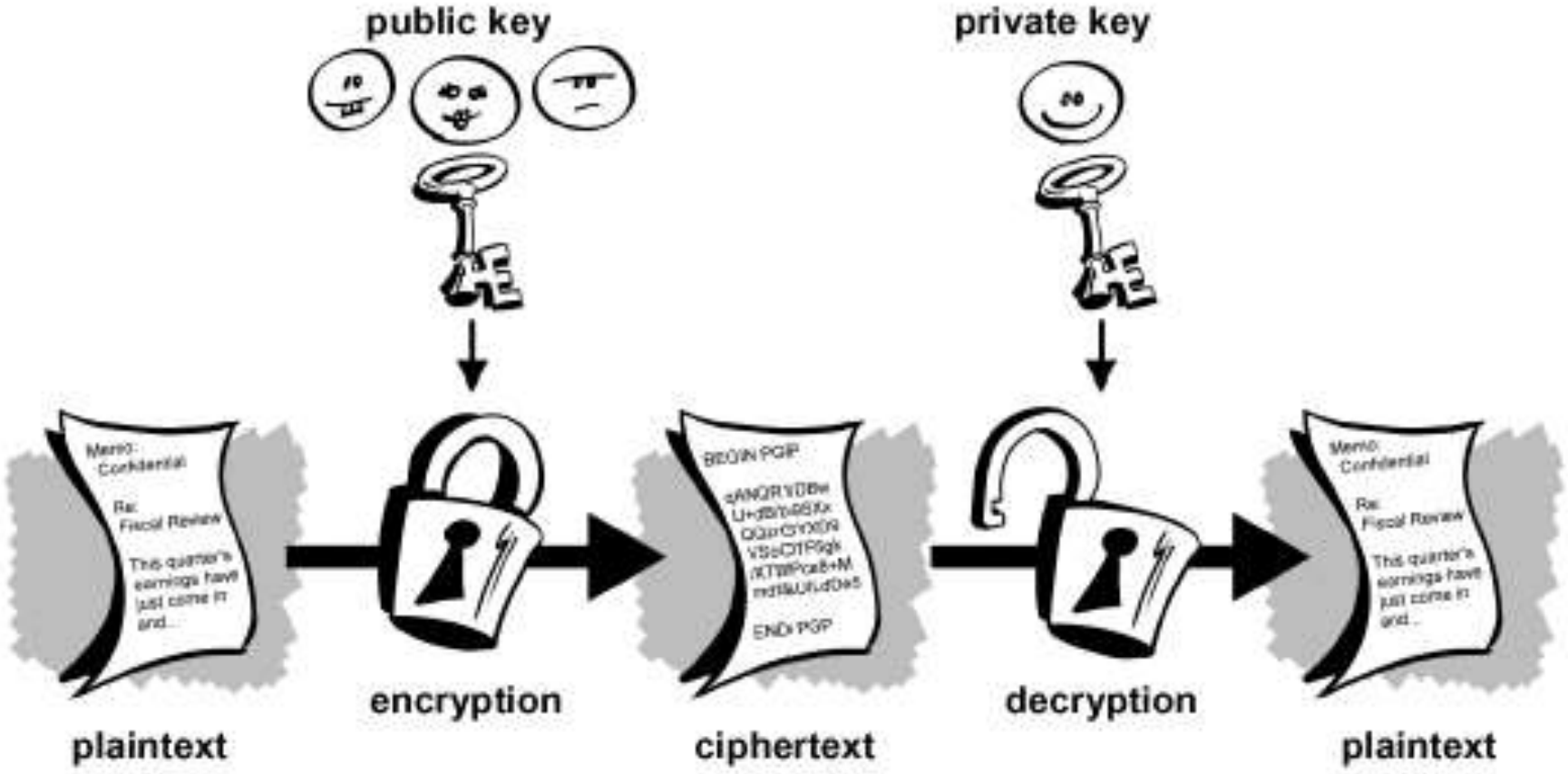
- Không gian khóa (Keyspace) là tổng số khóa có thể có của một hệ mã hóa.
  - Ví dụ nếu sử dụng khóa kích thước 64 bít → không gian khóa là $2^{64}$.
- Hàm băm (Hash function) là một ánh xạ chuyển các dữ liệu có kích thước thay đổi về dữ liệu có kích thước cố định.
  - Hàm băm 1 chiều (One-way hash function) 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.
- Phá 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 khóa mã (Key).

# 4.1.1 Mã hóa thông tin – Các thuật ngữ




Mã hóa khóa bí mật

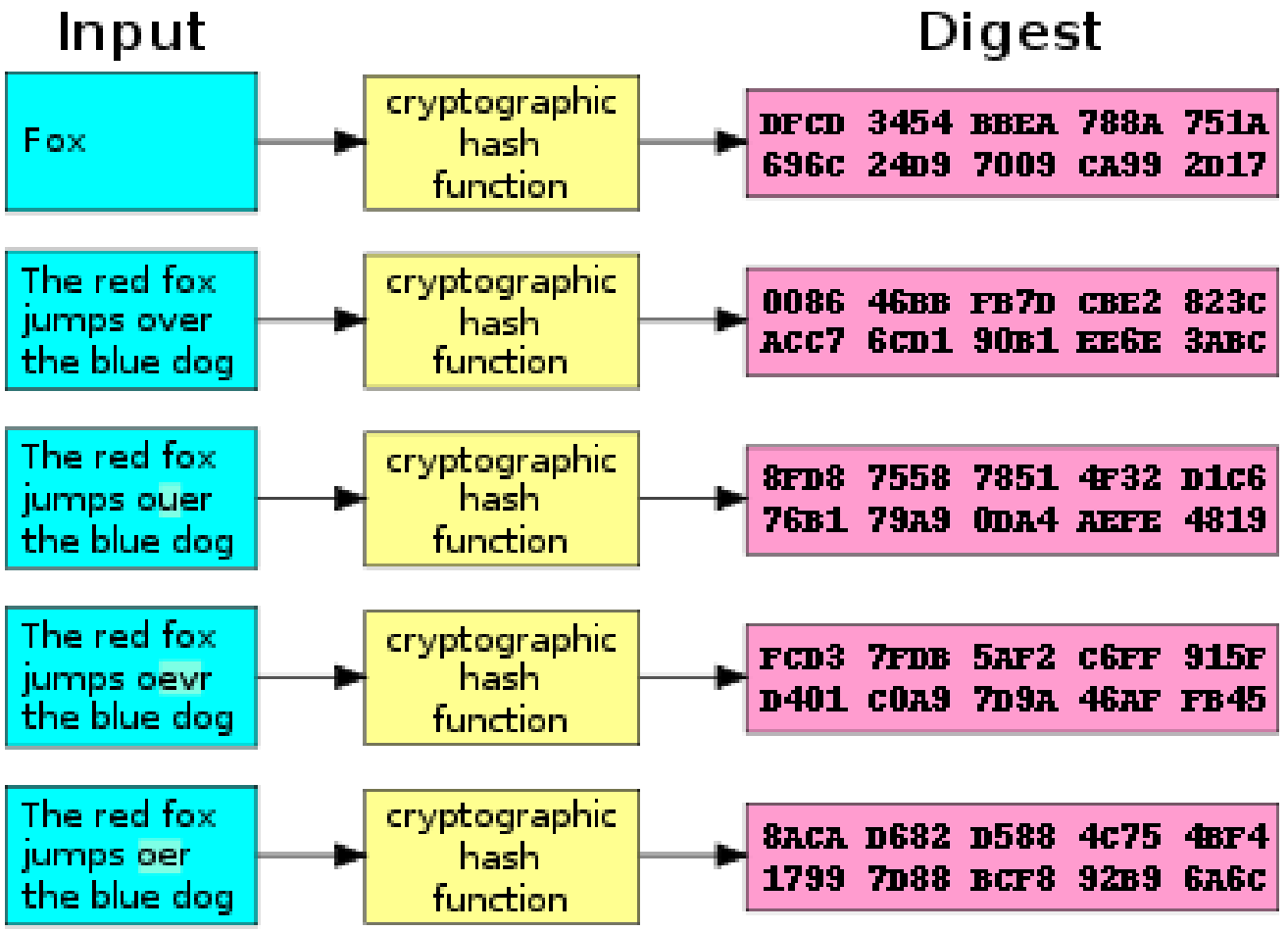
# 4.1.1 Mã hóa thông tin – Các thuật ngữ




Mã hóa khóa công khai

# 4.1.1 Mã hóa thông tin – Các thuật ngữ

Ví dụ về hàm băm (hash function)



## 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 để giải mã các thông điệp của quân địch.

## 4.1.4 Lịch sử mã hóa

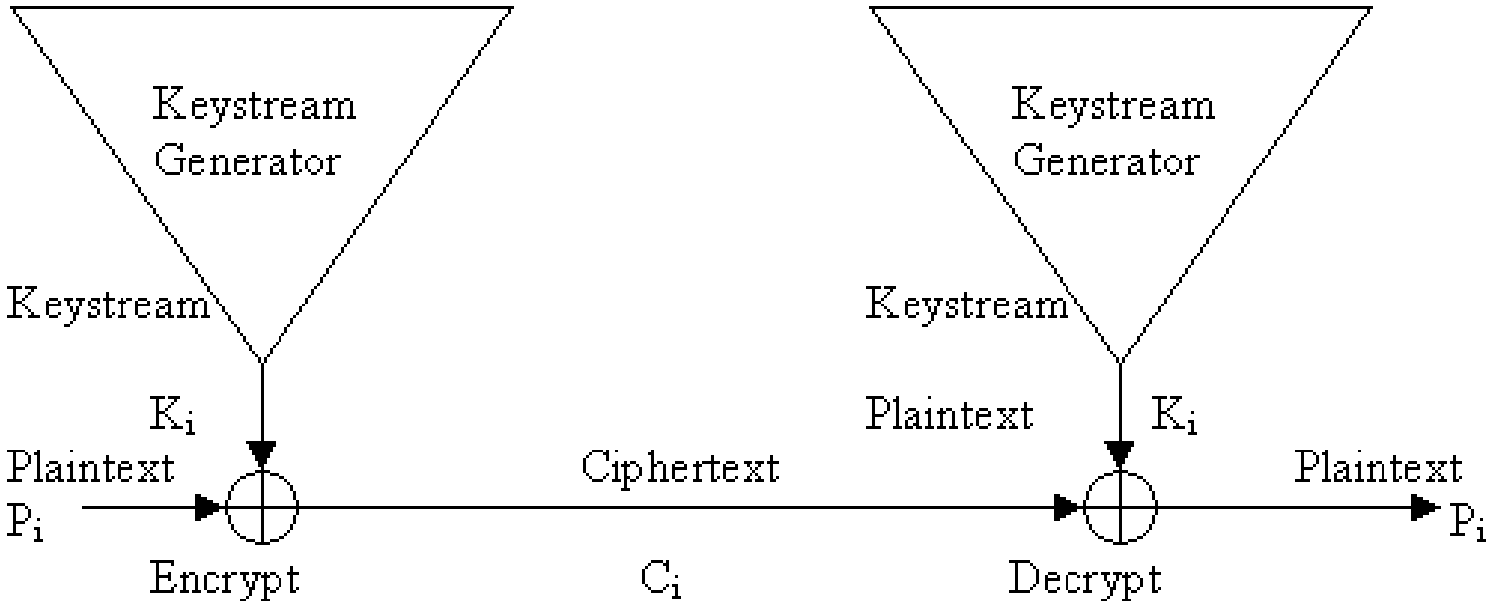
- Năm 1976 chuẩn mã hóa DES (Data Encryption Standard) được cơ quan an ninh quốc gia Mỹ thừa nhận và sử dụng rộng rãi.
- Năm 1976, hai nhà khoa học Whitman Diffie và Martin Hellman đã đưa ra khái niệm mã hóa bất đối xứng (Asymetric key cryptography) hay mã hóa khóa công khai (Public key cryptography)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

# 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 PGP (Pretty Good Privacy) ra đời.
- Năm 2000, chuẩn mã hóa AES (Advanced Encryption Standard) được thừa nhận.

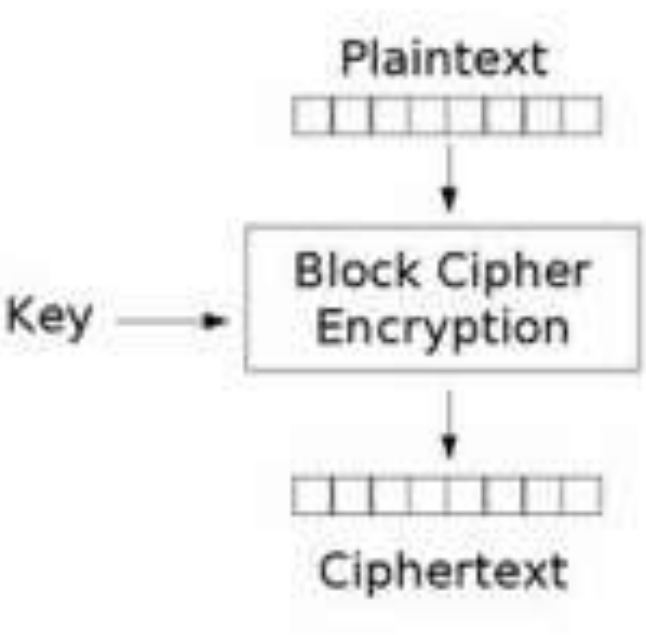
# 4.1.5 Mã hóa dòng và mã hóa khối

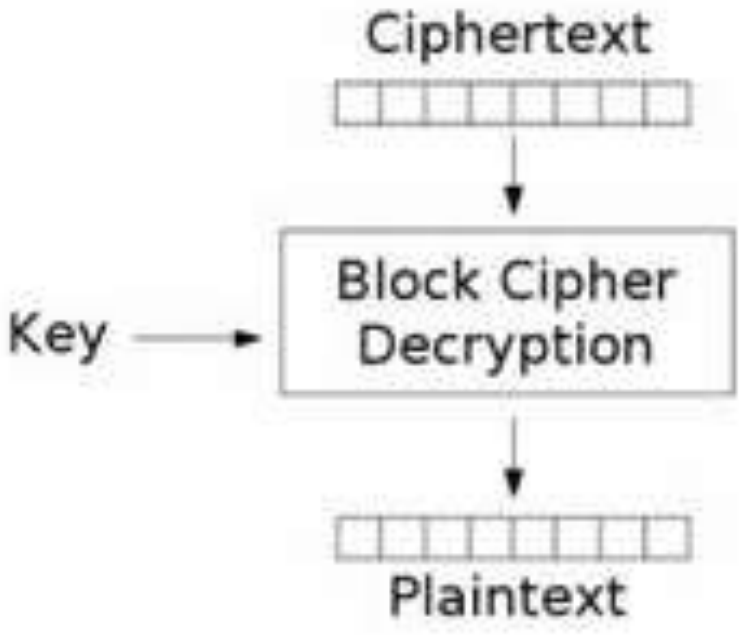
- Mã hóa dòng (Stream cipher) là kiểu mã hóa mà từng bít (hoặc ký tự) của dữ liệu được kết hợp với từng bít (hoặc ký tự) tương ứng của khóa để tạo thành bản mã.



# 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 (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.
  - Chế độ CBC (Cipher-Block Chaining): cùng khối bản rõ đầu vào, khối bản mã giống nhau với cùng khóa và phần nối đuôi. Khối mã $c_j$ phụ thuộc vào khối rõ $x_j$ và các khối rõ trước đó ($x_1$-$x_{j-1}$)
  - 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 ($x_1$-$x_{j-1}$).
  - Chế độ OFB (Output Feedback): cùng khối bản rõ đầu vào, khối bản mã khác nhau. 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.
- **Tính năng** (functionality): hệ thống có thể được sử dụng cho nhiều mục đích bảo mật.
- **Chế độ hoạt động** (methods of operation): cung cấp các tính năng khác nhau theo chế độ hoạt động.
- **Hiệu năng** (performance): có thể được đo bằng tốc độ mã hoá (bits/giây).
- **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 cập
  - 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 - 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 Các phương pháp mã hóa - 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 Các phương pháp mã hóa - 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

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

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

# 4.2 Các phương pháp mã hóa - 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 Các phương pháp mã hóa - 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 Các phương pháp mã hóa - 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 Các phương pháp mã hóa - 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 Các phương pháp mã hóa - 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 Các phương pháp mã hóa – PP 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 Các phương pháp mã hóa - Các 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.
- Một số hàm băm thông dụng:
  - MD2, MD4, MD5 (128 bit)
  - MD6 (0-512 bit)
  - SHA0, SHA1 (160 bit)
  - SHA2, SHA3 (SHA256, SHA384, SHA512)
  - CRC32 (32 bit)

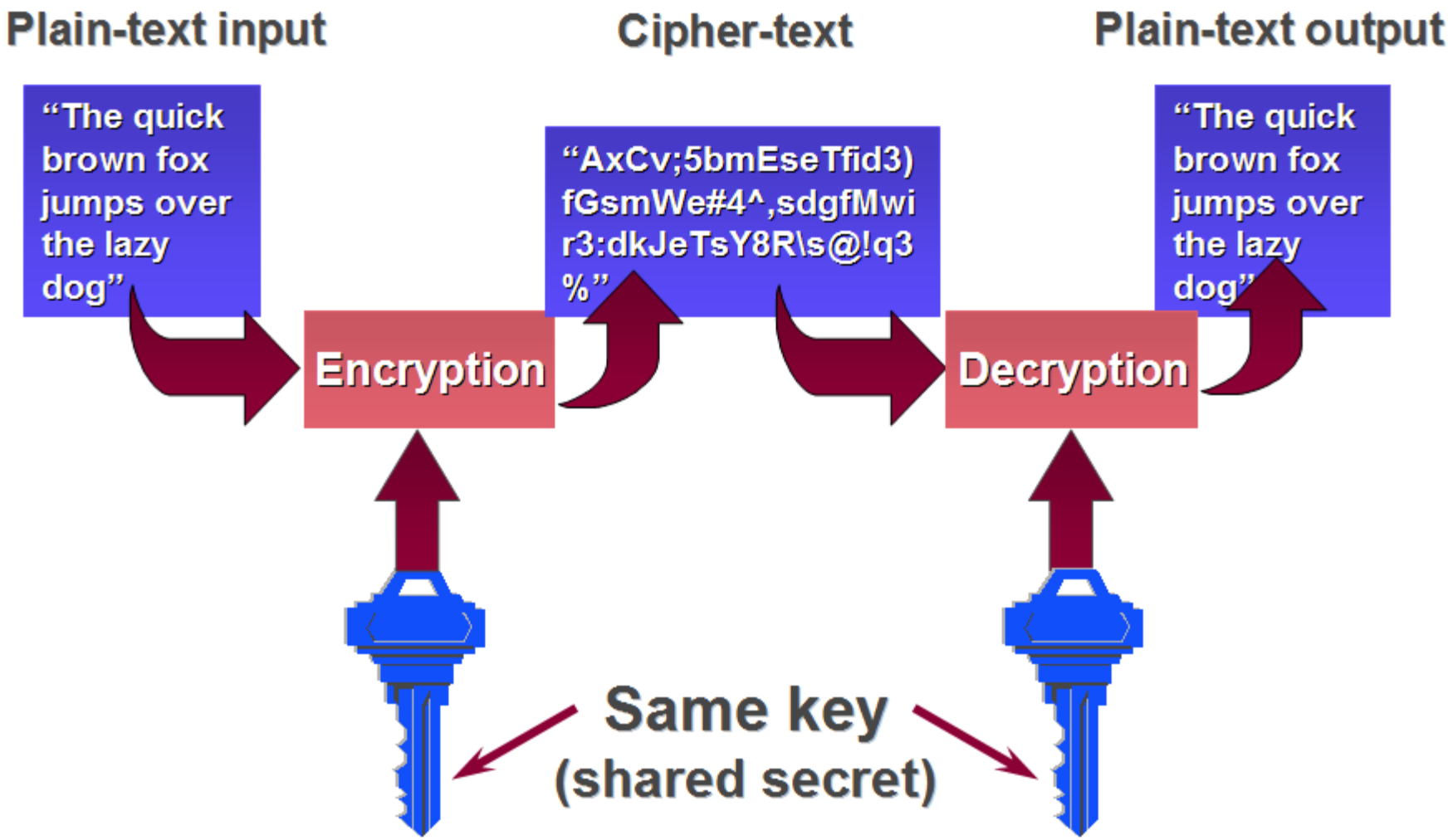
# 4.3 Các giải thuật mã hóa

- Các giải thuật mã hóa khóa đối xứng
  - DES, Triple-DES
  - AES, IDEA
  - Blowfish, Twofish
  - RC4, RC5
- Các giải thuật mã hóa khóa bất đối xứng
  - RSA
  - Rabin
  - ElGamal
- 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 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

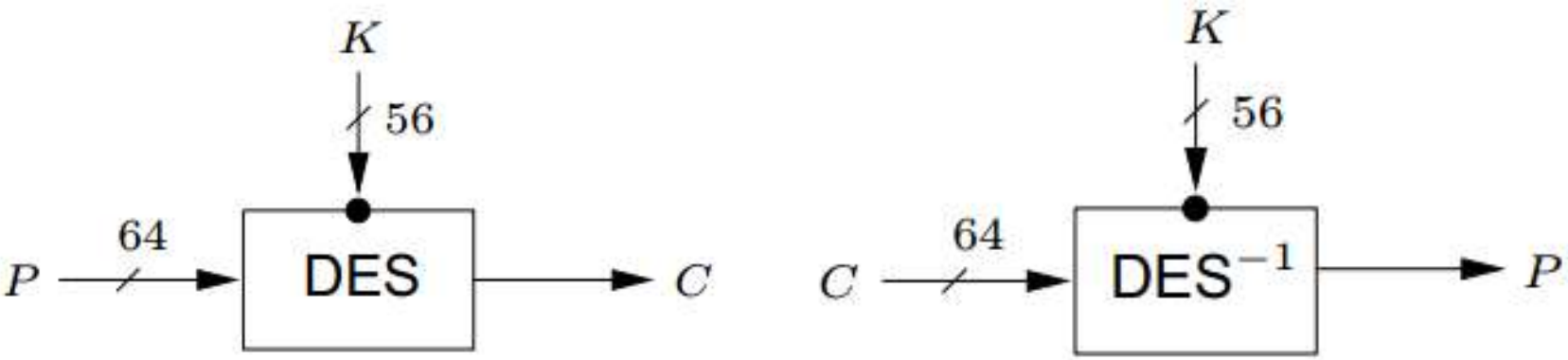


# 4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

- 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.
- Đặc điểm:
  - 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 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 $C$

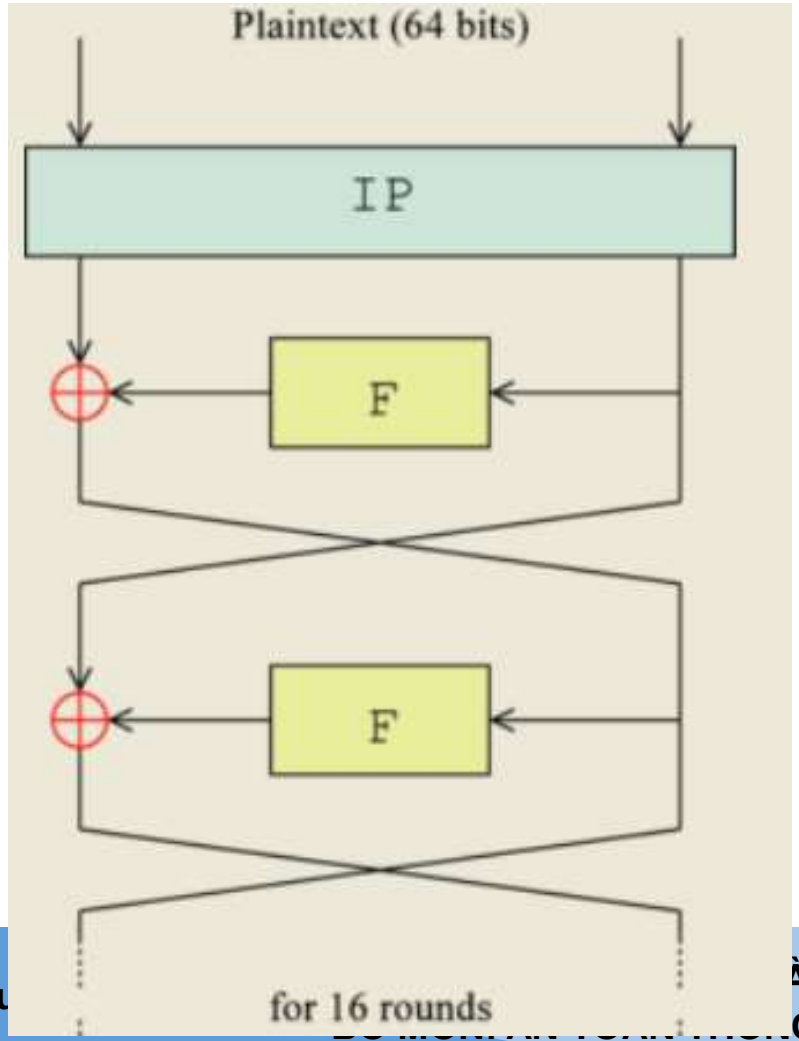
key $K$

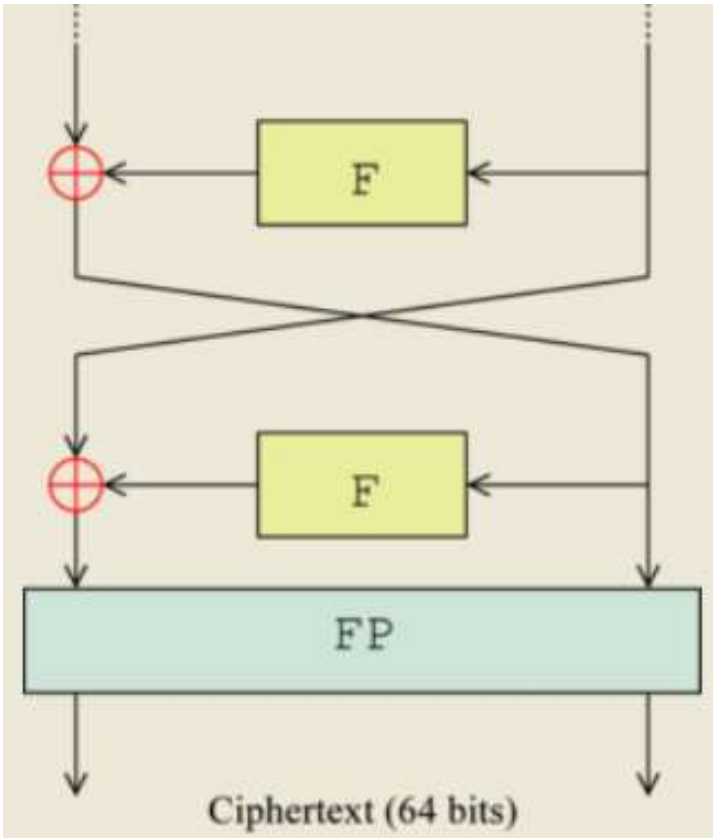
# 4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

- 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

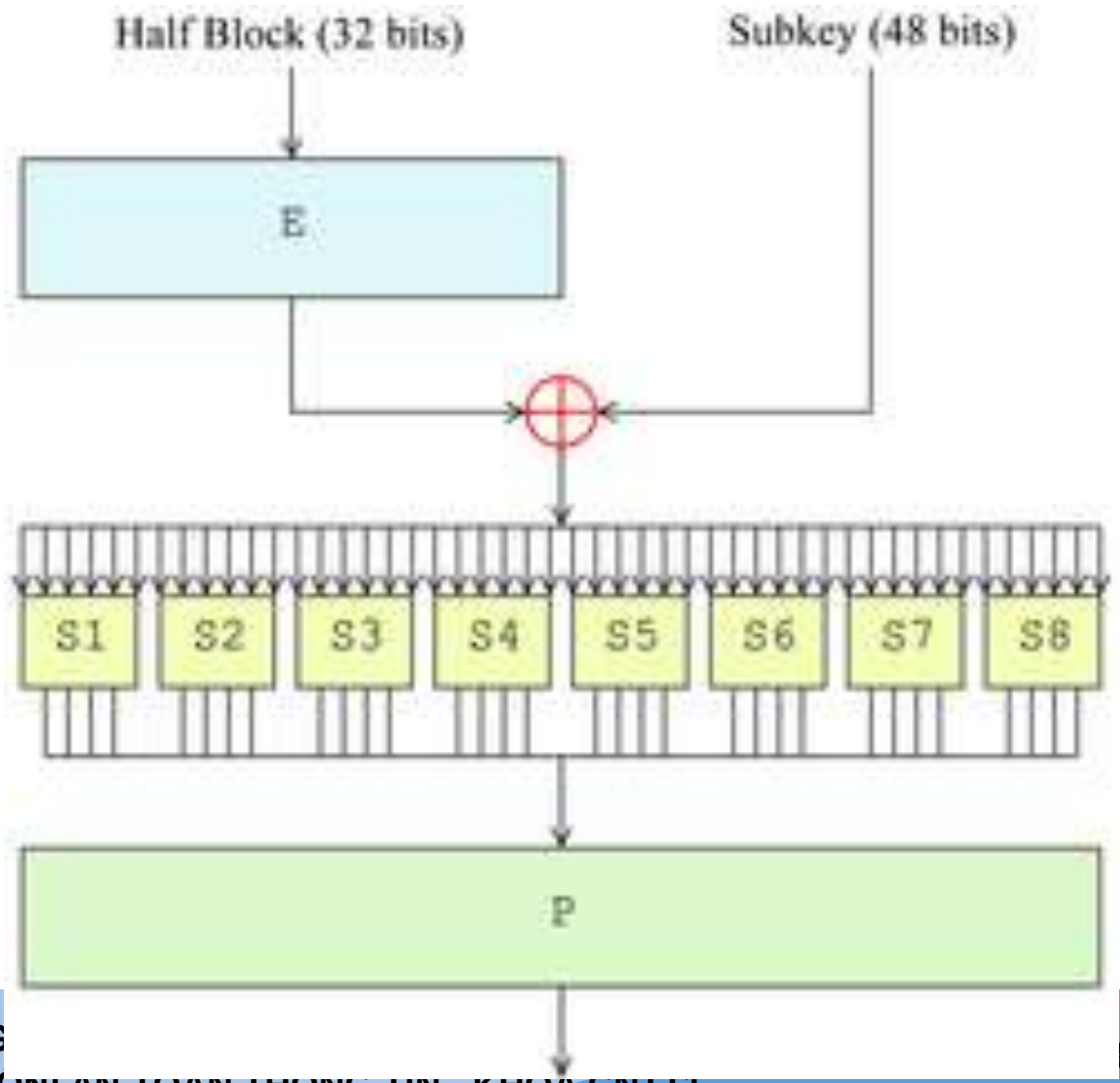


# 4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

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.

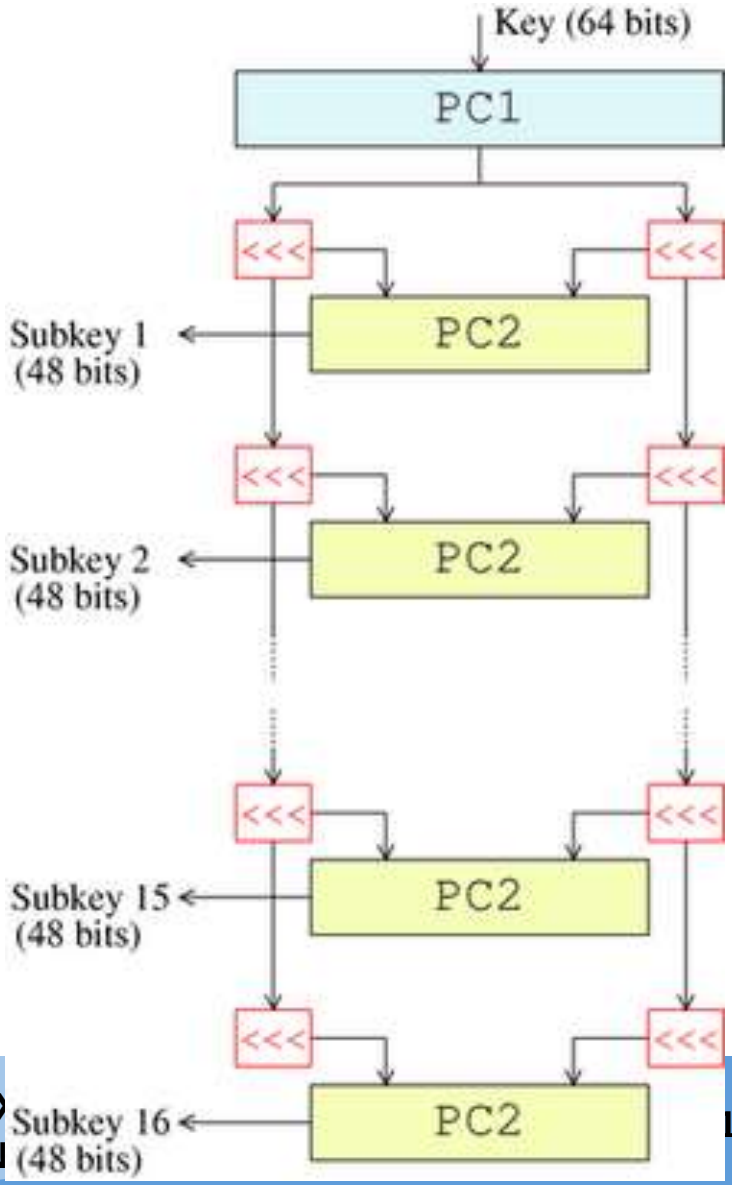


# 4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

- 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.



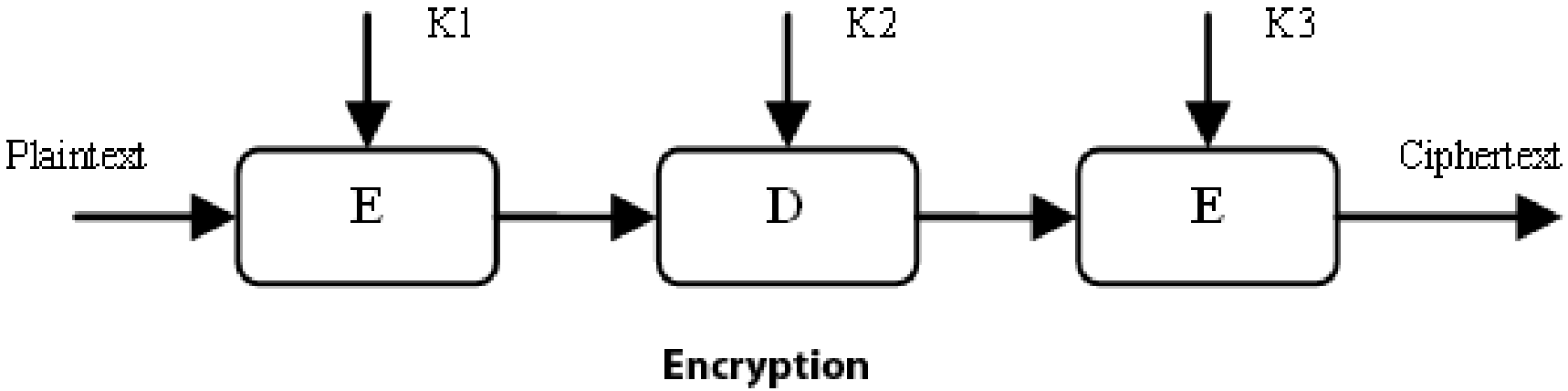
# 4.3.1 Các giải thuật mã hóa khóa đối xứng - DES

- 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.

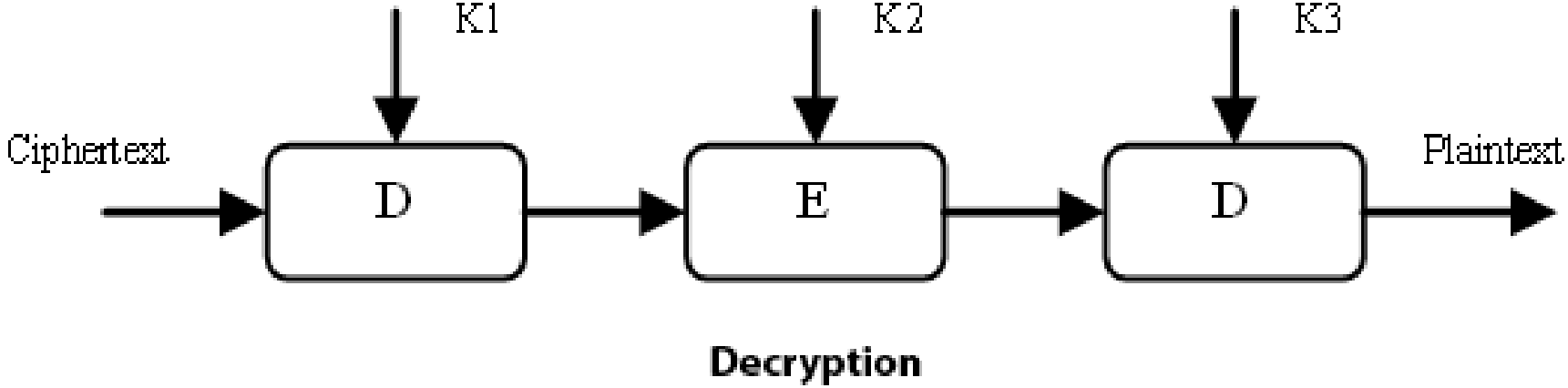
# 4.3.1 Các giải thuật mã hóa khóa đối xứng – Triple DES

- 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;
- Kích thước khối dữ liệu của AES là 128 bít;
- Kích thước khóa có thể là 128, 192, hoặc 256 bit (là bội của 32 và lớn nhất là 256 bít);
- 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

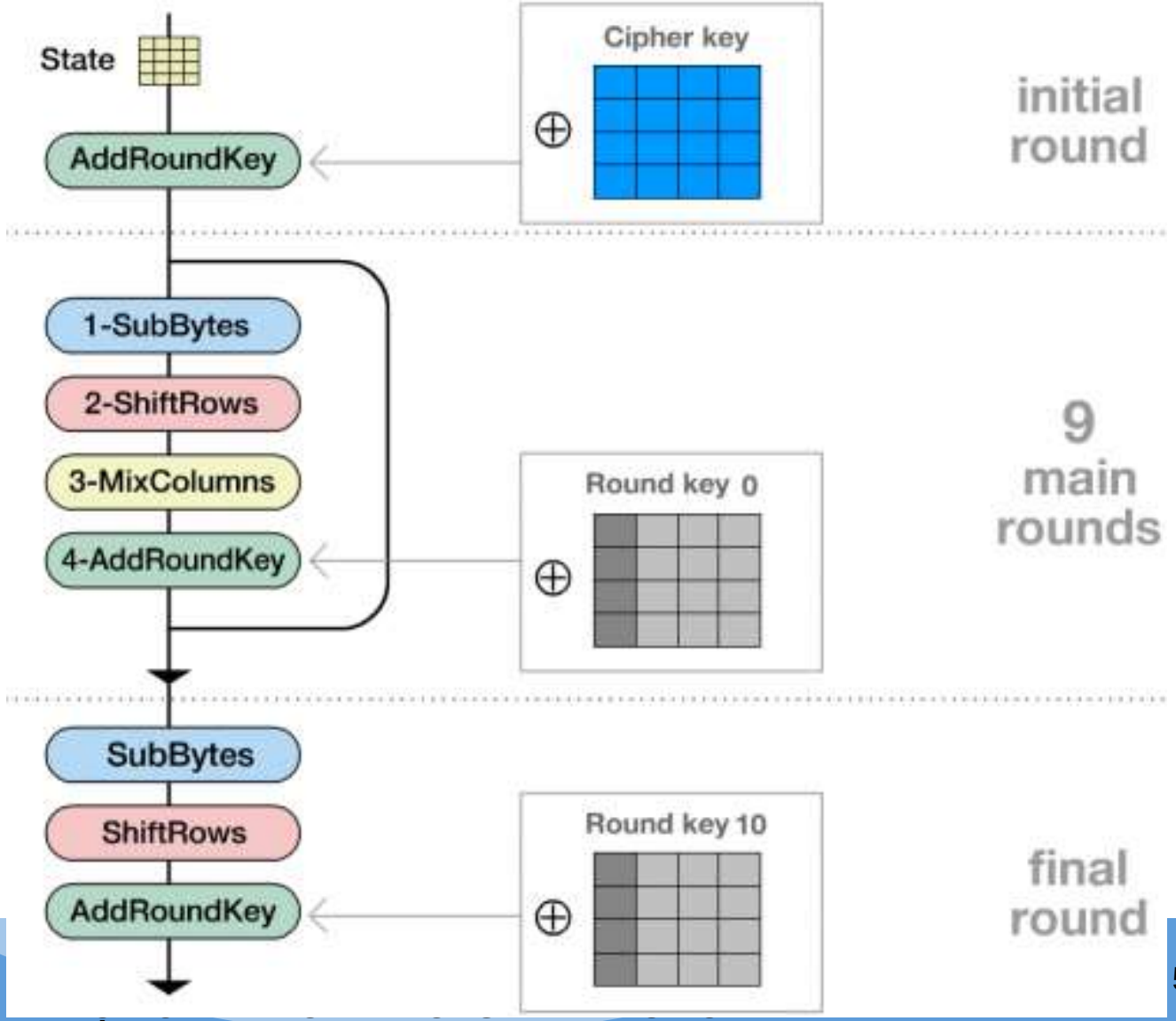
- Mô tả khái quát giải thuật AES:
  1. Mở rộng khóa (KeyExpansion): các khóa phụ dùng trong các vòng lặp được sinh ra từ khóa chính AES sử dụng thủ tục sinh khóa Rijndael.
  2. Vòng khởi tạo (InitialRound)
     a) AddRoundKey: Mỗi byte trong *state* được kết hợp với khóa phụ sử dụng XOR

# 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.

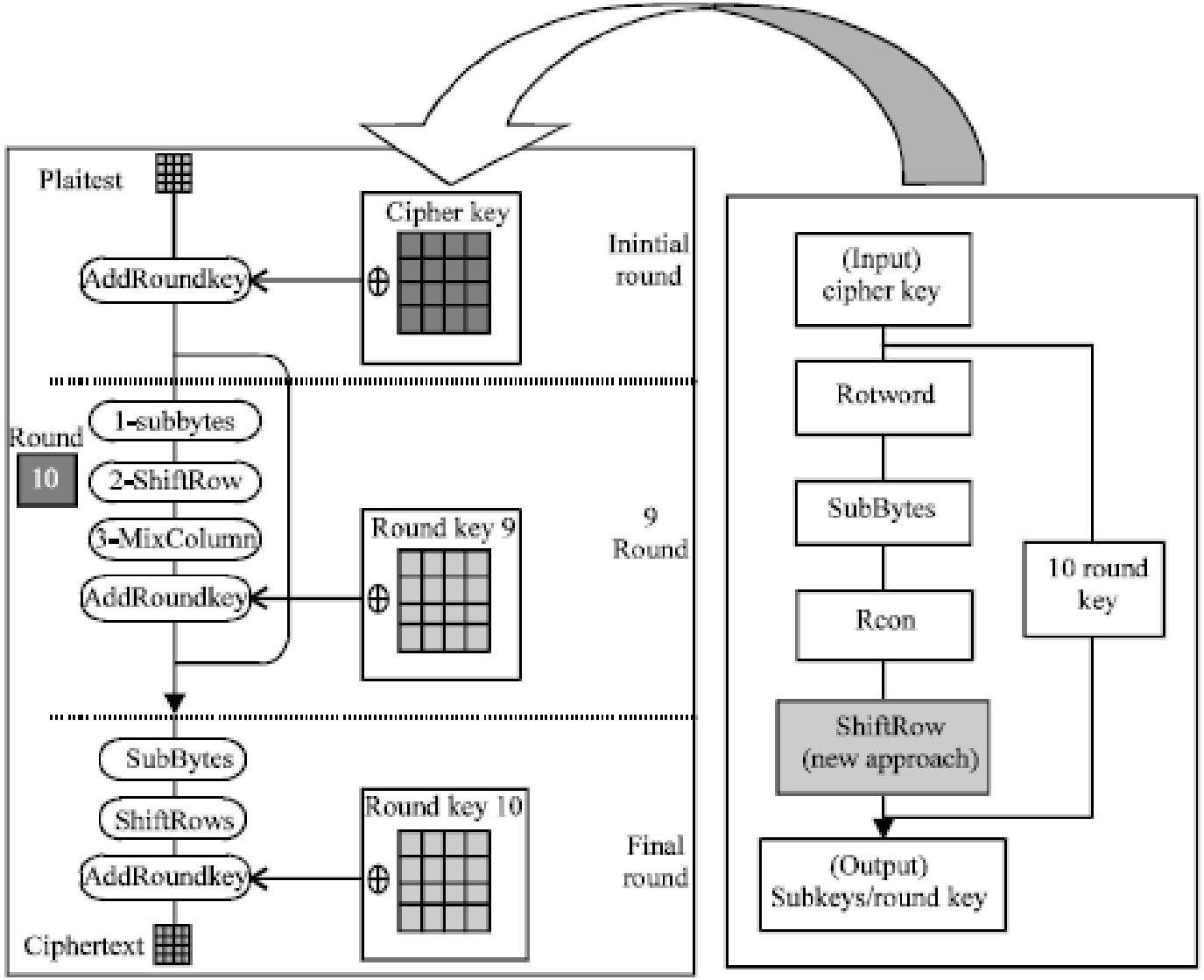
# 4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

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

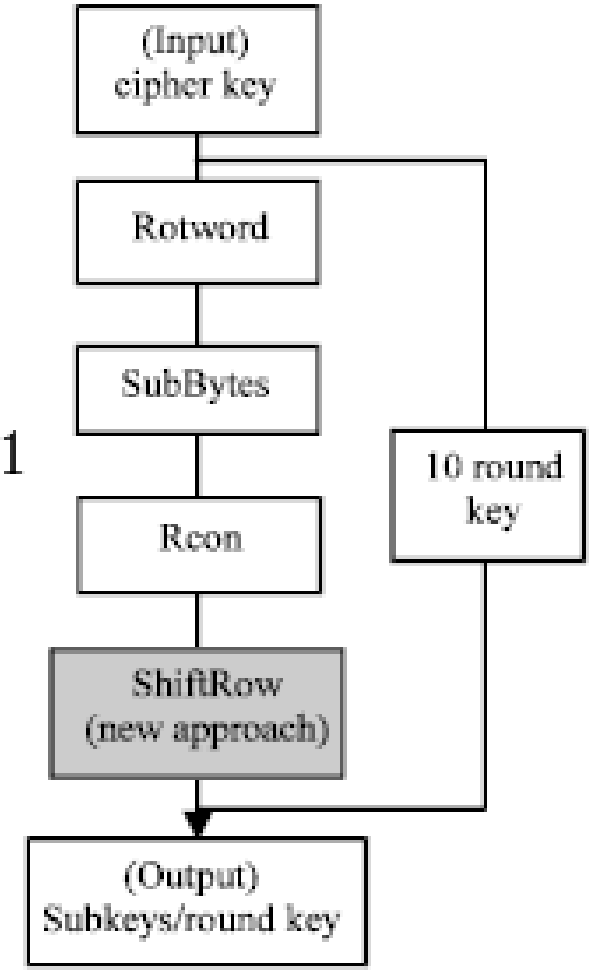


# 4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

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

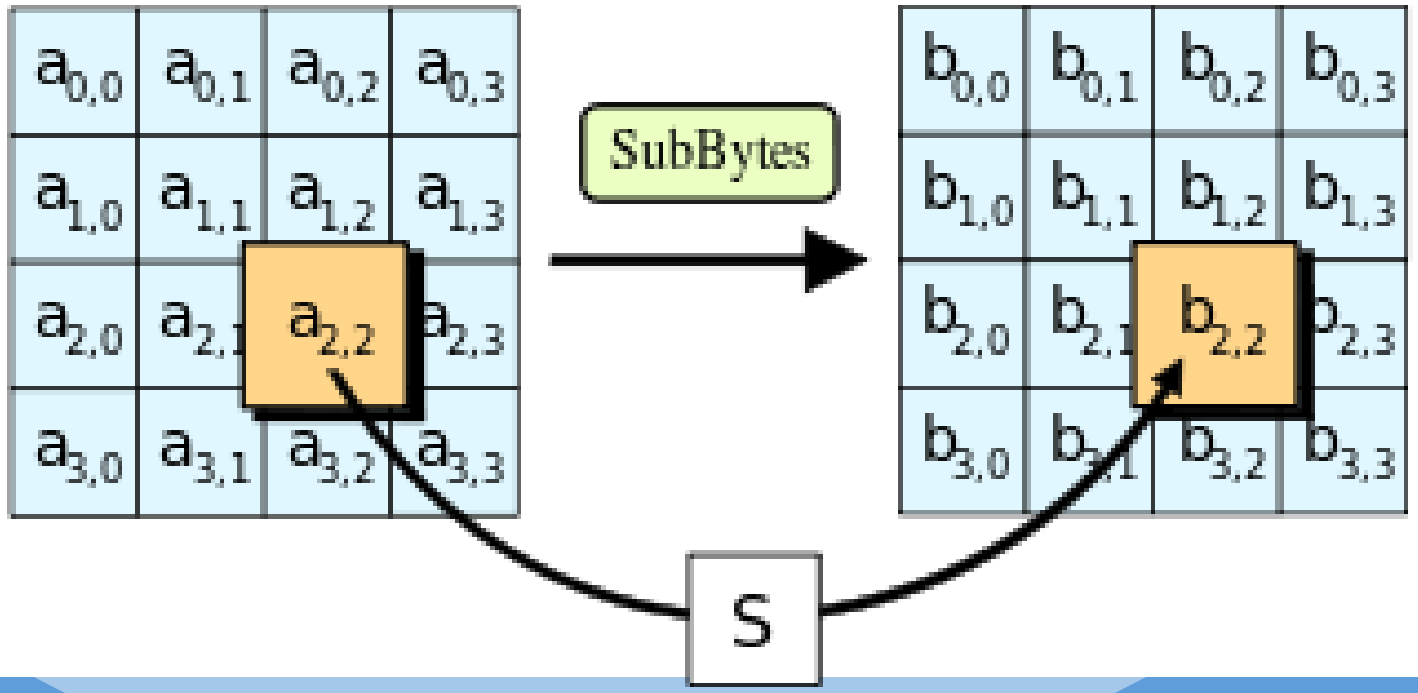
$$\text{rcon}(i) = x^{(i-1)} \mod x^8 + x^4 + x^3 + x + 1$$

  - ShiftRow



# 4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

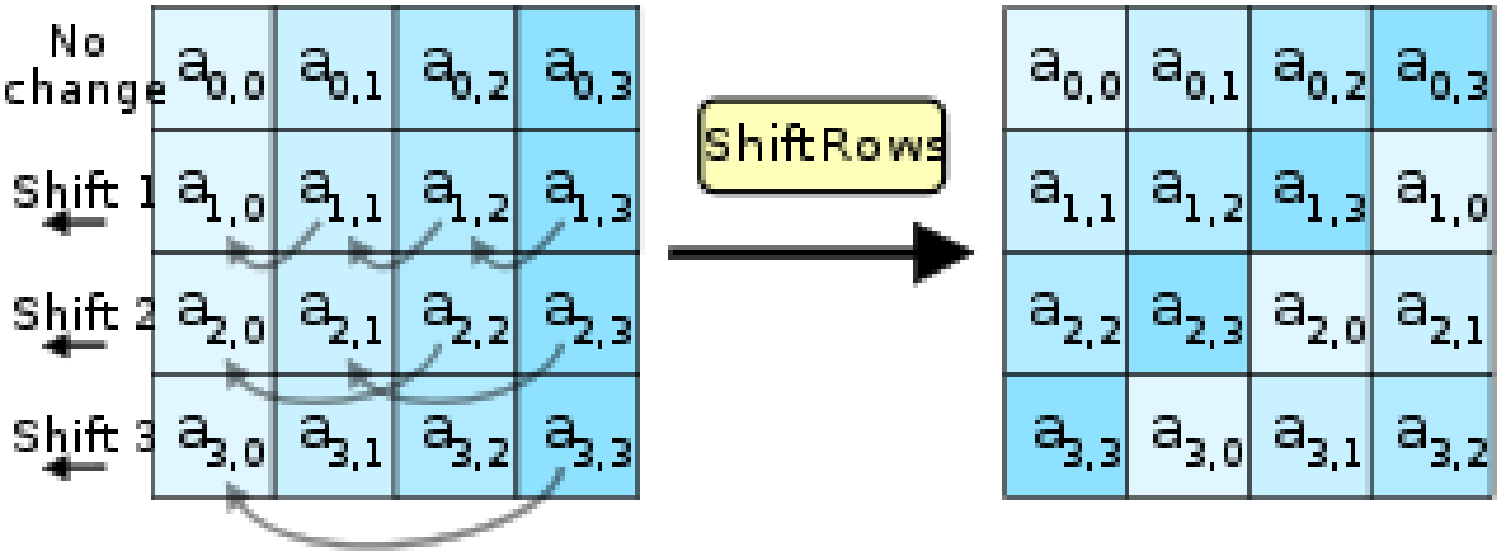
- Bước SubBytes:
  - Mỗi byte trong ma trận state được thay thế bởi 1 byte trong Rijndael S-box, hay $b_{ij} = S(a_{ij})$



# 4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

## ❖ Bước ShiftRows:

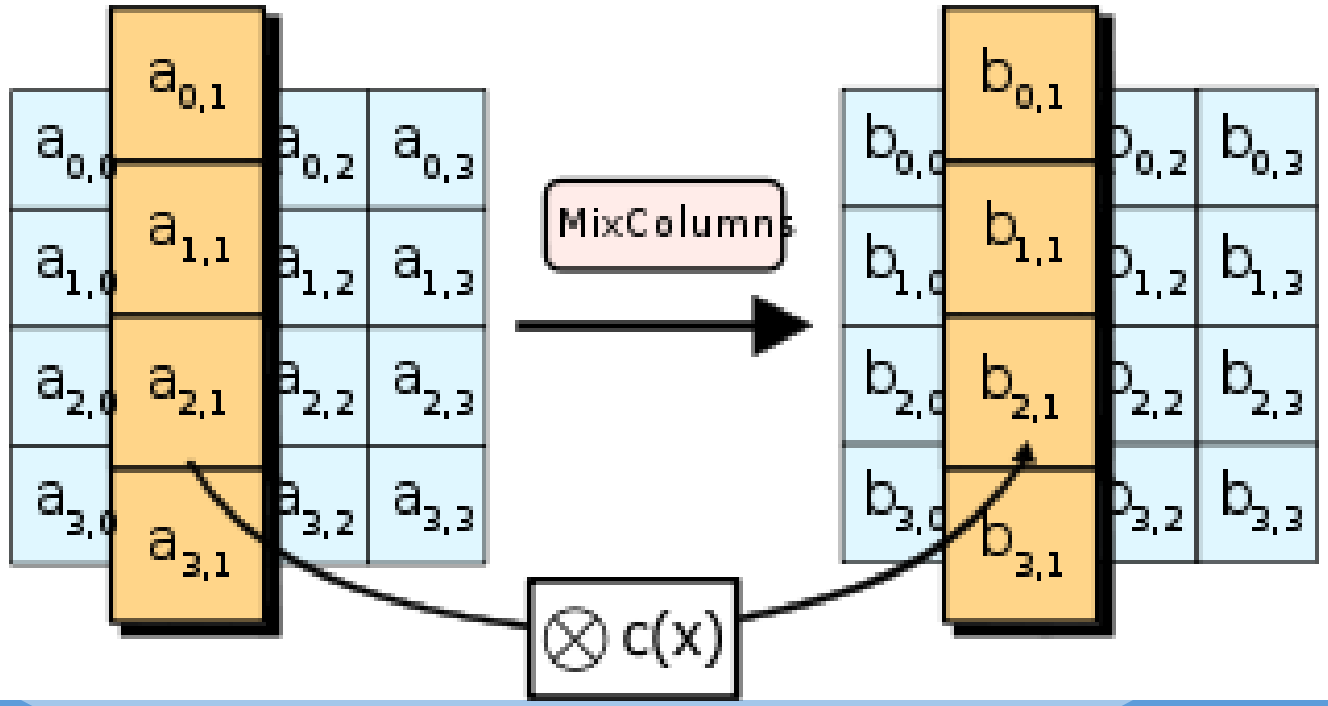
- Các dòng của ma trận state được dịch theo chu kỳ sang trái;
- Dòng thứ nhất giữ nguyên.



# 4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

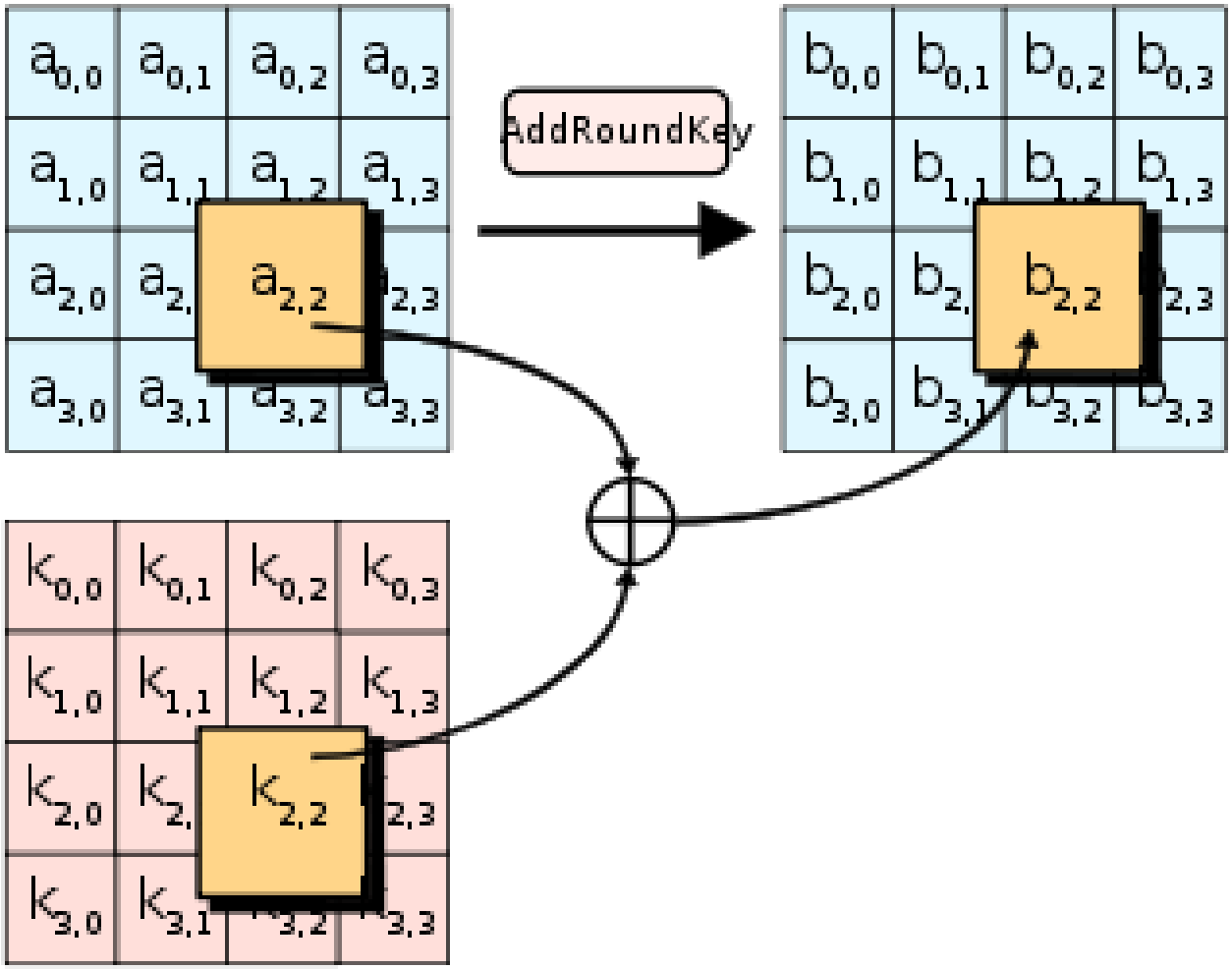
## ❖ Bước MixColumns:

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



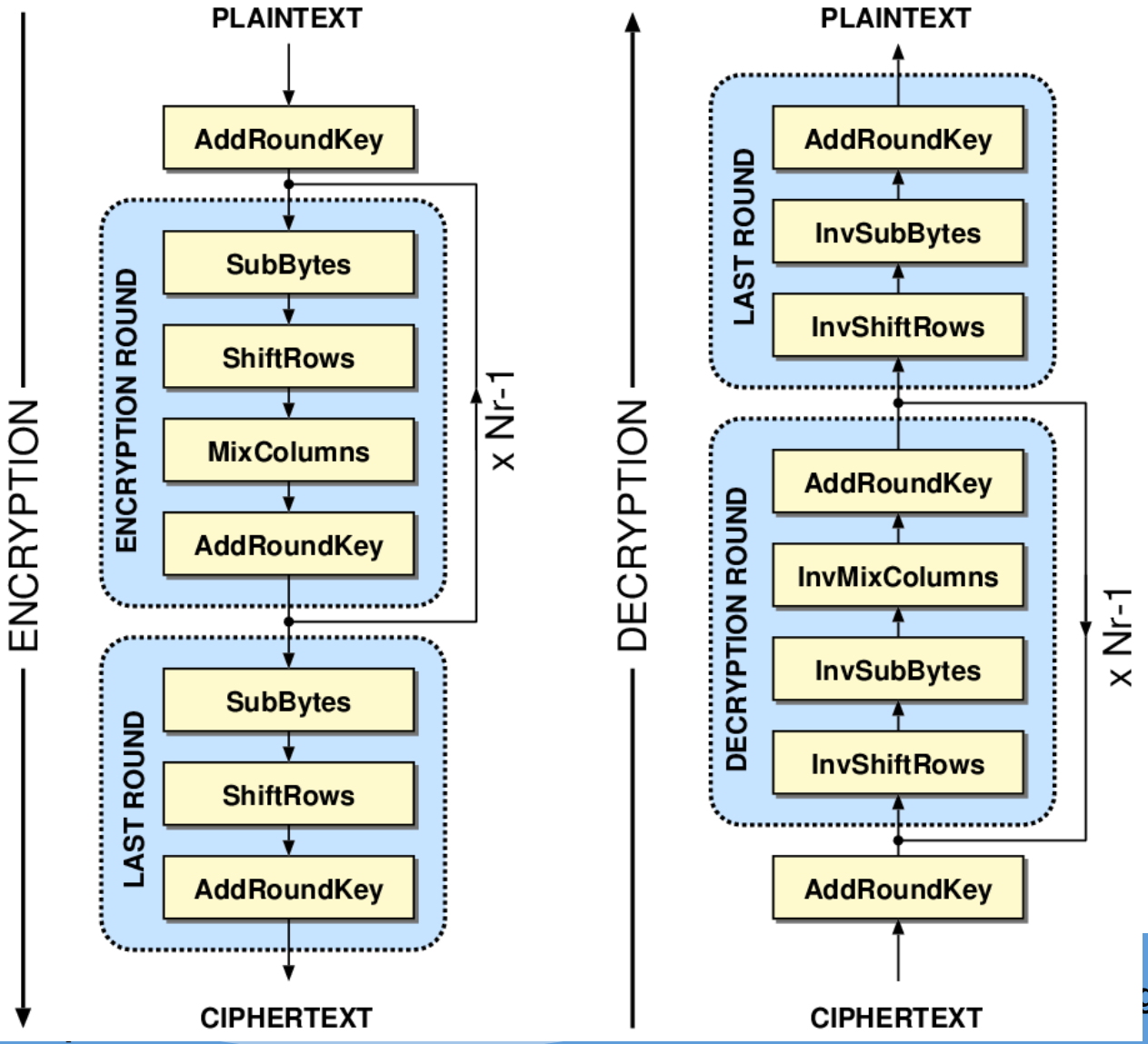
# 4.3.1 Các giải thuật mã hóa khóa đối xứng – AES

- 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



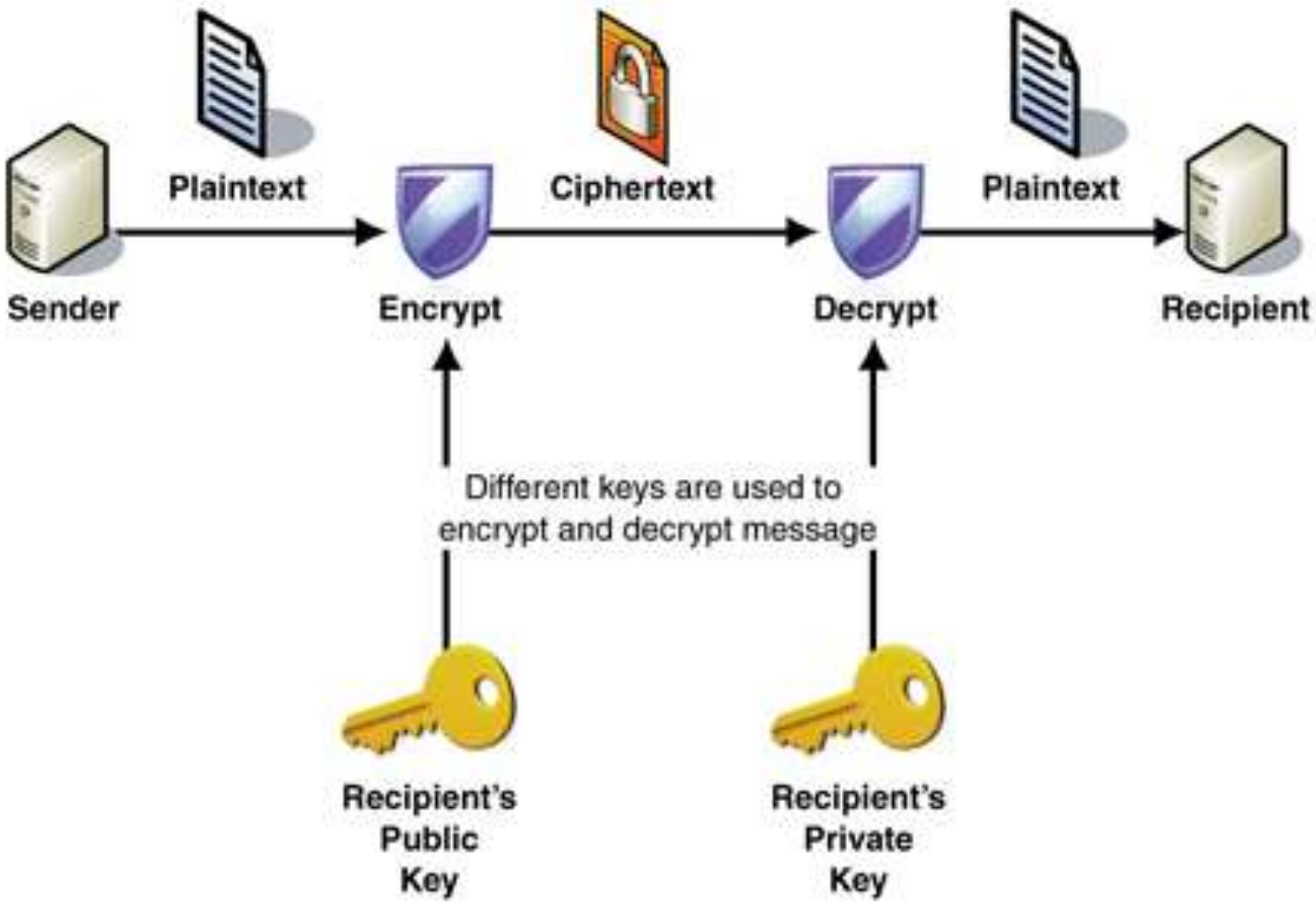
## 4.3.2 Các giải thuật mã hóa khóa bất đối xứng

- 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 cho mã hóa và một khóa cho giải mã.
- Đặc điểm:
  - Kích thước khóa lớn (1024 – 3072 bít)
  - Tốc độ chậm
  - Độ an toàn cao
  - Thuận lợi trong quản lý và phân phối khóa.

# 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



# 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 (số có hàng trăm chữ số thập phân);
- 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. 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 e sao cho $0 < e < \Phi(n)$ và $\gcd(e, \Phi(n)) = 1$
  - 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,
    - $c = m^e \bmod n = 6^7 \bmod 33 = 279936 \% 33 = 30$
    - → c = 30
  - Giải mã:
    - $m = c^d \bmod n = 30^3 \bmod 33 = 27000 \% 33 = 6$
    - → 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 lẻ 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 $\Phi$(n).
  - Nếu p và q được chọn ngẫu nhiên thì 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 150 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 thao tác 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;
  - 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 ;
```

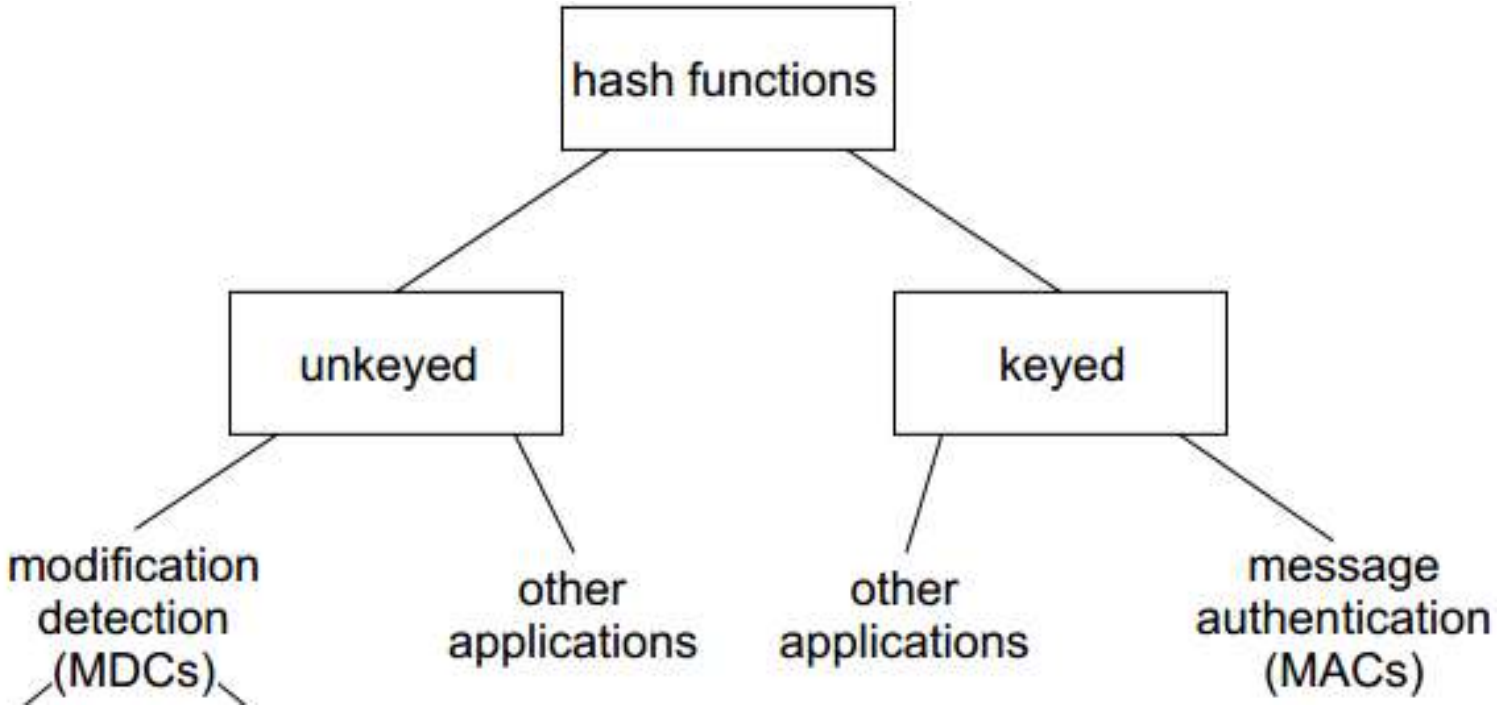
## 4.3.3 Các hàm băm

- Hàm băm (hash function) là một hàm toán học h có tối thiểu 2 thuộc tính:
  - Nén (compression): h là một ánh xạ từ chuỗi đầu vào x có chiều dài bất kỳ sang một chuỗi đầu ra h(x) chó 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

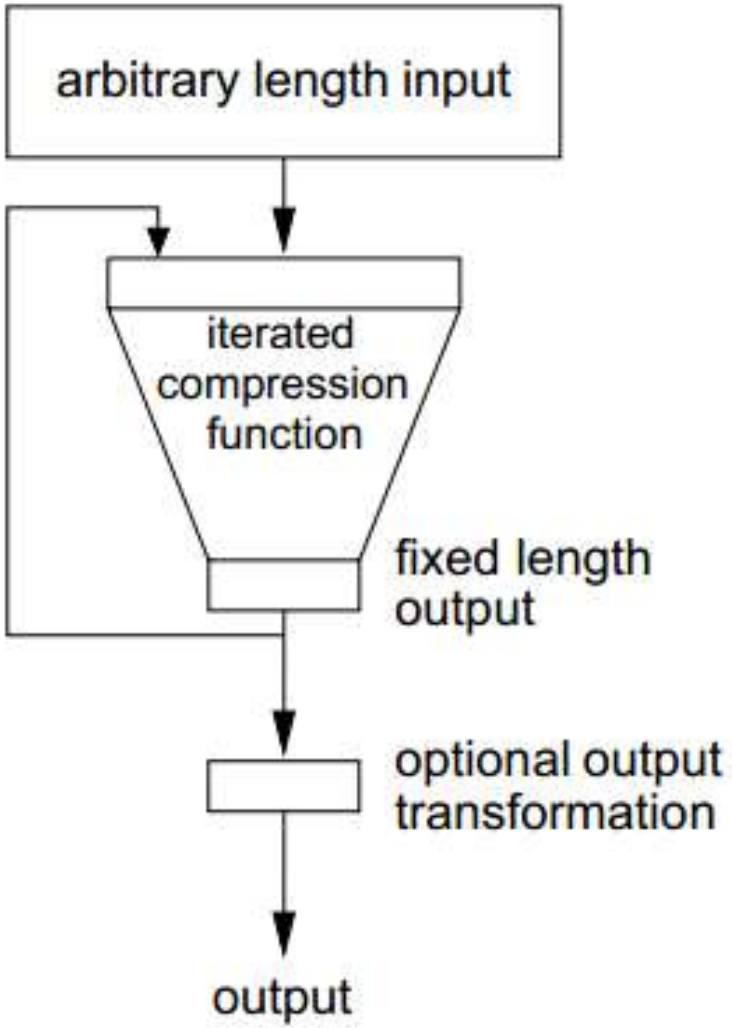
- Phân loại hàm băm theo tính năng:
  - Mã phát hiện sửa đổi (MDC - Modification detection codes)
    - MDC thường được sử dụng để tạo chuỗi đại diện cho thông điệp và dùng kết hợp với các biện pháp khác để đảm bảo tính toàn vẹn của thông điệp;
    - MDC thuộc loại hàm băm không khóa;
    - Hai loại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

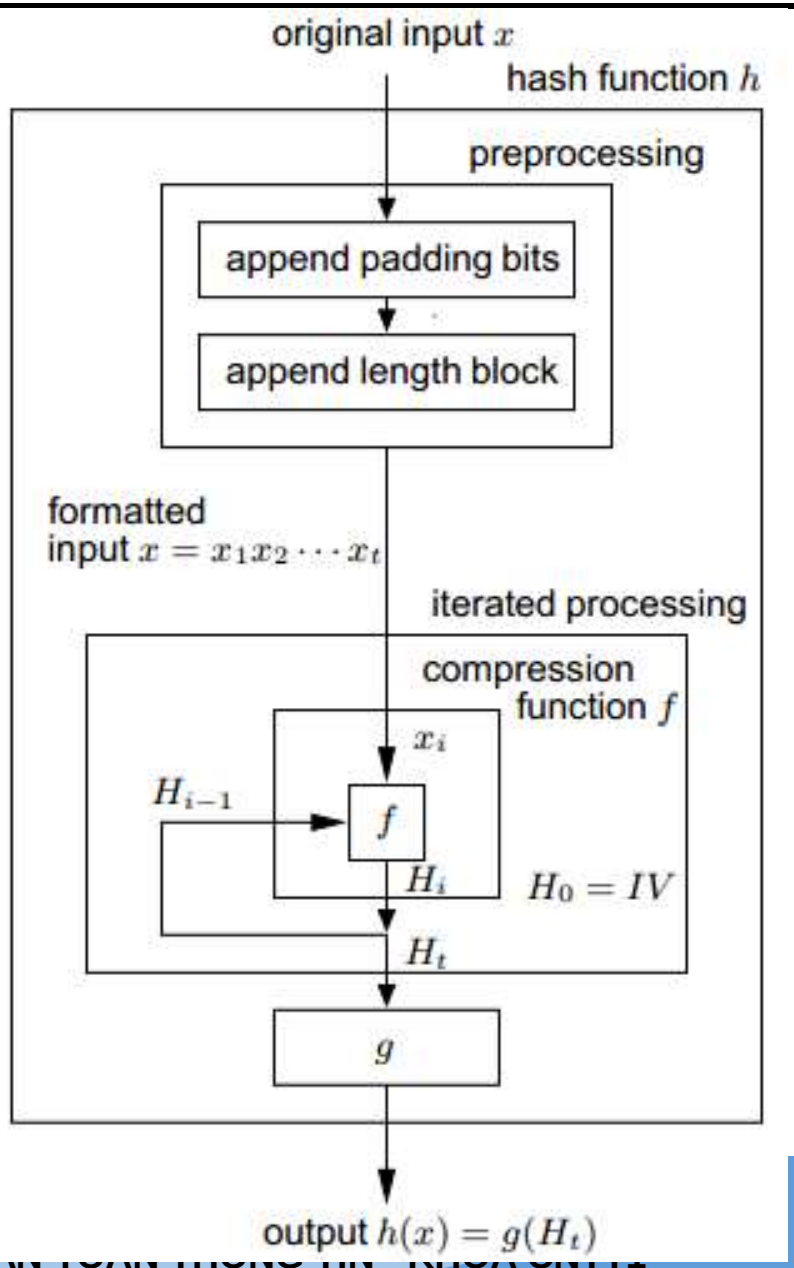
# 4.3.3 Các hàm băm

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

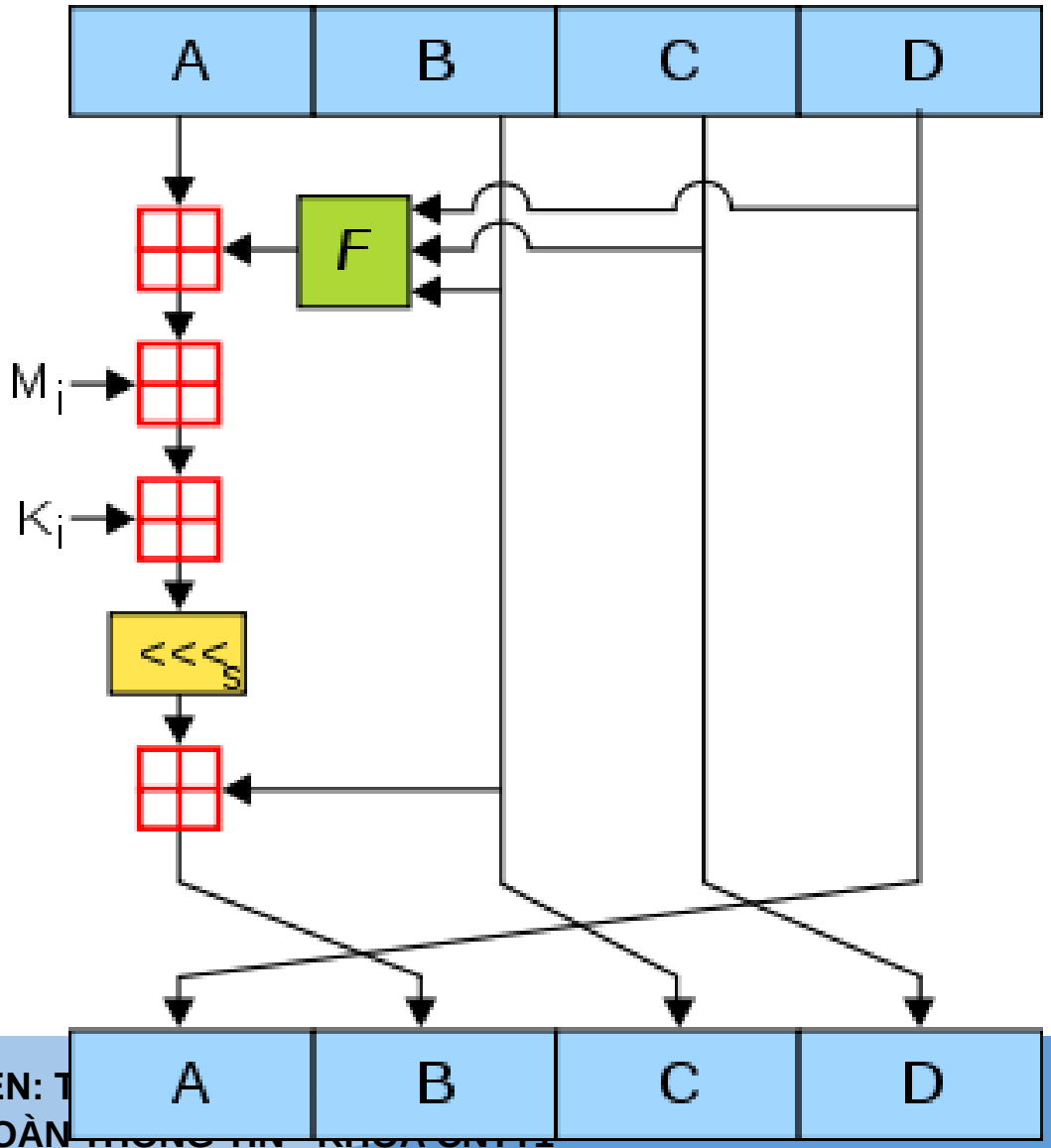
- Lưu đồ xử lý một thao tác của MD5:
  - A, B, C, D: các từ 32 bit
  - Mi: khối 32 bit thông điệp đầu vào;
  - Ki: 32 bit hằng. Mỗi sử dụng một hằng khác nhau;
  - <<<s: thao tác dịch trái s bit
  - ⊞ biểu diễn cộng modulo 32 bít;
  - F: hàm phi tuyến tính, gồm 4 loại:

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

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

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

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



# 4.3.3 Các hàm băm – SHA1

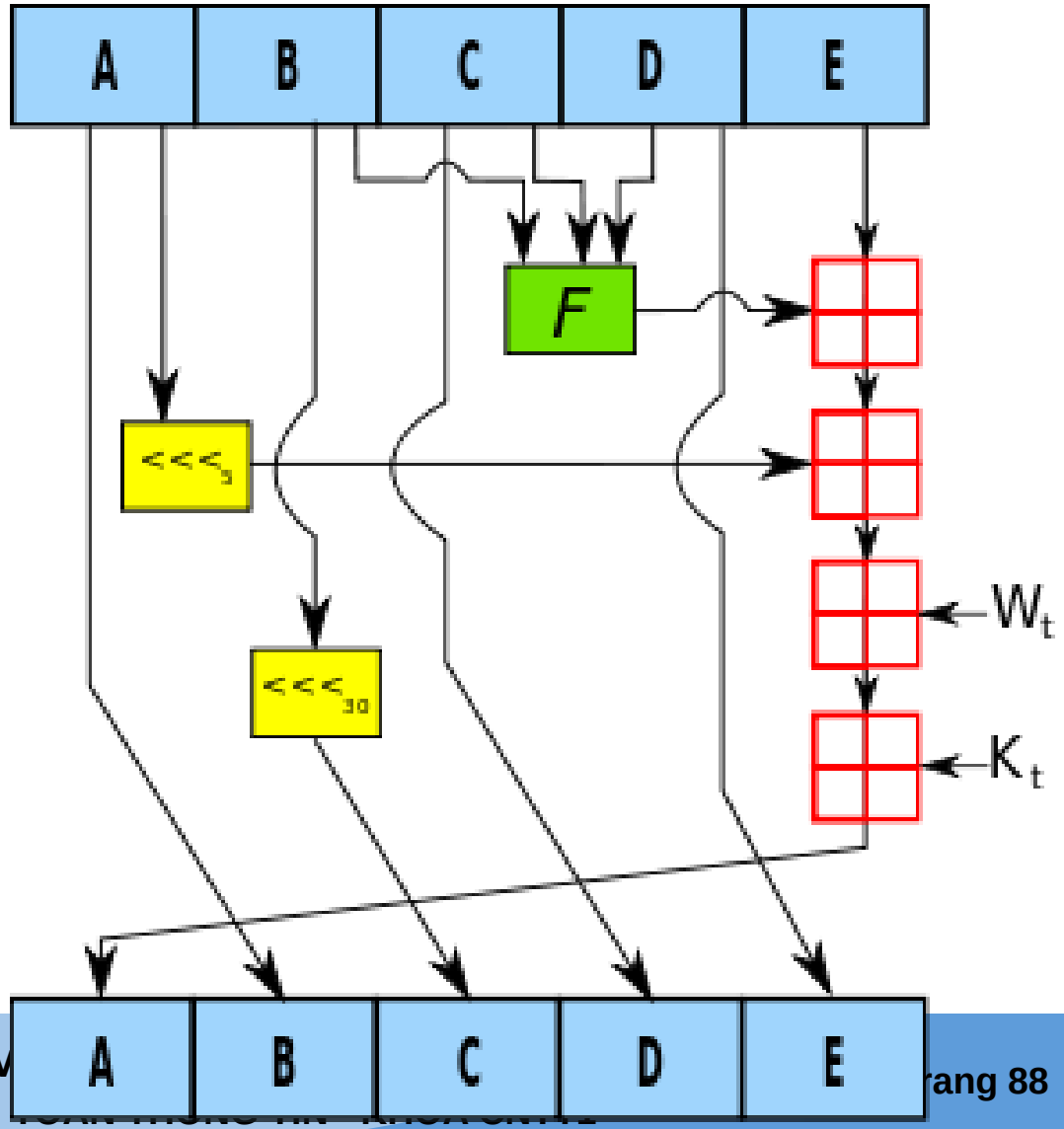
- SHA1 (Secure Hash Function) được NSA (Mỹ) thiết kế năm 1995 để thay thế cho SHA0;
- Chuỗi đầu ra của SHA1 có kích thước 160 bít và thương được biểu diễn thành 40 số hexa;
- Họ hàm băm SHA: SHA-0, SHA-1, SHA-2, SHA-3:
  - SHA0 ít được sử dụng trên thực tế;
  - SHA1 tương tự SHA0, nhưng đã khắc phục 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.
- SHA1 được sử dụng rộng rãi để đảm bảo tính xác thực và toàn vẹn thông điệp.

# 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 Quản lý khóa và phân phối khóa

- Một số khái niệm
- Các kỹ thuật phân phối khóa bí mật
- Các kỹ thuật phân phối khóa công khai
- Các giao thức phân phối và thỏa thuận khóa

# 4.4 QL và phân phối khóa – Khái niệm

- Quan hệ khóa (Keying relationship): là trạng thái mà trong đó các bên tham gia truyền thông chia sẻ dữ liệu chia sẻ (thường là khóa hoặc thành phần tạo ra khóa) để sử dụng cho các kỹ thuật mã hóa;
  - Các dữ liệu chia sẻ có thể là:
    - Khóa bí mật
    - Khóa công khai
    - Các giá trị khởi tạo
    - Các tham số bổ sung không bí mật.
- Quản lý khóa (Key management) là một tập các kỹ thuật cho phép thiết lập và duy trì các quan hệ khóa giữ các bên có thẩm quyền.

# 4.4 QL và phân phối khóa – Khái niệm

- Cụ thể, quản lý khóa gồm các kỹ thuật và thủ tục cho phép:
  - Khởi tạo các người dụng hệ thống (system users) trong một vùng (domain);
  - Sinh khóa, phân phối và cài đặt các dữ liệu khóa;
  - Kiểm soát việc sử dụng các dữ liệu khóa;
  - Cập nhật, thu hồi và hủy các dữ liệu khóa;
  - Lưu, sao lưu/khôi phục và lưu trữ các dữ liệu khóa.

# 4.4 QL và phân phối khóa – Khái niệm

❖ Phân loại khóa theo mục đích sử dụng:

| Term | Meaning |
|---|---|
| private key, public key | paired keys in an asymmetric cryptographic system |
| symmetric key | key in a symmetric (single-key) cryptographic system |
| secret | adjective used to describe private or symmetric key |

| ↓ Cryptographic objective (usage) | Algorithm type | |
|---|---|---|
| | public-key | symmetric-key |
| confidentiality† | encryption | encryption |
| data origin authentication‡ | signature | MAC |
| key agreement | Diffie-Hellman | various methods |
| entity authentication (by challenge-response protocols) | 1. signature<br>2. decryption<br>3. customized | 1. MAC<br>2. encryption |

# 4.4 QL và phân phối khóa – Khái niệm

- Quản lý khóa đóng vai trò quan trọng trong việc cung cấp các tính năng:
  - Tính bí mật
  - Toàn vẹn
  - Xác thực
  - Không thể chối bỏ
  - Chữ ký số.
- Quản lý khóa phù hợp sẽ đảm bảo cho các thông tin khóa được an toàn, đặc biệt khi có nhiều thực thể tham gia truyền thông.
  - Thông tin khóa an toàn → đảm bảo tính an toàn của hệ mã hóa.

# 4.4 QL và phân phối khóa – Khái niệm

- Các mối đe dọa đối với quản lý khóa:
  - Các khóa bí mật bị lộ;
  - Tính xác thực của các khóa bí mật và công khai bị thỏa hiệp (compromise). Tính xác thực bao gồm các hiểu biết và việc kiểm chứng thông tin nhận dạng của một bên mà khóa được chia sẻ;
  - Sử dụng trái phép các khóa bí mật và công khai:
    - Sử dụng các khóa đã hết hiệu lực;
    - Sử dụng các khóa sai mục đích.

# 4.4 QL và phân phối khóa – Khái niệm

- Chính sách an ninh và vấn đề quản lý khóa.
  - Quản lý khóa luôn được thực hiện trong khuôn khổ chính sách an ninh cụ thể;
- Chính sách an ninh mô tả các mục về quản lý khóa:
  - Các thực tế và thủ tục cần thực hiện trong các khía cạnh kỹ thuật và quản trị khóa tự động hoặc bằng tay;
  - Trách nhiệm của các bên có liên quan;
  - Các bản ghi dữ liệu cần phải lưu để tạo các báo cáo về các vấn đề có liên quan đến an toàn khóa.

# 4.4 QL và phân phối khóa – Khái niệm

- Các mô hình thiết lập khóa (Key establishment) đơn giản:
  - Vấn đề phân phối $n^2$ khóa:
    - Nếu một hệ thống có n người dùng tham gia truyền thông sử dụng kỹ thuật mã hóa khóa đối xứng và mỗi cặp người dùng cần trao đổi thông tin an toàn:
      - Mỗi cặp người dùng cần chia sẻ một khóa bí mật duy nhất;
      - Mỗi người dùng cần sở hữu n-1 khóa bí mật;
      - Tổng số khóa cần quản lý trong hệ thống là $n(n-1)/2 \approx n^2$;
      - Số khóa cần quản lý sẽ rất lớn nếu số người dùng lớn.
    - Có thể sử dụng máy chủ trung tâm để quản lý và phân phối khóa.

# 4.4 QL và phân phối khóa – Khái niệm

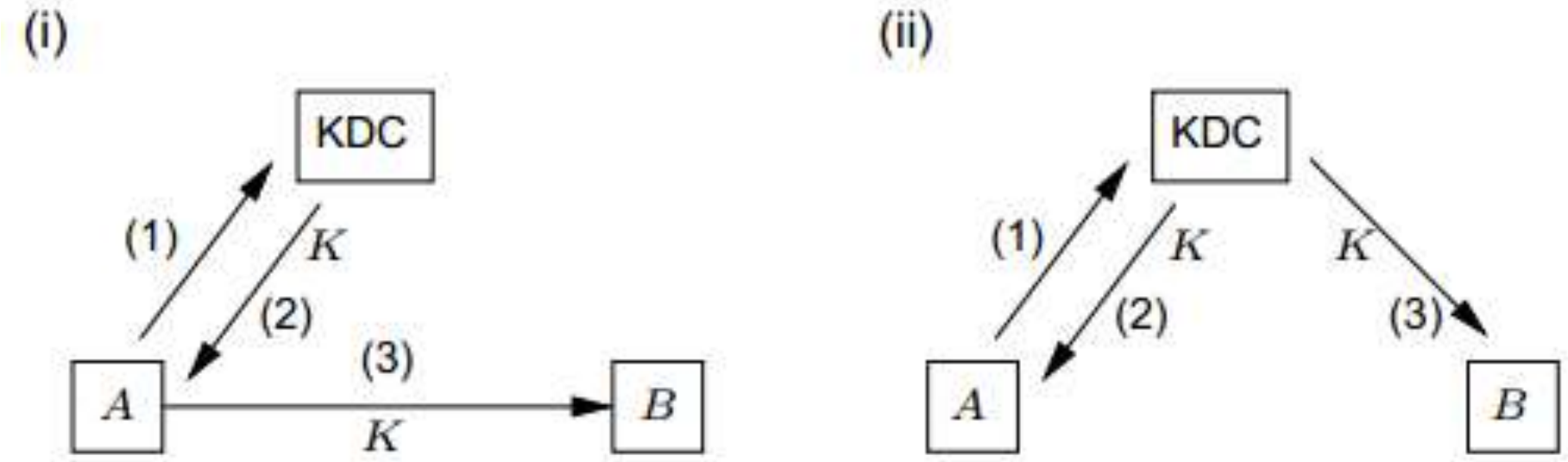
- Các mô hình thiết lập khóa (Key establishment) đơn giản:
  - Phân phối khóa điểm – điểm (Point-to-point key distribution)
  - Trung tâm phân phối khóa (Key distribution center – KDC)
  - Trung tâm dịch hóa (Key translation center – KTC)

# 4.4 QL và phân phối khóa – Khái niệm

- Phân phối khóa điểm – điểm (Point-to-point key distribution):
  - Việc phân phối khóa chỉ liên quan trực tiếp đến 2 thực thể tham gia truyền thông.

# 4.4 QL và phân phối khóa – Khái niệm

- Trung tâm phân phối khóa (Key distribution center – KDC)

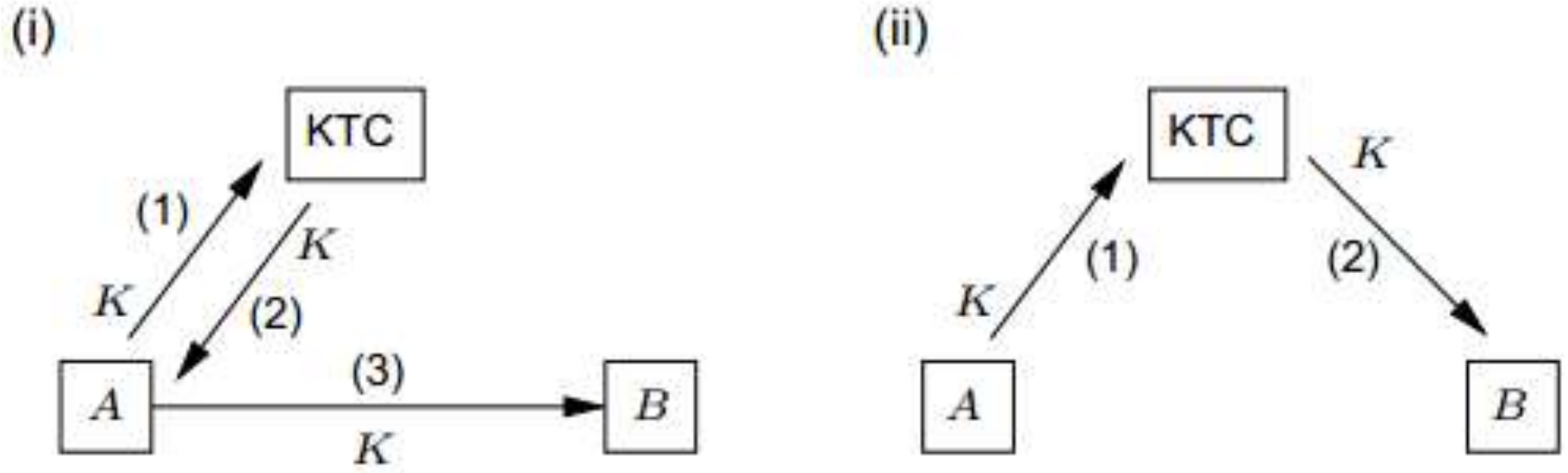


# 4.4 QL và phân phối khóa – Khái niệm

- Trung tâm phân phối khóa (Key distribution center – KDC)
  - KDC được sử dụng để phân phối khóa;
  - Người dùng chia sẻ khóa với KDC, nhưng không chia sẻ khóa với nhau.
- Thủ tục phân phối khóa:
  - A yêu cầu chia sẻ khóa với B;
  - Trung tâm phân phối khóa T sẽ tạo ra hoặc lấy khóa có sẵn K và gửi khóa đã mã hóa dưới dạng $K_{AT}$ cho A;
  - T cũng có thể gửi khóa cho B dưới dạng $K_{BT}$ thông qua A (hình i);
  - T cũng có thể gửi khóa trực tiếp cho B dưới dạng $K_{BT}$ (hình ii).

# 4.4 QL và phân phối khóa – Khái niệm

- Trung tâm dịch chuyển khóa (Key translation center – KTC)



# 4.4 QL và phân phối khóa – Khái niệm

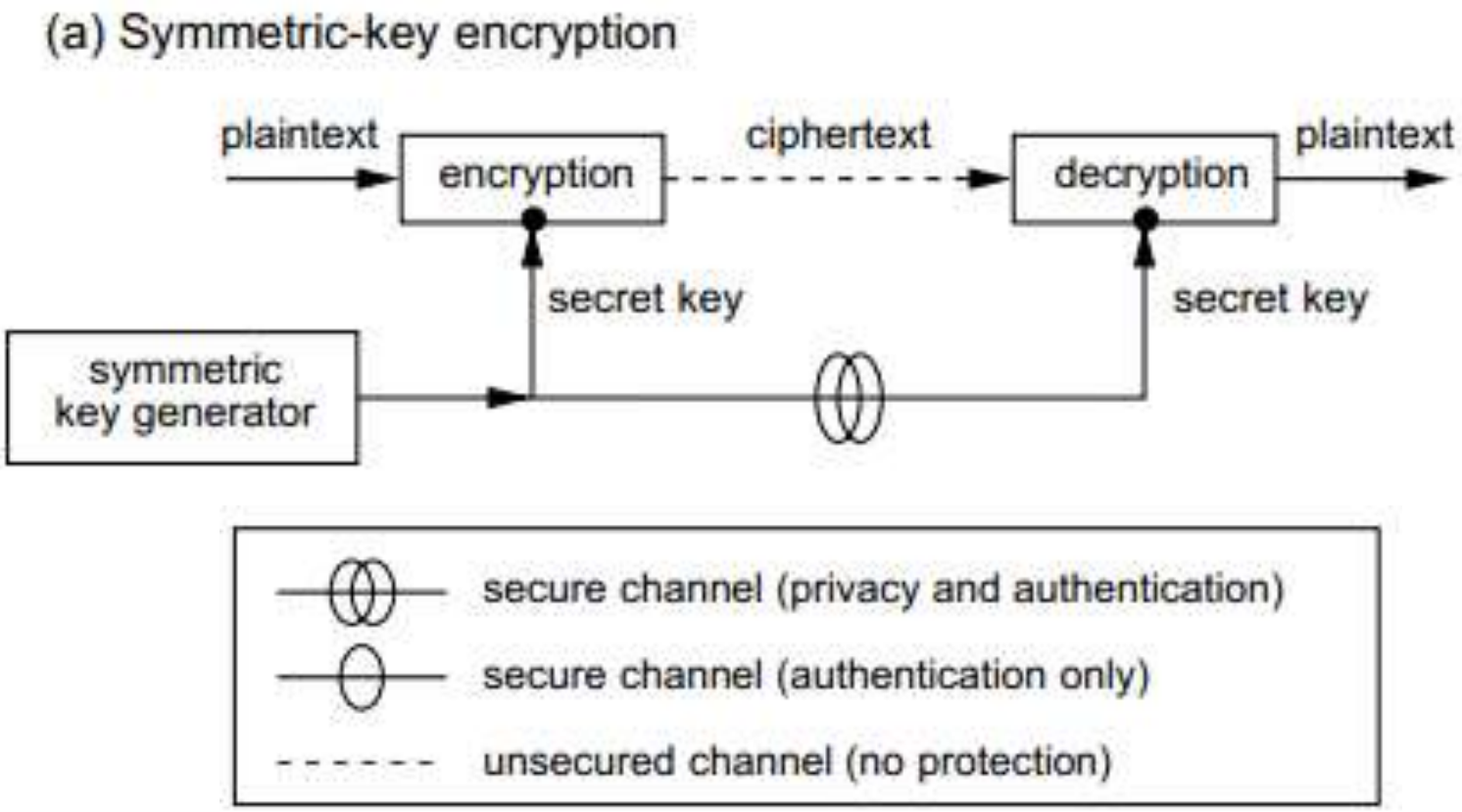
- Trung tâm dịch chuyển khóa (Key translation center – KTC)
  - Vai trò của KTC tương tự KDC;
  - Tuy nhiên, một bên tham gia truyền thông sẽ cung cấp khóa phiên (session key).
- Thủ tục thực hiện:
  - A gửi khóa K đến trung tâm T dưới dạng mã hóa $K_{AT}$;
  - Trung tâm T giải mã lấy khóa K;
  - T mã hóa lại khóa K dưới dạng $K_{BT}$;
  - $K_{BT}$ có thể được gửi cho B thông qua A hoặc gửi trực tiếp đến B.

# 4.4 QL và phân phối khóa – Khái niệm

- So sánh KDC và KTC:
  - KDC cho phép sinh khóa tập trung;
  - KTC cho phép sinh khóa phân tán;
  - Cả KDC và KTC yêu cầu có một máy chủ tin cậy (trusted server).
- Ưu nhược điểm của quản lý khóa tập trung (KDC+KTC)
  - Hiệu quả trong lưu trữ khóa: mỗi bên chỉ cần duy trì một khóa bí mật dài hạn với bên tin cậy (không phải với bên trao đổi thông tin);
  - Cả hệ thống có thể bị mất an toàn nếu trung tâm quản lý khóa bị thỏa hiệp (điều khiển);
  - Trung tâm quản lý khóa có thể thành điểm nút cổ chai;
  - Dịch vụ sẽ phải ngừng nếu trung tâm quản lý khóa gặp trục trặc;
  - Cần có một máy chủ tin cậy ở chế độ trực tuyến.

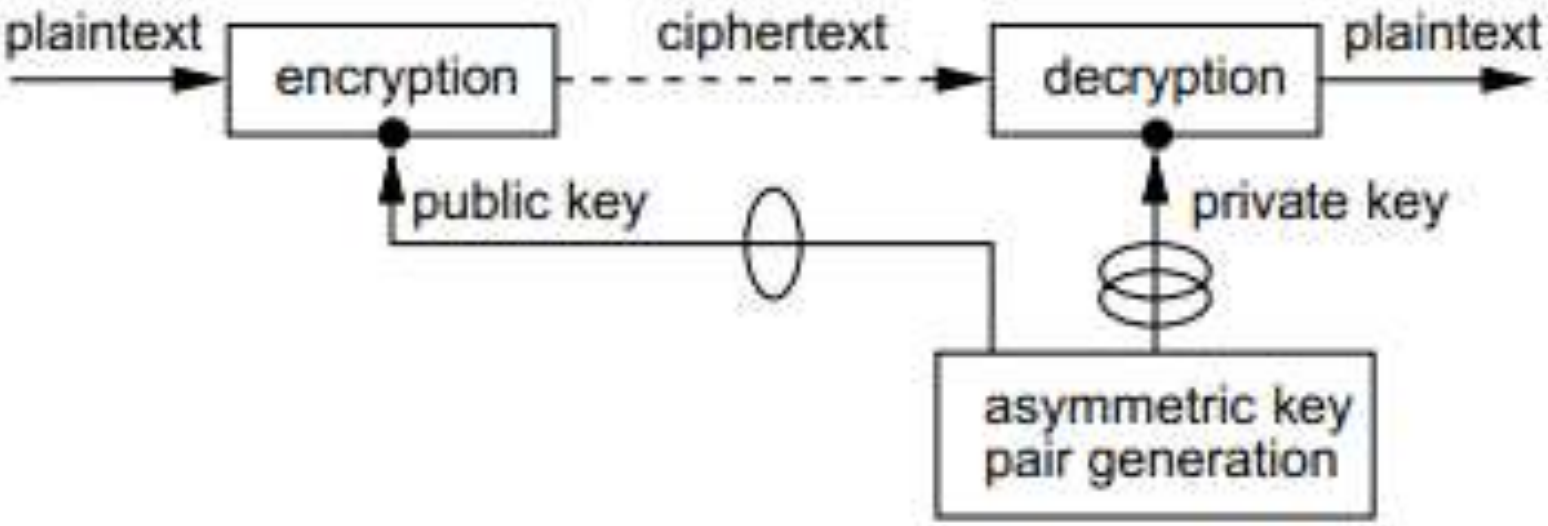
# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

❖ Mô hình tạo và sử dụng khóa – Hệ mã hóa khóa bí mật



# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

❖ Mô hình tạo và sử dụng khóa – Hệ mã hóa khóa công khai



# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

- Phân loại các lớp khóa theo khả năng sử dụng:
  - Khóa chủ (Master key):
    - Là các khóa ở mức cao nhất và không được bảo vệ bằng các kỹ thuật mật mã.
    - Các khóa chu thường được chuyển giao trực tiếp và được bảo vệ bằng các cơ chế kiểm soát vật lý.
  - Khóa dùng cho trao đổi khóa (Key – encrypting keys):
    - Là những khóa được sử dụng để vận chuyển hoặc lưu trữ các khóa khác.
    - Các khóa này cũng có thể được bảo vệ bằng khóa khác.
  - Khóa dữ liệu (Data keys):
    - Là các khóa được sử dụng để mã hóa dữ liệu cho người dùng.
    - Thường là các khóa ngắn hạn.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

- Phân loại các lớp khóa theo thời gian sử dụng:
  - Khóa dài hạn (long-term keys):
    - Là các khóa được sử dụng trong một khoảng thời gian dài;
    - Gồm: khóa chủ, khóa dùng cho trao đổi khóa, hoặc khóa dùng cho thỏa thuận khóa.
  - Khóa ngắn hạn:
    - Là các khóa được sử dụng trong một khoảng thời gian ngắn hoặc chỉ trong một phiên làm việc;
    - Gồm các khóa được trao đổi trong quá trình trao đổi khóa, thỏa thuận khóa, dùng để mã hóa dữ liệu của người dùng.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

- Trung tâm dịch khóa (KTC):
  - A sở hữu khóa dài hạn $K_{AT}$ – chia sẻ với KTC;
  - B sở hữu khóa dài hạn $K_{BT}$ – chia sẻ với KTC;
  - Trung tâm dịch khóa T là một máy chủ tin cậy, cho phép hai bên A và B không trực tiếp chia sẻ thông tin khóa thiết lập kênh truyền thông an toàn sử dụng hai khóa dài hạn $K_{AT}$ và $K_{BT}$.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

- Thuật toán phân phối khóa sử dụng KTC:
  - Tóm tắt: A tương tác với T và B.
  - Kết quả: A gửi được thông điệp bí mật M (có thể là 1 khóa phiên) đến B.
  - Ký hiệu: E là thuật toán mã hóa khóa bí mật, M có thể là khóa phiên K.
  - Khởi tạo 1 lần: A và T chia sẻ khóa $K_{AT}$ và B và T chia sẻ khóa $K_{BT}$.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

- Thuật toán phân phối khóa sử dụng KTC:
  - Các thông điệp trao đổi:

$$A \rightarrow T: \quad A,\ E_{K_{AT}}(B, M) \qquad (1)$$
$$A \leftarrow T: \quad E_{K_{BT}}(M, A) \qquad (2)$$
$$A \rightarrow B: \quad E_{K_{BT}}(M, A) \qquad (3)$$

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

- Thuật toán phân phối khóa sử dụng KTC:
  - Mô tả các bước thực hiện:
    - A mã hóa M và số định danh của B (người nhận) sử dụng khóa $K_{AT}$ và gửi thông điệp kèm theo số định danh của A cho T.
    - T giải mã thông điệp, xác định được người nhận là B. T mã hóa M sử dụng khóa $K_{BT}$ để chuyển cho B.
    - T gửi lại thông điệp đã dịch cho A để chuyển cho B, hoặc T có thể gửi thẳng cho B.
    - B giải mã thông điệp sử dụng khóa $K_{BT}$ để có M.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

- Chứng chỉ khóa đối xứng (Symmetric-key certificates) cung cấp một phương tiện cho phép KTC:
  - Không phải duy trì một CSDL an toàn lưu các khóa bí mật của người dùng (hoặc phải sao chép CSDL này đến nhiều máy chủ);
  - Hoặc không phải yêu cầu các khóa bí mật của người dùng từ một CSDL an toàn.
- Mỗi chứng chỉ khóa đối xứng có một thời hạn sử dụng xác định.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

- Cơ chế sử dụng chứng chỉ khóa đối xứng:
  - Thực thể B sở hữu khóa dài hạn $K_{BT}$ – chia sẻ với KTC T;
  - Khóa dài hạn $K_{BT}$ được lưu trên T nhúng trong chứng chỉ khóa đối xứng $E_{K_T}(K_{BT}, B)$;
  - $K_T$ là khóa chủ của T và chỉ T được biết;
  - Chứng chỉ khóa đối xứng được sử dụng như là bản ghi nhớ cho chính T (chỉ T mới có thể mở chứng chỉ này);
  - Chứng chỉ khóa đối xứng cũng có thể được chuyển cho B khi có yêu cầu truy nhập khóa $K_{BT}$ để dịch thông điệp;
  - Thay vì phải lưu trữ toàn bộ khóa dài hạn của người dùng, với chứng chỉ khóa đối xứng, T chỉ cần lưu an toàn khóa chủ $K_T$ của mình.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

- Thuật toán phân phối khóa sử dụng KTC với chứng chỉ khóa đối xứng và khỏa chủ $K_T$:
  - Các thông điệp trao đổi:

$$A \rightarrow T: \quad SCert_A,\ E_{K_{AT}}(B, M),\ SCert_B \quad (1)$$
$$A \leftarrow T: \quad E_{K_{BT}}(M, A) \quad (2)$$
$$A \rightarrow B: \quad E_{K_{BT}}(M, A) \quad (3)$$

$$SCert_A = E_{K_T}(K_{AT}, A), SCert_B = E_{K_T}(K_{BT}, B)$$

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa bí mật

- Thuật toán phân phối khóa sử dụng KTC với chứng chỉ khóa đối xứng và khỏa chủ $K_T$:
  - Một CSDL công cộng có thể được sử dụng để lưu định danh người dùng và chứng chỉ khóa đối xứng của họ;
  - Các bước thực hiện:
    - A mã hóa M và số định danh của B (người nhận) sử dụng khóa $K_{AT}$ và gửi thông điệp kèm chứng chỉ khóa đối xứng của A và B lấy từ CSDL công cộng cho T.
    - T sử dụng $K_T$ để giải mã chứng chỉ của A và B, lấy được $K_{AT}$ và $K_{BT}$. T sử dụng $K_{AT}$ giải mã thông điệp để có B và M. Đồng thời T kiểm tra định danh của B có trùng với định danh lưu trong chứng chỉ khóa đối xứng của B.
    - T mã hóa M sử dụng khóa $K_{BT}$ để chuyển cho B. T gửi lại thông điệp đã dịch cho A để chuyển cho B, hoặc T có thể gửi thẳng cho B.
    - B giải mã thông điệp sử dụng khóa $K_{BT}$ để có M.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa công khai

- Các kỹ thuật phân phối khóa công khai thường giả thiết các bên tham gia truyền thông sơ hữu khóa công khai có tính xác thực (authentic public keys);
  - Là các khóa công khai được tạo và sử dụng hợp pháp.
- Việc phân phối khóa công khai cần đảm bảo tính xác thực của chủ thể khóa công khai;
- Các phương pháp phân phối khóa công khai:
  - Trao đổi kiểu điểm-điểm thông qua kênh tin cậy;
  - Truy nhập trực tiếp vào danh mục công cộng (public-key registry);
  - Sử dụng một máy chủ trực tuyến tin cậy;
  - Sử dụng một máy chủ không trực tuyến và chứng chỉ;
  - Sử dụng các hệ thống đảm bảo tính xác thực với các tham số công cộng.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa công khai

- Trao đổi khóa công khai kiểu điểm-điểm thông qua kênh tin cậy:
  - Các bên trực tiếp trao đổi khóa công khai với nhau thông qua các kênh tin cậy như thư bảo đảm hoặc các phương tiện chuyển giao đảm bảo khác;
  - Có thể sử dụng với các trao đổi không thường xuyên;
  - Thích hợp với các hệ thống đóng kín hoặc cỡ nhỏ.
  - Nhược điểm:
    - Bất tiện do trễ lớn;
    - Các kênh tin cậy dùng riêng đắt tiền.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa công khai

- Trao đổi khóa công khai thông qua truy nhập trực tiếp vào danh mục công cộng:
  - Một CSDL công cộng tin cậy được thiết lập, bao gồm tên người dùng và khóa công khai tương ứng;
  - CSDL công cộng này có thể được vận hành bởi 1 bên tin cậy;
  - Người dùng có thể truy nhập khóa công khai từ CSDL này;
  - Một phương pháp thực hiện được sử dụng phổ biến là cây xác thực khóa công khai (Tree authentication of public keys).

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa công khai

- Trao đổi khóa công khai thông qua sử dụng một máy chủ trực tuyến tin cậy:
  - Máy chủ trực tuyến tin cậy cung cấp truy nhập đến CSDL công cộng các khóa công khai;
  - Khóa công khai được ký và gửi cho bên yêu cầu;
  - Kênh truyền không đòi hỏi phải bí mật;
  - Bên yêu cầu sử dụng khóa công khai của máy chủ để xác thực chữ ký của máy chủ và qua đó kiểm tra tính xác thực, toàn vẹn của khóa;
  - Nhược điểm:
    - Máy chủ phải luôn trực tuyến;
    - Máy chủ có thể trở thành điểm nút cổ chai.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa công khai

- Trao đổi khóa công khai thông qua sử dụng một máy chủ không trực tuyến và chứng chỉ:
  - Bên A liên hệ với một bên tin cậy (được gọi là Cơ quan chứng thực - Certification Authority (CA)) để đăng ký khóa công khai của mình và nhận được chữ ký xác nhận khóa công khai của CA;
  - CA cấp một chứng chỉ (Certificate) cho khóa công khai của A: chứng chỉ kết hợp khóa công khai của A với thông tin định danh của A;
  - Khi A đã có chứng chỉ khóa công khai (Public key certificate), A có thể gửi khóa công khai cho các bên có liên quan bằng cách gửi chứng chỉ khóa công khai.
  - Chứng chỉ khóa công khai cũng có thể được đưa vào danh mục công cộng và người dùng có thể truy nhập.

# 4.4 QL và phân phối khóa – Kỹ thuật PP khóa công khai

- Trao đổi khóa công khai thông qua sử dụng các hệ thống đảm bảo tính xác thực với các tham số công cộng:
  - Các hệ thống dựa trên định danh (Identity-based systems);
  - Các hệ thống sử dụng các khóa được chứng thực mặc nhiên (implicitly certified keys);
  - Có khả năng phát hiện các sửa đổi với các tham số công cộng.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa

- Vận chuyển khóa dựa trên mã hóa khóa đối xứng
  - Cập nhật khóa kiểu điểm – điểm
  - AKEP - Authenticated Key Exchange Protocol
  - Giao thức không khóa Shamir
  - Kerberos
- Vận chuyển khóa dựa trên mã hóa khóa công khai
  - Vận chuyển khóa dựa trên mã hóa khóa công khai không có chữ ký
  - Vận chuyển khóa dựa trên mã hóa khóa công khai kết hợp với chữ ký.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

- Cập nhật khóa kiểu điểm – điểm (Point-to-point key update):
  - Phương pháp này cần sử dụng một khóa chia sẻ dài hạn K giữa 2 thực thể (A, B) cần trao đổi khóa ngắn hạn;
  - Khóa dài hạn K được trao đổi bằng một kênh an toàn;
  - Các ký hiệu sử dụng:
    - $r_A$ là một số ngẫu nhiên;
    - $t_A$ là tem thời gian;
    - $n_A$ là số thứ tự (sequence number) do thực thể A tạo ra;
    - E là thuật toán mã hóa khóa đối xứng.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Cập nhật khóa kiểu điểm – điểm:

- Vận chuyển khóa đơn giản (cần 1 lần gửi):
  - A → B: $E_K(r_A)$:
    - A tạo một số ngẫu nhiên $r_A$ (chính là khóa phiên W cần trao đổi), mã hóa bằng khóa dài hạn K và gửi cho B;
    - B nhận được $E_K(r_A)$, giải mã sử dụng khóa K để có được $r_A$.
  - A → B: $E_K(r_A, t_A, B)$:
    - A tạo một số ngẫu nhiên $r_A$, mã hóa bộ kết hợp $(r_A, t_A, B)$ bằng khóa dài hạn K và gửi cho B;
    - B giải mã thông điệp sử dụng khóa dài hạn K để có được $r_A$. B kiểm tra tem thời gian và định danh B trong thông điệp để đảm bảo thông điệp là hợp lệ và nó được gửi cho B.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Cập nhật khóa kiểu điểm – điểm:

- Vận chuyển khóa sử dụng giao thức thách thức – trả lời (challenge – response):

$$A \leftarrow B : n_B \quad (1)$$
$$A \rightarrow B : E_K(r_A, n_B, B^*) \quad (2)$$

$$A \leftarrow B : n_B \quad (1)$$
$$A \rightarrow B : E_K(r_A, n_A, n_B, B^*) \quad (2)$$
$$A \leftarrow B : E_K(r_B, n_B, n_A, A^*) \quad (3)$$

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức vận chuyển khóa AKEP2:

- Tóm tắt: A và B trao đổi 3 thông điệp để chuyển khóa phiên W;
- Kết quả: Các thực thể được xác thực và khóa phiên W được xác thực ngầm định.
- Khởi tạo:
  - A và B đã chia sẻ 2 khóa dài hạn K và K'; K và K' nên khác nhau nhưng không cần độc lập;
  - $h_K$ là một MAC để xác thực thực thể;
  - $h'_{K'}$ là một hoán vị giả ngẫu nhiên, hoặc một hàm băm 1 chiều có khóa dùng cho tạo khóa.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức vận chuyển khóa AKEP2:

- Các thông điệp trao đổi:
  - Giả thiết thông điệp T = (B, A, $r_A$, $r_B$)

$$
\begin{aligned}
A \rightarrow B: &\quad r_A && (1)\\
A \leftarrow B: &\quad T,\ h_K(T) && (2)\\
A \rightarrow B: &\quad (A, r_B),\ h_K(A, r_B) && (3)\\
W = h'_{K'}(r_B)
\end{aligned}
$$

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức vận chuyển khóa AKEP2:

- Mô tả các bước:
  - A tạo số ngẫu nhiên $r_A$ và gửi cho B;
  - B tạo số ngẫu nhiên $r_B$ và gửi cho A thông điệp $T = (B, A, r_A, r_B)$, kèm theo MAC của T sử dụng khóa K là $h_K(T)$;
  - Khi nhận được thông điệp $(T, h_K(T))$, A kiểm tra thông tin nhận dạng và $r_A$ (phải trùng với giá trị A gửi đi ở bước 1). Tiếp theo A kiểm tra MAC;
  - Tiếp theo A gửi B bộ giá trị $(A, r_B)$, kèm theo MAC của nó;
  - Khi nhận được thông điệp ở bước 3, B kiểm tra MAC và $r_B$ (phải trùng với giá trị B gửi đi ở bước 2);
  - Cả A và B tính toán khóa phiên $W = h'_{K'}(r_B)$.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức không khóa Shamir:

- Tóm tắt: A và B trao đổi 3 thông điệp thông qua kênh công cộng;
- Kết quả: Khóa bí mật K được chuyển bí mật từ A đến B (không đảm bảo xác thực);
- Khởi tạo (định nghĩa và xuất bản các tham số hệ thống):
  - Lựa chọn và xuất bản để dùng chung một số nguyên tố p sao cho việc tính toán logarit rời rạc modulo của p là không khả thi;
  - A và B chọn tương ứng 2 số nguyên bí mật a và b, với $0 \leq a, b \leq p - 2$, mỗi số a, b là số nguyên tố cùng nhau với p-1;
  - A và B lần lượt tính $a^{-1} \bmod (p-1)$ và $b^{-1} \bmod (p-1)$.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức không khóa Shamir:

- Các thông điệp trao đổi:

$$A \rightarrow B : K^a \bmod p \quad (1)$$
$$A \leftarrow B : (K^a)^b \bmod p \quad (2)$$
$$A \rightarrow B : (K^{ab})^{a^{-1}} \bmod p \quad (3)$$

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức không khóa Shamir:

- Mô tả các bước:
  - A chọn khóa ngẫu nhiên K để chuyển cho B, sao cho $1 \leq K \leq p - 1$;
  - A tính toán $(K^a \bmod p)$ và gửi B thông điệp (1);
  - Nhận được thông điệp (1) từ A, B tính $((K^a \bmod p)^b \bmod p)$ và gửi A thông điệp (2);
  - Nhận được thông điệp (2) từ B,
    A tính $(((K^a \bmod p)^b \bmod p)^{a^{-1}} \bmod (p-1))$ và gửi B thông điệp (3).
    Kết quả A có được $(K^b \bmod p)$, và gửi kết quả này cho B;
  - Nhận được thông điệp (3) từ A, B tính $((K^b \bmod p)^{b^{-1}} \bmod (p-1))$
    → K mod P là khóa chia sẻ giữa 2 bên.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức xác thực Kerberos:

- Kerberos cung cấp khả năng xác thực thực thể và thiết lập khóa sử dụng kỹ thuật mã hóa khóa đối xứng và một bên thứ ba.
- Kerberos liên quan đến 3 thực thể:
  - Máy khách A (Client);
  - Máy chủ hoặc máy kiểm tra B (Server / Verifier)
  - Máy chủ tin cậy T (Kerberos authentication server).
- A và B không chia sẻ khóa bí mật, nhưng T chia sẻ khóa bí mật dài hạn với cả A và B.
- Mục đích: B kiểm tra thông tin nhận dạng A.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức xác thực Kerberos:

- Kerberos cung cấp khả năng xác thực thực thể và thiết lập khóa sử dụng kỹ thuật mã hóa khóa đối xứng và một bên thứ ba.
- Kerberos liên quan đến 3 thực thể:
  - Máy khách A (Client);
  - Máy chủ hoặc máy kiểm tra B (Server / Verifier)
  - Máy chủ tin cậy T (Kerberos authentication server).
- A và B không chia sẻ khóa bí mật, nhưng T chia sẻ khóa bí mật dài hạn với cả A và B.
- Mục đích: B kiểm tra thông tin nhận dạng A.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức xác thực Kerberos:

- Mô tả vắn tắt giao thức:
  - A yêu cầu thông tin nhận dạng từ T để A xác thực bản thân với B;
  - T đóng vai trò như một trung tâm phân phối khóa KDC, T gửi A một khóa phiên (Session key) và 1 ticket đã được mã hóa cho B; Trong ticket có chứa khóa phiên và định danh của A;
  - A chuyển cho B ticket và ticket cho phép B kiểm tra thông tin nhận dạng của A.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức xác thực Kerberos (đơn giản hóa):

- Tóm tắt: A tương tác với máy chủ tin cậy T và bên B;
- Kết quả: A được xác thực với B, kèm theo thiết lập khóa;
- Ký hiệu:
  - E là giải thuật mã hóa khóa bí mật;
  - $N_A$ là số (nonce) chọn bởi A, $T_A$ là tem thời gian từ đồng hồ của A;
  - k là khóa phiên do T chọn; k sẽ được chia sẻ bởi A và B;
  - L là khoảng thời gian hợp lệ (còn gọi là thời gian sống).
- Khởi tạo:
  - A và T chia sẻ khóa $K_{AT}$, B và T chia sẻ khóa $K_{BT}$;
  - Đ.nghĩa: $ticket_B = E_{K_{BT}}(k, A, L)$; $authenticator = E_k(A, T_A, A_{Subkey})$

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức xác thực Kerberos (đơn giản hóa):

- Các thông điệp trao đổi:

$$
\begin{aligned}
A \rightarrow T: &\quad A, B, N_A && (1)\\
A \leftarrow T: &\quad \text{ticket}_B, E_{K_{AT}}(k, N_A, L, B) && (2)\\
A \rightarrow B: &\quad \text{ticket}_B, \text{authenticator} && (3)\\
A \leftarrow B: &\quad E_k(T_A, B^*_{\text{subkey}}) && (4)
\end{aligned}
$$

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức xác thực Kerberos (đơn giản hóa):

- Các bước thực hiện:
  - A sinh $N_A$ và gửi thông điệp (1) gồm (A, B, $N_A$) đến T;
  - T sinh khóa phiên k, tạo khoảng thời gian hợp lệ L cho ticket (gồm thời gian kết thúc và có thể cả thời gian bắt đầu);
  - T mã hóa (k, $N_A$, L, B) sử dụng khóa $K_{AT}$;
  - T tạo 1 ticket chứa (k, A, L) và mã hóa ticket bằng $K_{BT}$;
  - T gửi thông điệp (2) cho A;
  - Nhận được thông điệp (2), A giải mã phần ngoài ticket và có được (k, $N_A$, L, B);

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức xác thực Kerberos (đơn giản hóa):

- Các bước thực hiện:
  - A kiểm tra $N_A$ vừa giải mã với $N_A$ tạo ra ban đầu và lưu L để tham chiếu;
  - A tạo authenticator = $E_k(A, T_A, A_{Subkey})$, $A_{Subkey}$ là tùy chọn;
  - A gửi thông điệp (3) gồm $ticket_B$ và authenticator đến B;
  - Nhận được thông điệp (3), B giải mã ticket sử dụng khóa $K_{BT}$ để khôi phục (k, A, L).
  - B sử dụng k để giải mã authenticator.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa đối xứng

## ❖ Giao thức xác thực Kerberos (đơn giản hóa):

- Các bước thực hiện:
  - B kiểm tra:
    - Tên nhận dạng A trong ticket và authenticator phải trùng nhau;
    - Tem thời gian $T_A$ trong authenticator phải hợp lệ;
    - Thời gian cục bộ của B phải nằm trong giới hạn hợp lệ L;
  - ➔ Nếu tất cả các kiểm tra là passed → B thông báo A đã được xác thực và lưu $A_{Subkey}$ (nếu có).

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

- Vận chuyển khóa dựa trên mã hóa khóa công khai không có chữ ký – Giao thức Needham-Schroeder:
  - Tóm tắt: A và B trao đổi 3 thông điệp;
  - Kết quả: Xác thực thực thể, xác thực khóa và vận chuyển khóa.
  - Các ký hiệu:
    - $P_X(Y)$ chỉ việc mã hóa dữ liệu Y bằng khóa công khai của X;
    - $P_X(Y_1, Y_2)$ chỉ việc mã hóa dữ liệu ghép $P_X(Y_1, Y_2)$ bằng khóa công khai của X;
    - $k_1$, $k_2$ là khóa phiên đối xứng do A và B lựa chọn tương ứng.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

- Vận chuyển khóa dựa trên mã hóa khóa công khai không có chữ ký – Giao thức Needham-Schroeder:
  - Khởi tạo một lần:
    - Các bên A, B đều sở hữu khóa công khai đảm bảo tính xác thực, hoặc mỗi bên đều có chứng chỉ số chứa khóa công khai. Khi đó cần thêm 1 thông điệp để chuyển chứng chỉ.
  - Các thông điệp:

$$A \rightarrow B: \quad P_B(k_1, A) \quad (1)$$
$$A \leftarrow B: \quad P_A(k_1, k_2) \quad (2)$$
$$A \rightarrow B: \quad P_B(k_2) \quad (3)$$

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

- Vận chuyển khóa dựa trên mã hóa khóa công khai không có chữ ký – Giao thức Needham-Schroeder:
  - Mô tả các bước:
    - A gửi B thông điệp (1);
    - B giải mã (1) để khôi phục $k_1$ và tạo thông điệp (2) gửi B;
    - A giải mã (2), kiểm tra $k_1$ đảm bảo trùng với $k_1$ trong (1). Nếu $k_1$ chưa bao giờ được sử dụng, việc này xác thực được B và đảm bảo B biết khóa $k_1$. A tạo thông điệp (3) gửi B;
    - B giải mã (3), kiểm tra $k_2$ đảm bảo trùng với $k_2$ trong (2).
    - Khóa phiên dùng chung cho A và B có thể được tính từ hàm $f(k_1, k_2)$ – f có thể là hàm một chiều.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

- Vận chuyển khóa dựa trên mã hóa khóa công khai không có chữ ký – Giao thức Needham-Schroeder:
  - Nhận xét: Giao thức Needham-Schroeder có thể được cải tiến để không cần mã hóa thông điệp (3):
    - A, B sinh các số ngẫu nhiên $r_1$, $r_2$.

$$\begin{aligned} A \rightarrow B: &\quad P_B(k_1, A, r_1) && (1') \\ A \leftarrow B: &\quad P_A(k_2, r_1, r_2) && (2') \\ A \rightarrow B: &\quad r_2 && (3') \end{aligned}$$

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

- Vận chuyển khóa dựa trên mã hóa khóa công khai kết hợp với chữ ký:
  - Các ký hiệu:
    - y là dữ liệu đầu vào;
    - $S_A(y)$ là chữ ký của A sử dụng khóa riêng trên dữ liệu y;
    - $P_B(y)$ là bản mã của dữ liệu y, sử dụng khóa công khai của B;
  - Giả thiết:
    - Không thể khôi phục đc y từ chữ ký $S_A(y)$.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

- Vận chuyển khóa dựa trên mã hóa khóa công khai kết hợp với chữ ký – Các phương pháp:
  - Mã hóa các khóa đã được ký (Encrypting signed keys);
  - Mã hóa và ký riêng rẽ (Encrypting and signing separately);
  - Ký các khóa đã mã hóa (Signing encrypted keys);
  - Giao thức xác thực X.509.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

- Mã hóa các khóa đã được ký:
  - k là khóa phiên cần trao đổi giữa A và B;
  - $t_A$ là tem thời gian A gửi kèm xác định thời gian tồn tại hợp lệ của thông điệp;
  - $S_A()$: là chữ ký số của A, ký sử dụng khóa bí mật của A;
  - $P_B()$: là bản mã sử dụng khóa công khai của B.

$$A \rightarrow B: \quad P_B(k,\ t_A{}^*,\ S_A(B, k, t_A{}^*))$$

  - Trường hợp $S_A()$ có thể giải mã (như sử dụng RSA):

$$A \rightarrow B: \quad P_B(S_A(B, k, t_A{}^*))$$

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

- Mã hóa và ký riêng rẽ :
  - Khóa k được mã hóa sử dụng khóa công khai của B, và k được ký bởi khóa bí mật của A riêng rẽ.
  - Điều kiện: Thuật toán chữ ký không cho phép khôi phục k từ chữ ký $S_A(k)$. Cần sử dụng các hàm băm 1 chiều để tạo bản tóm tắt thông điệp cho tạo chữ ký.

$$A \rightarrow B: \quad P_B(k,\ t_A{}^*),\ S_A(B, k, t_A{}^*)$$

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

- Ký các khóa đã mã hóa:
  - Xác thực thực thể với tem thời gian (1)
  - Xác thực thực thể sử dụng giao thức Thách thức – Trả lời (2): khóa k được tính từ $k_1$ và $k_2$.

**(1)** $A \rightarrow B: \quad t_A{}^*,\ P_B(A,k),\ S_A(B,t_A{}^*,P_B(A,k))$

$A \rightarrow B: \quad r_A$

**(2)** $A \leftarrow B: \quad r_B,\ P_A(B,k_1),\ S_B(r_B,r_A,A,P_A(B,k_1))$

$A \rightarrow B: \quad P_B(A,k_2),\ S_A(r_A,r_B,B,P_B(A,k_2))$

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

## ❖ Giao thức xác thực X.509:

- Giao thức xác thực X.509 cho phép xác thực thực thể "mạnh" (strong authentication) với 2 thông điệp (2-way) hoặc 3 thông điệp (3-way) trao đổi;
  - Xác thực dựa trên 2 thông điệp trao đổi: Xác thực thực thể với tem thời gian;
  - Xác thực dựa trên 3 thông điệp trao đổi: Xác thực thực thể sử dụng giao thức Thách thức – Trả lời.
- Strong authentication: là phương pháp xác thực mạnh hơn so với phương pháp dùng mật khẩu truyền thống.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

## ❖ Giao thức xác thực X.509 - two-way authentication:

- Các ký hiệu:
  - $P_X(y)$: bản mã của dữ liệu y sử dụng khóa công khai của X;
  - $S_X(y)$: chữ ký của X sử dụng khóa riêng của X trên dữ liệu y;
  - $r_A$, $r_B$: các số ngẫu nhiên sinh bởi A, B và không được dùng lại (để tránh tấn công kiểu phát lại);
  - $cert_X$: chứng chỉ số của X, kết hợp khóa công khai của X với thông tin định danh của X. Khóa công khai của X cho phép mã hóa và kiểm tra chữ ký (giải mã).

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

## ❖ Giao thức xác thực X.509 - two-way authentication:

- Cài đặt hệ thống:
  - Mỗi bên tham gia cần có cặp khóa công khai để mã hóa và khóa riêng để ký thông điệp;
  - A phải lấy và xác thực được khóa công khai của B để mã hóa trước khi thực hiện các bước xác thực tiếp theo. Việc này có thể cần bổ sung việc trao đổi các thông điệp và tính toán.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

## ❖ Giao thức xác thực X.509 - two-way authentication:

- Cho:

$$D_A = (t_A, r_A, B, \text{data}_1{}^*, P_B(k_1)^*)$$

$$D_B = (t_B, r_B, A, r_A, \text{data}_2{}^*, P_A(k_2)^*)$$

- Các thông điệp trao đổi:

$$A \rightarrow B: \quad cert_A, D_A, S_A(D_A) \quad (1)$$

$$A \leftarrow B: \quad cert_B, D_B, S_B(D_B) \quad (2)$$

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

## ❖ Giao thức xác thực X.509 - two-way authentication:

- Mô tả các bước thực hiện:
  - A tạo tem thời gian $t_A$ chỉ rõ thời gian hết hạn của thông điệp, tạo $r_A$, tùy chọn tạo khóa bí mật $k_1$, và gửi B thông điệp (1);
  - B kiểm tra tính xác thực của $cert_A$ (kiểm tra chữ ký, ngày hết hạn,…), tách lấy khóa công khai của A và kiểm tra chữ ký của A trên khối dữ liệu $D_A$.
  - B kiểm tra định danh của nó trong thông điệp (1), tem thời gian $t_A$ của A, và tham số $r_A$ có bị lặp (dùng lại) hay không;
  - Nếu tất cả các kiểm tra đều cho kết quả là hợp lệ, B xác nhận việc xác thực A thành công. B giải mã $P_B(k_1)$ sử dụng khóa riêng của mình và lưu $k_1$ làm khóa chia sẻ.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

## ❖ Giao thức xác thực X.509 - two-way authentication:

- Mô tả các bước thực hiện:
  - B tạo tem thời gian $t_B$, tạo $r_B$, tùy chọn tạo khóa bí mật $k_2$, và gửi A thông điệp (2). Các thành phần $data_2$ và khóa bí mật $k_2$ là tùy chọn;
  - A cũng tiến hành các thủ tục kiểm tra các thông tin trong thông điệp (2) tương tự B đã thực hiện. Nếu tất cả các kiểm tra đều cho kết quả hợp lệ, A xác nhận việc xác thực B thành công;
  - A lưu khóa $k_2$ để sử dụng. Như vậy A và B đã xác thực được nhau và cùng chia sẻ khóa $k_1$ và $k_2$.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

## ❖ Giao thức xác thực X.509 - two-way authentication:

- Nhận xét:
  - Chuẩn X.509 giả thiết sử dụng một hệ mã hóa khóa công khai như RSA được sử dụng, trong đó, một cặp khóa có thể được sử dụng để mã hóa và tạo chữ ký. Tuy nhiên, nó có thể được sử đổi để hoạt động để làm việc với các khóa dùng để mã hóa và tạo chữ ký riêng rẽ (không phải thuộc 1 cặp);
  - Do chuẩn X.509 không gồm tên nhận dạng của A trong bản mã $P_B(k_1)$ thuộc $D_A$, nên không thể đảm bảo bên ký thực sự biết (hoặc là nguồn của) khóa mã bản rõ.

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

- G.thức xác thực X.509 - three-way authentication:
  - Tóm tắt: A và B trao đổi 3 thông điệp;
  - Kết quả: Hai bên A và B xác thực được nhau và trao đổi khóa chia sẻ mà không cần sử dụng tem thời gian;

# 4.5 QL & PP khóa – Các giao thức PP & thỏa thuận khóa - Vận chuyển khóa dựa trên mã hóa khóa công khai

- G.thức xác thực X.509 - three-way authentication:
  - Điểm khác biệt so với X.509 two-way authentication:
    - Tem thời gian $t_A$, $t_B$ được đặt là 0 và không cần kiểm tra;
    - Khi nhận được thông điệp (2), A kiểm tra $r_A$ nhận được phải giống $r_A$ ban đầu;
    - A gửi thông điệp thứ 3 cho B:

$$A \rightarrow B: \quad (r_B, B),\ S_A(r_B, B) \quad (3)$$

    - Khi nhận được thông điệp (3), B kiểm tra chữ ký trùng với chuỗi được tạo từ bản rõ, định danh B phải khớp và tham số $r_B$ nhận được phải trùng với $r_B$ gửi đi trong thông điệp (2).

# 4.6 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

# 4.6.1 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;
  - Một hệ chữ ký số (Digital Signature Scheme) bao gồm giải thuật tạo chữ ký số và giải thuật kiểm tra chữ ký số.

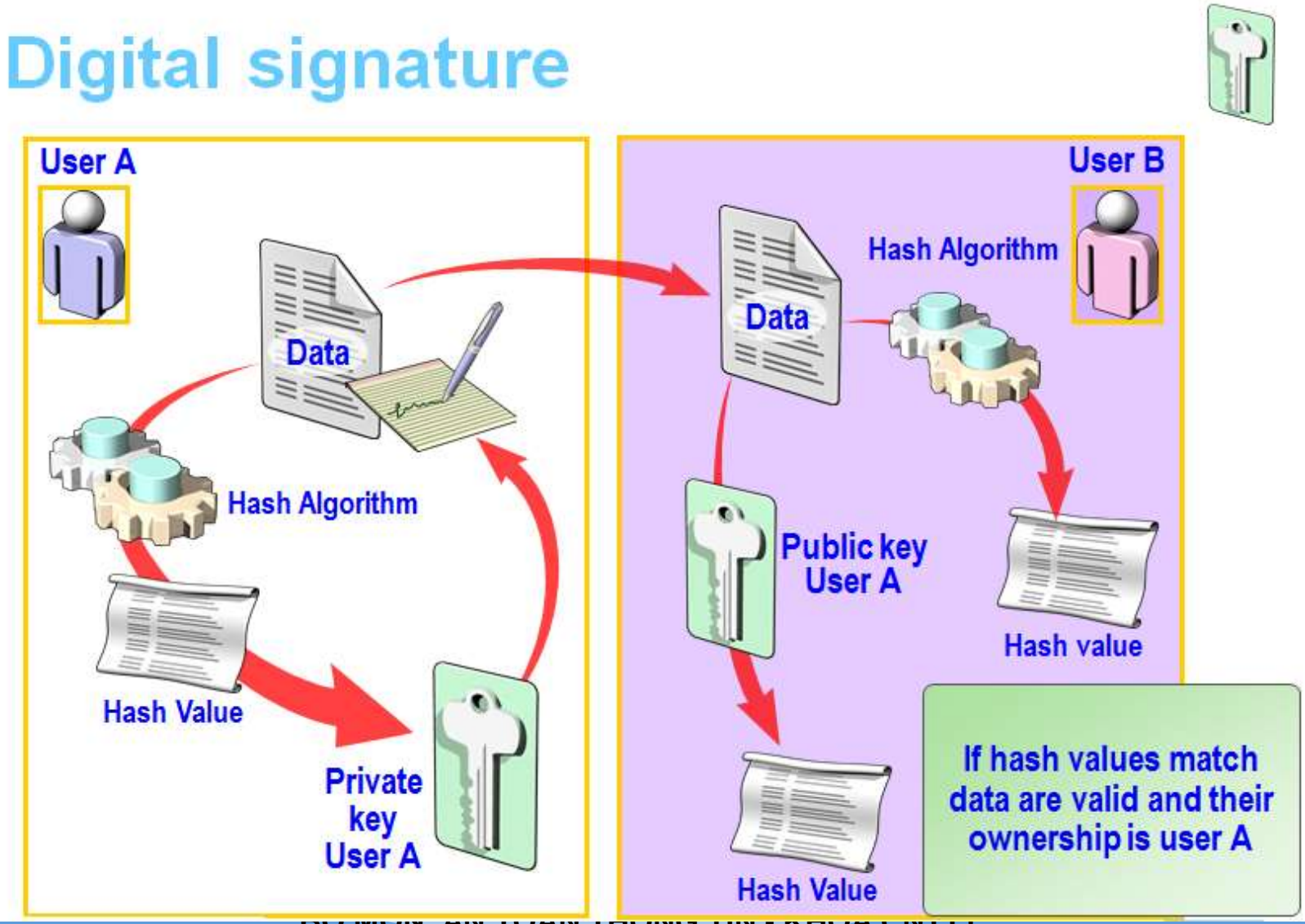
# 4.6.1 Chữ ký số

- Một số khái niệm:
  - 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.

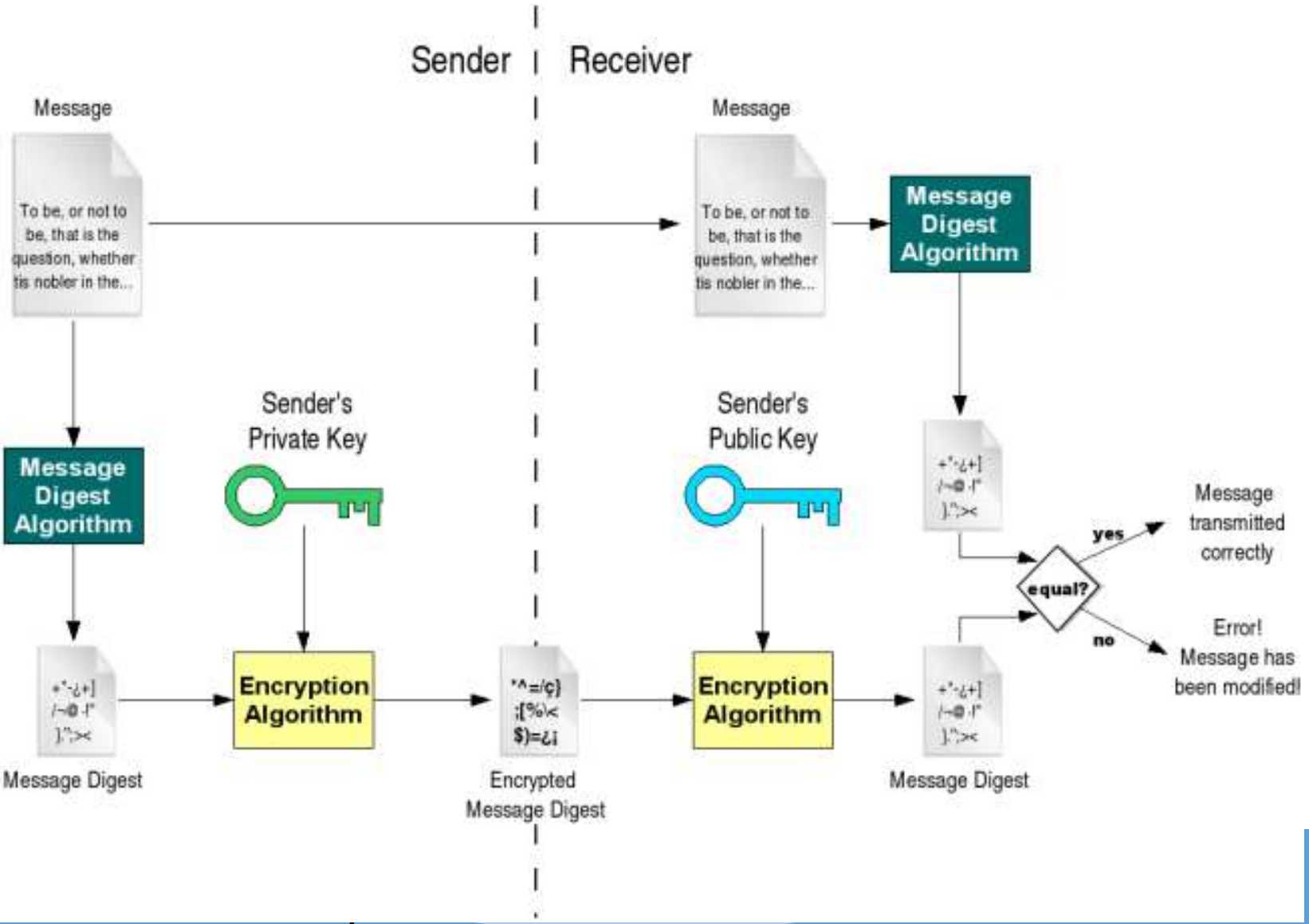
# 4.6.1 Chữ ký số


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

# 4.6.1 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

# 4.6.1 Chữ ký số - Quá trình ký và kiểm tra



# 4.6.1 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.

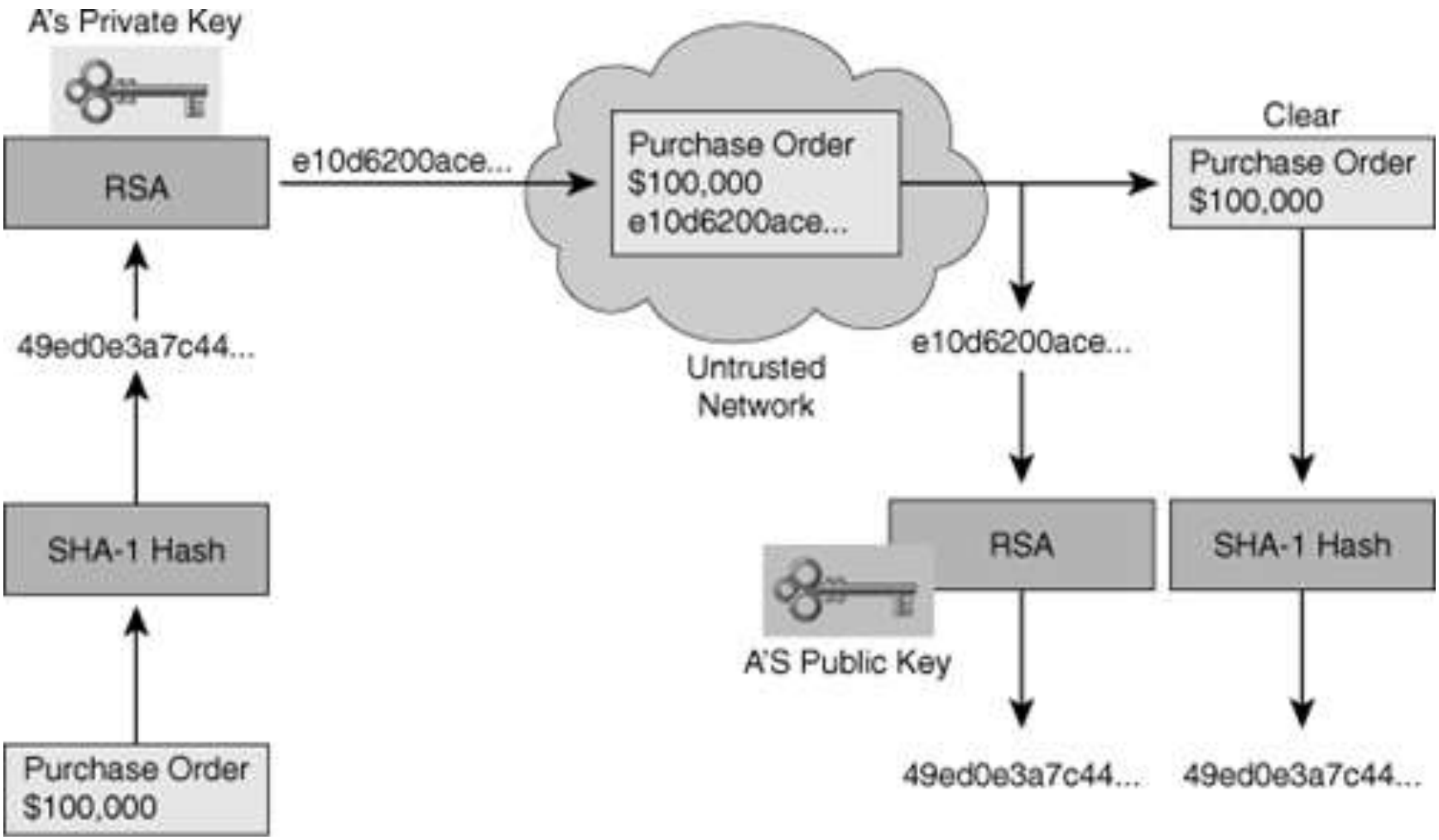
# 4.6.1 Chữ ký số - Quá trình kiểm tra

- 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.

# 4.6.1 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.

# 4.6.1 Chữ ký số - Giải thuật chữ ký số RSA



# 4.6.1 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.

# 4.6.1 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.

# 4.6.1 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.

# 4.6.1 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).

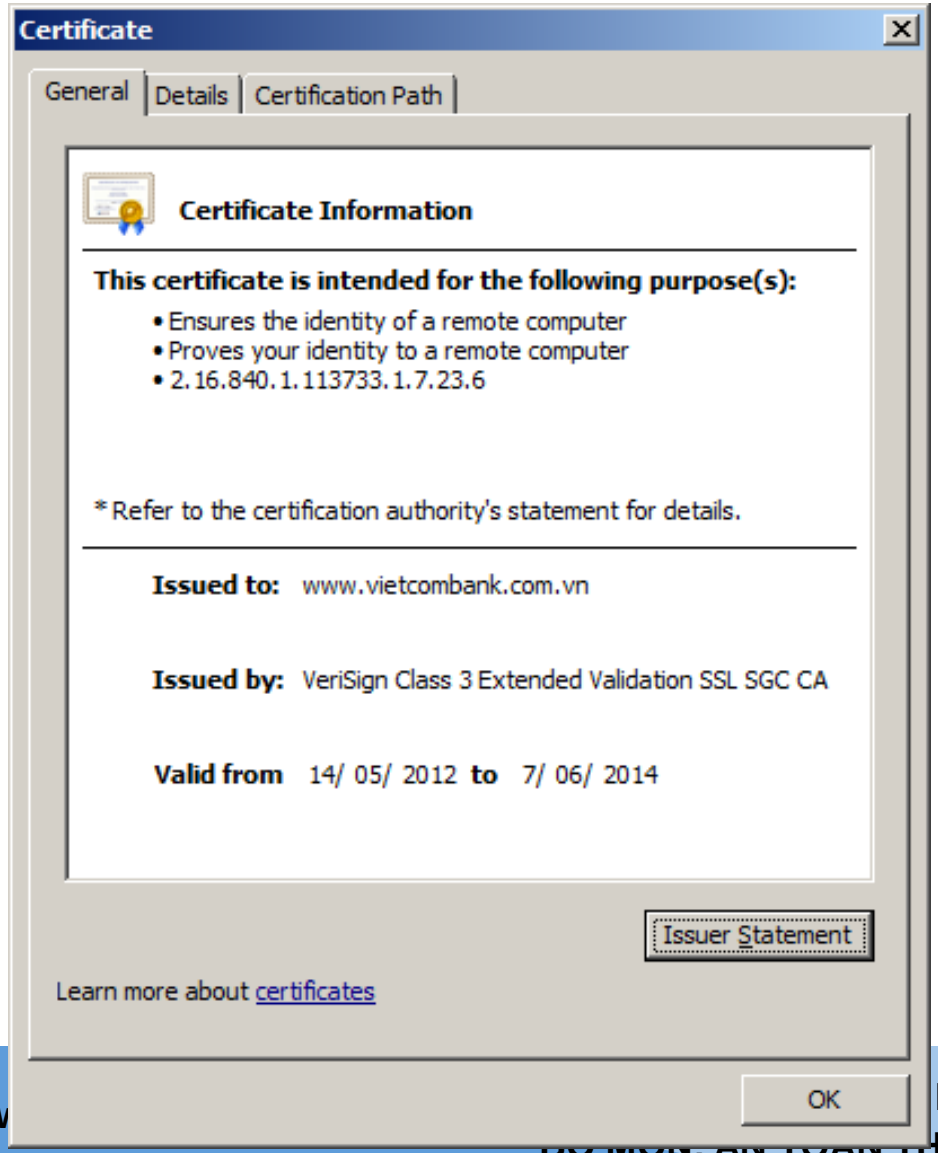
# 4.6.1 Chữ ký số - Giải thuật chữ ký số DSA

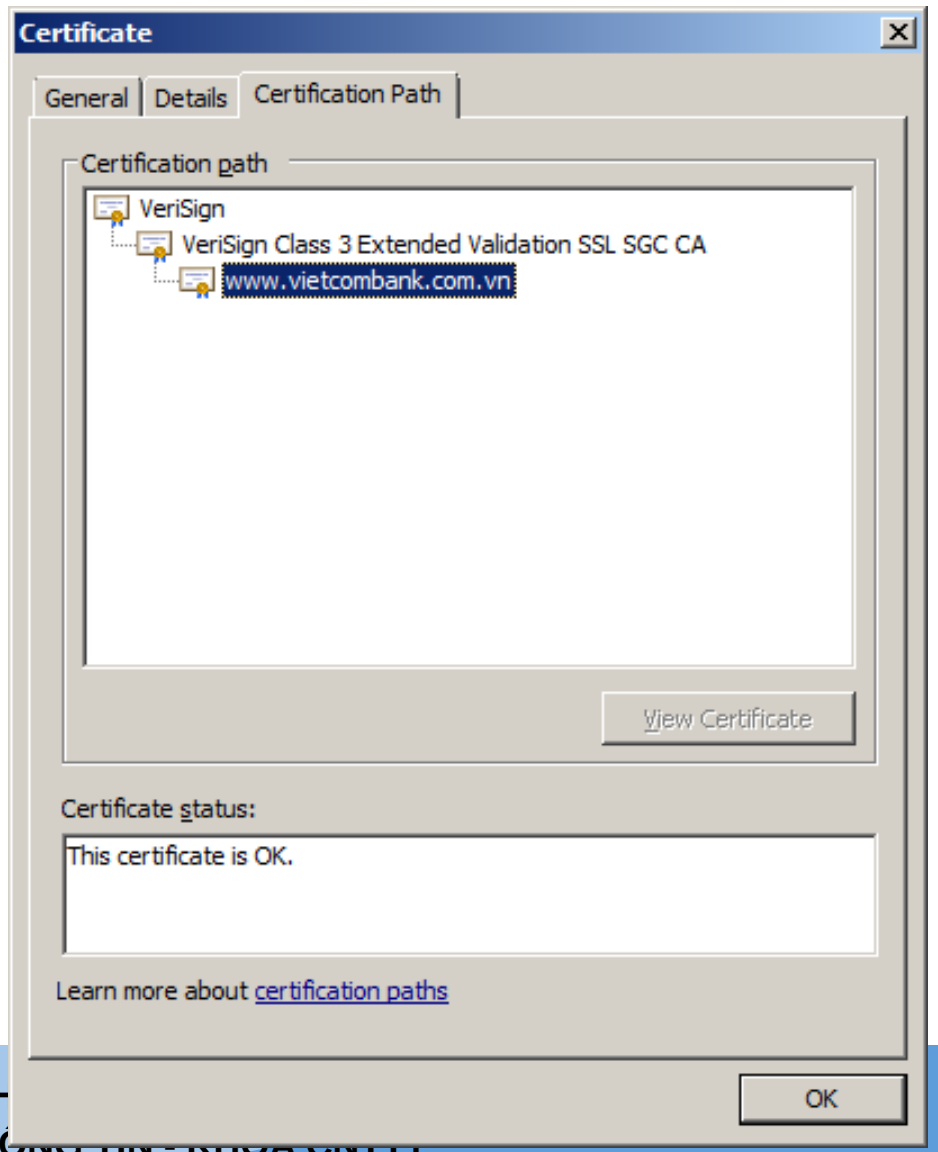
- Kiểm tra chữ ký của thông điệp:
  - Loại bỏ chữ ký nếu r và s không thỏa mãn $0 < r , s < q$;
  - Tính H(m) từ thông điệp nhận được;
  - Tính $w = s^{-1} \bmod q$;
  - Tính $u_1 = H(m) * w \bmod q$;
  - Tính $u_2 = r * w \bmod q$;
  - Tính $v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$;
  - Chữ ký là xác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

# 4.6.2 Chứng chỉ số - Giới thiệu

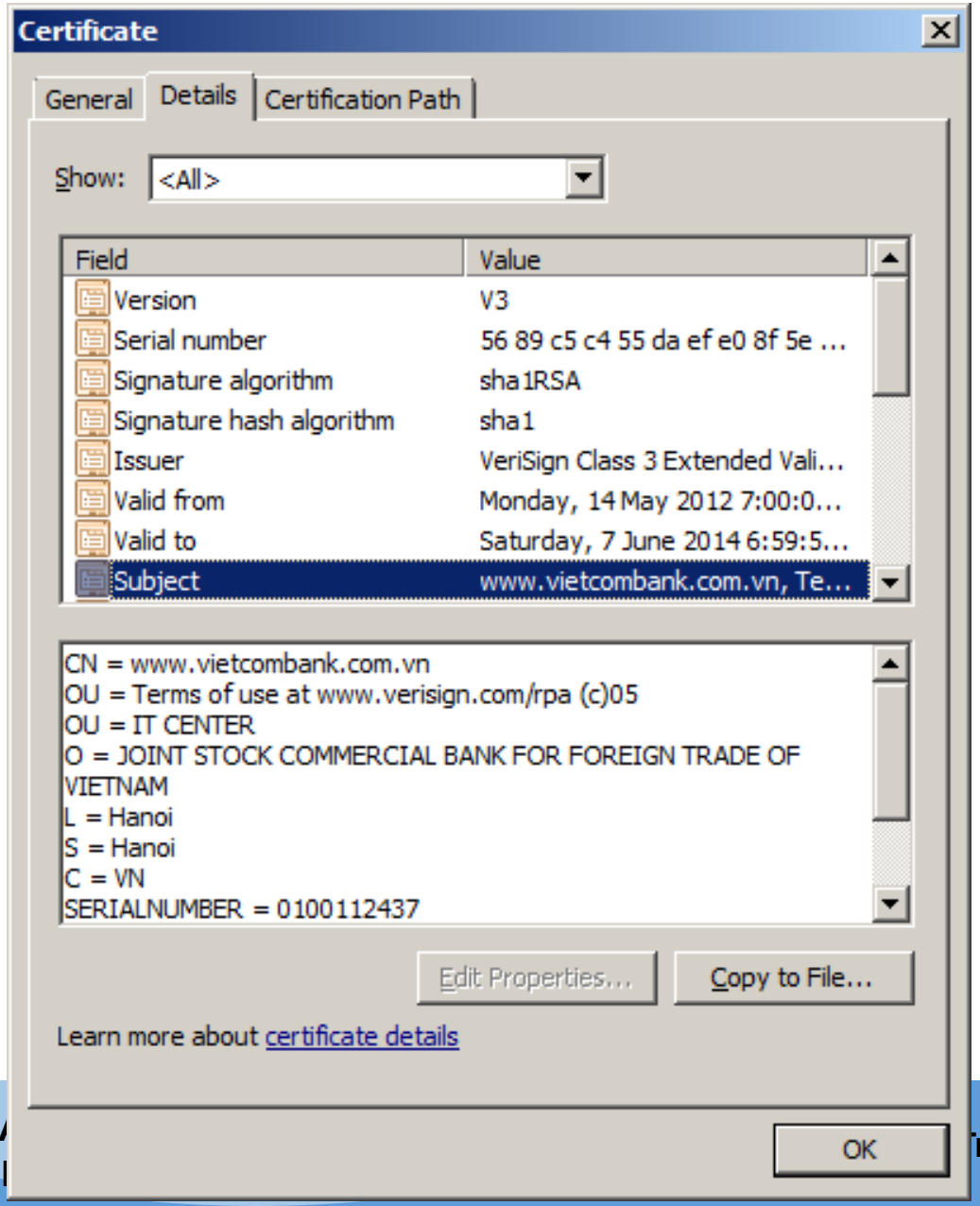
- Chứng chỉ số (Digital certificate), còn gọi là chứng chỉ khóa công khai (Public key certificate), hay chứng chỉ nhận dạng (Identity certificate) là một tài liệu điện tử sử dụng một ***chữ ký số*** để liên kết một ***khóa công khai*** và ***thông tin nhận dạng*** của một thực thể:
  - Chữ ký số: là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

# 4.6.2 Chứng chỉ số - Nội dung



# 4.6.2 Chứng chỉ số - Nội dung



## 4.6.2 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;

# 4.6.2 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;

## 4.6.2 Chứng chỉ số - Nội dung

- Nội dung của trượng 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.

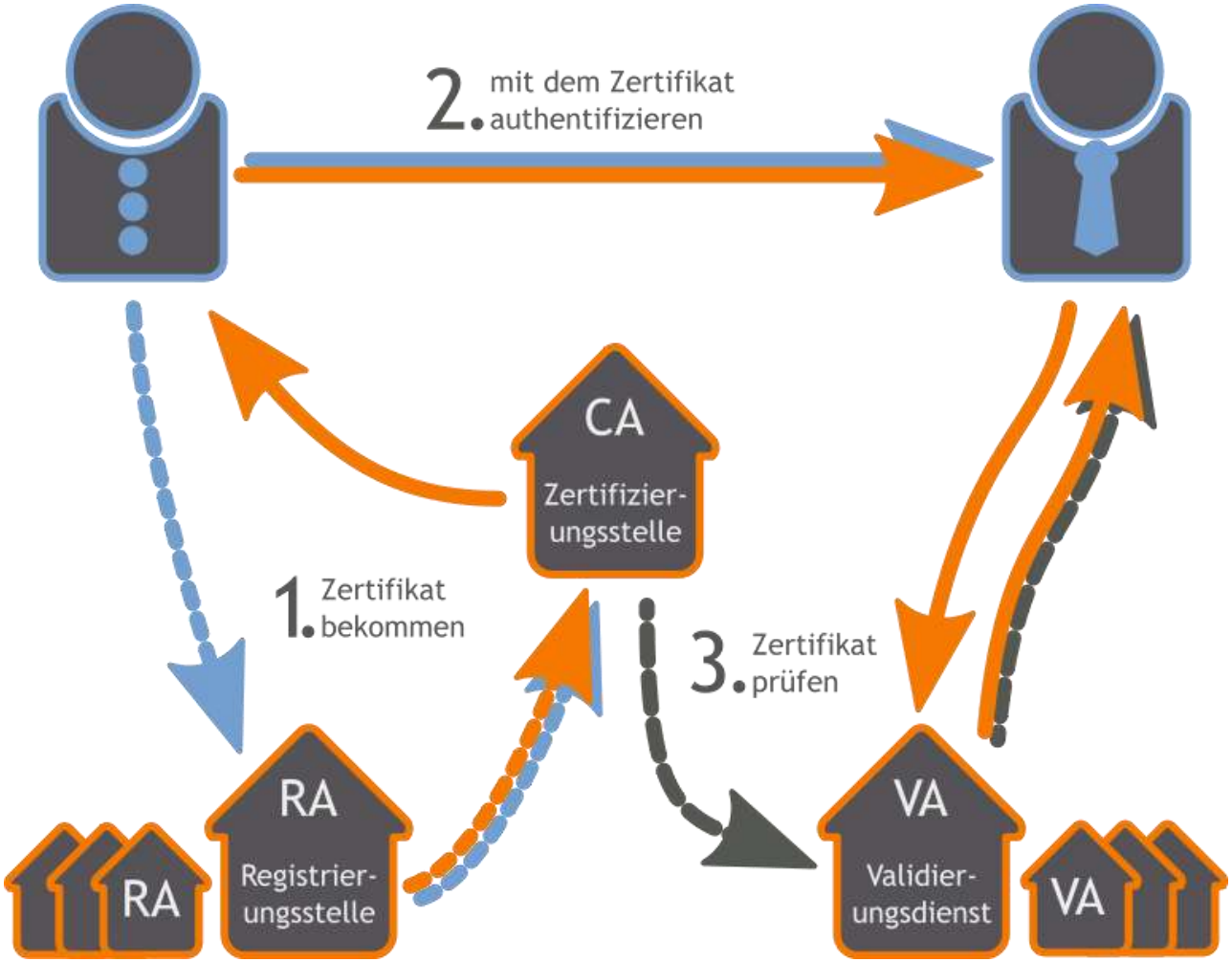
# 4.6.2 Chứng chỉ số - Sử dụng

- 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);
  - 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.

# 4.6.3 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;
- 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;

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



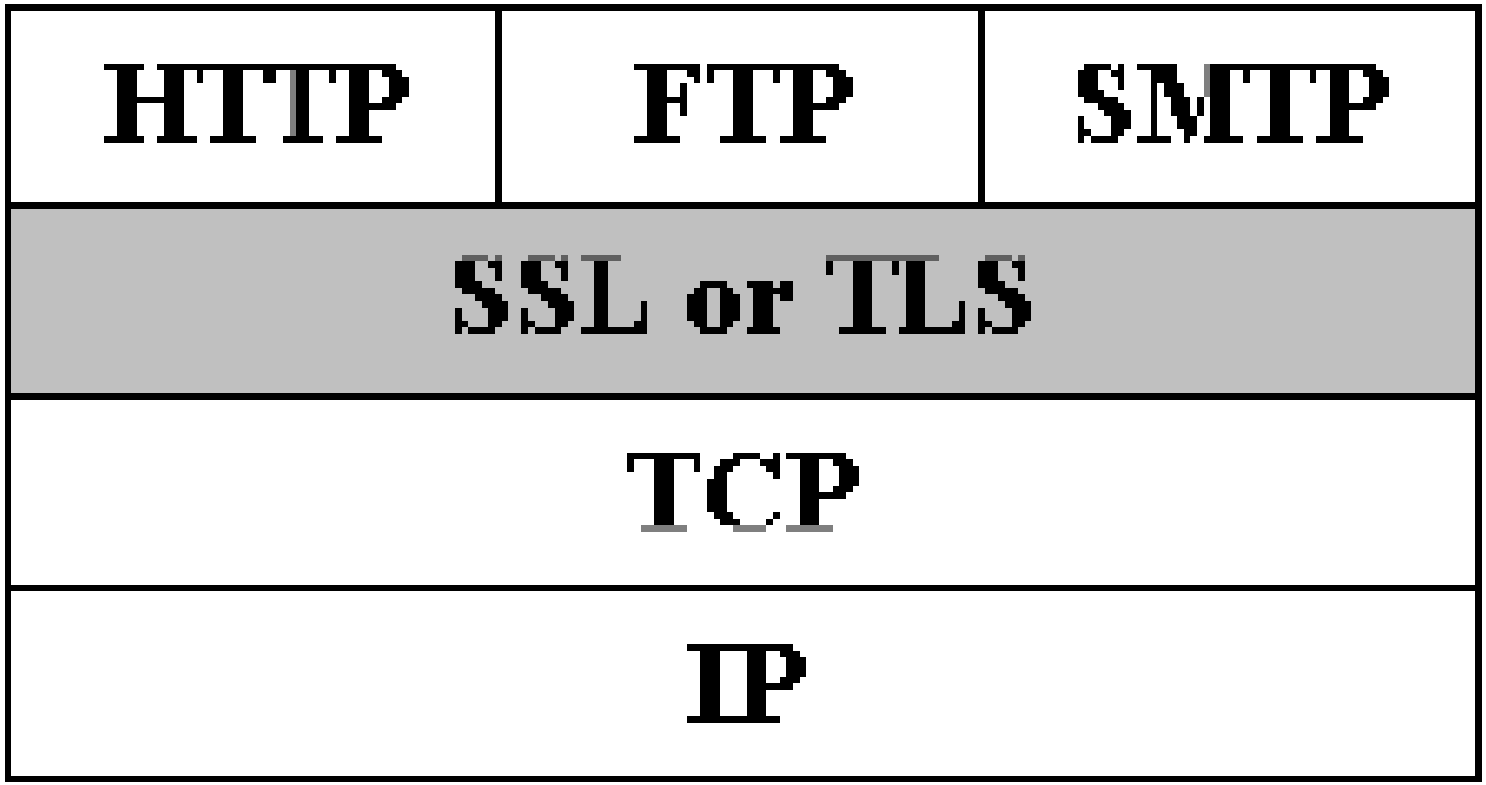
# 4.7 Các giao thức đảm bảo ATTT dựa trên mã hóa

- 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)

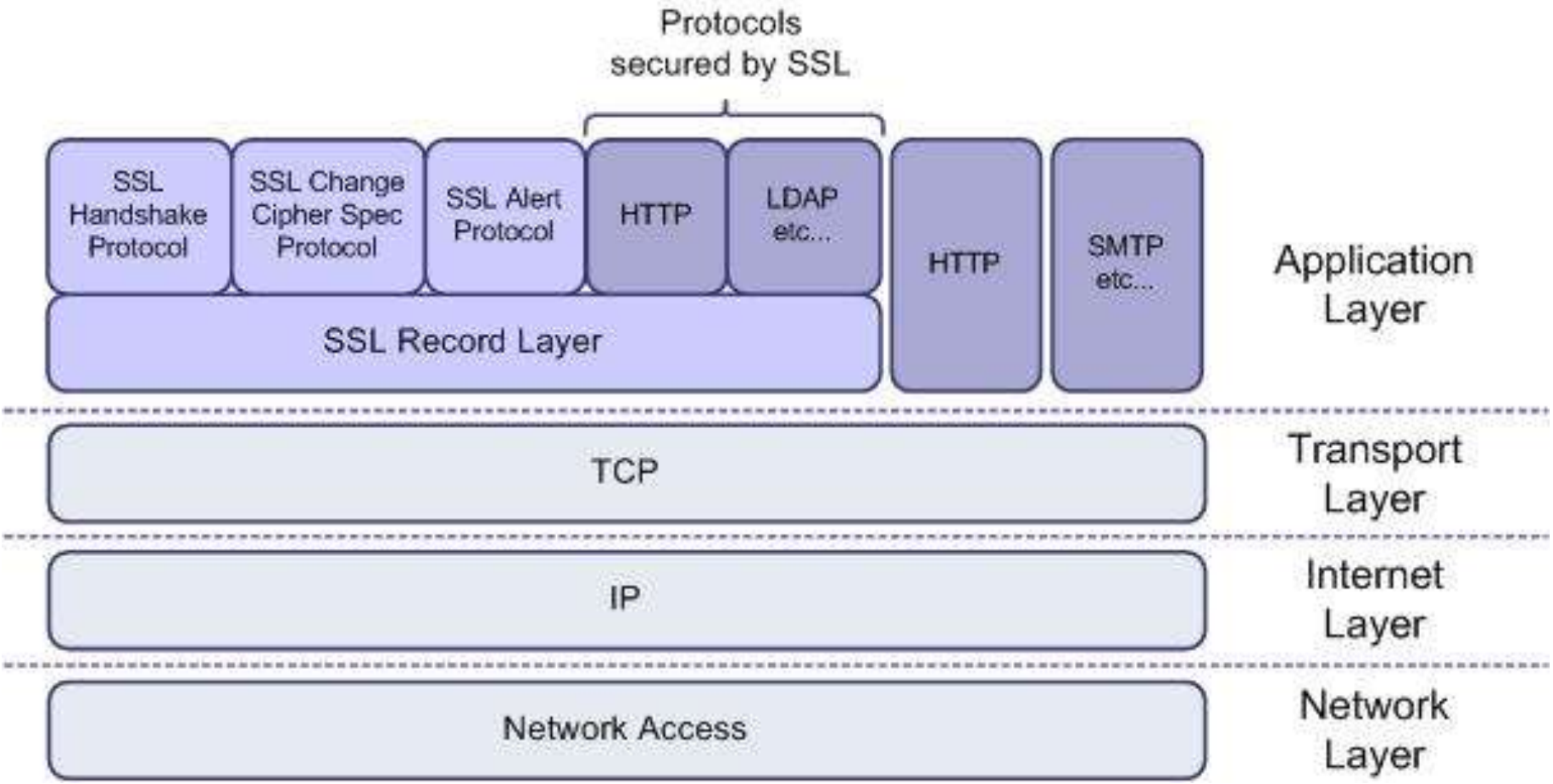
# 4.7 Các giao thức đảm bảo ATTT – SSL/TLS

- SSL do công ty Netscape phát minh, TLS được xây dựng dựa trên SSL và do IETF phê chuẩn.
- 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).

# 4.7 Các giao thức đảm bảo ATTT – SSL/TLS



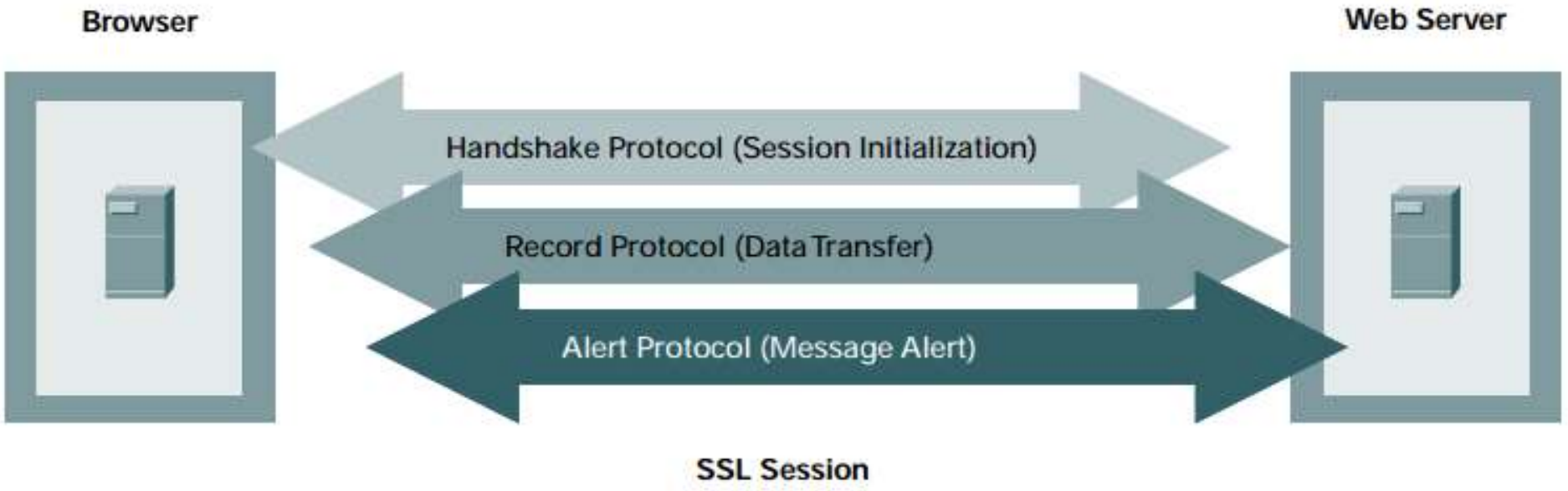
# 4.7 Các giao thức đảm bảo ATTT – SSL/TLS



# 4.7 Các giao thức đảm bảo ATTT – SSL/TLS

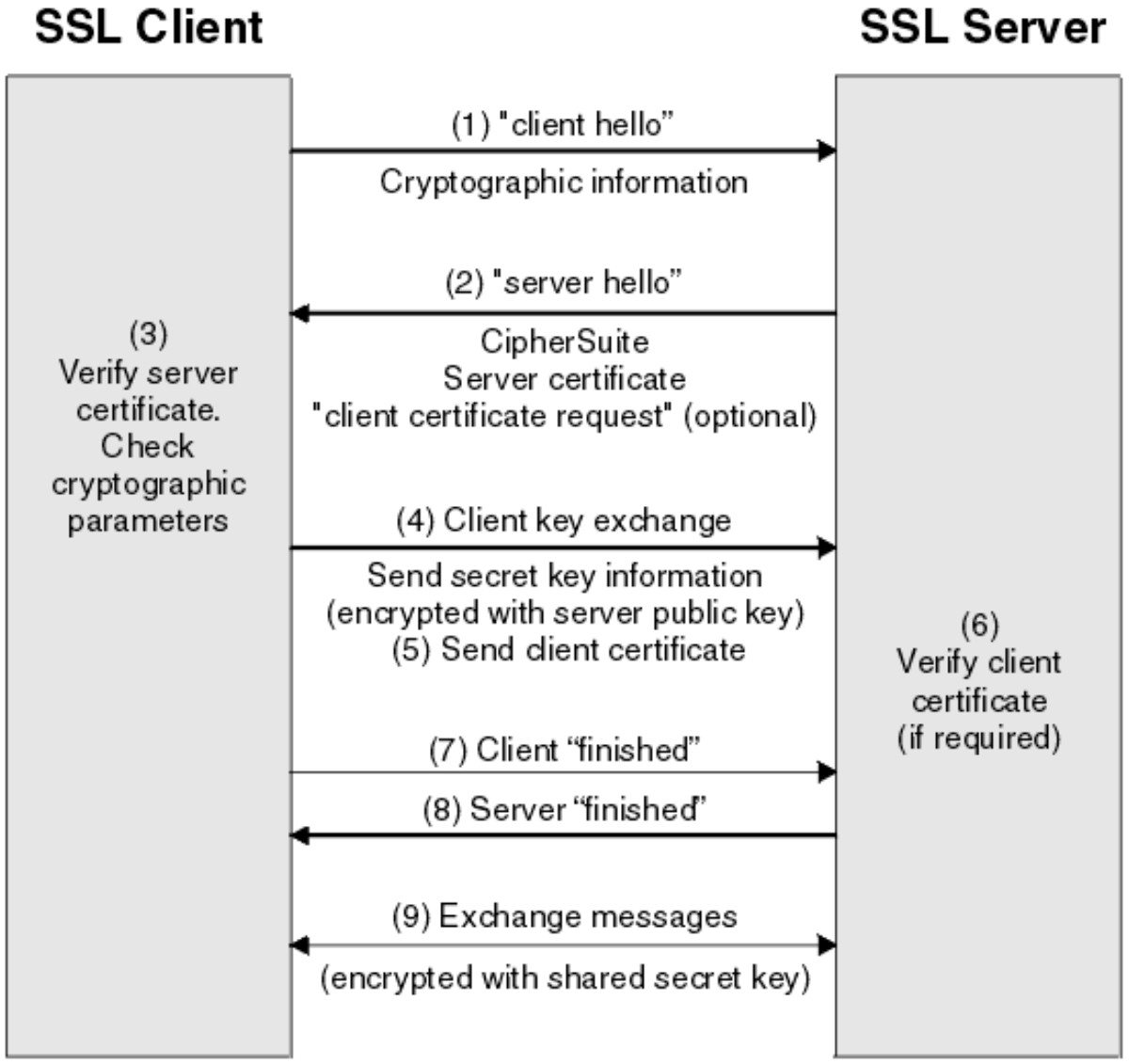
- Các giao thức con của SSL:
  - SSL Handshake Protocol: Giao thức bắt tay của SSL. Có nhiệm vụ trao đổi các thông điệp xác thực thực thể và thiết lập các thông số cho phiên làm việc;
  - SSL Change Cipher Spec Protocol: Giao thức thiết lập việc sử dụng các bộ mã hóa được hỗ trợ bởi cả 2 bên truyền thông;
  - 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.

# 4.7 Các giao thức đảm bảo ATTT – SSL/TLS



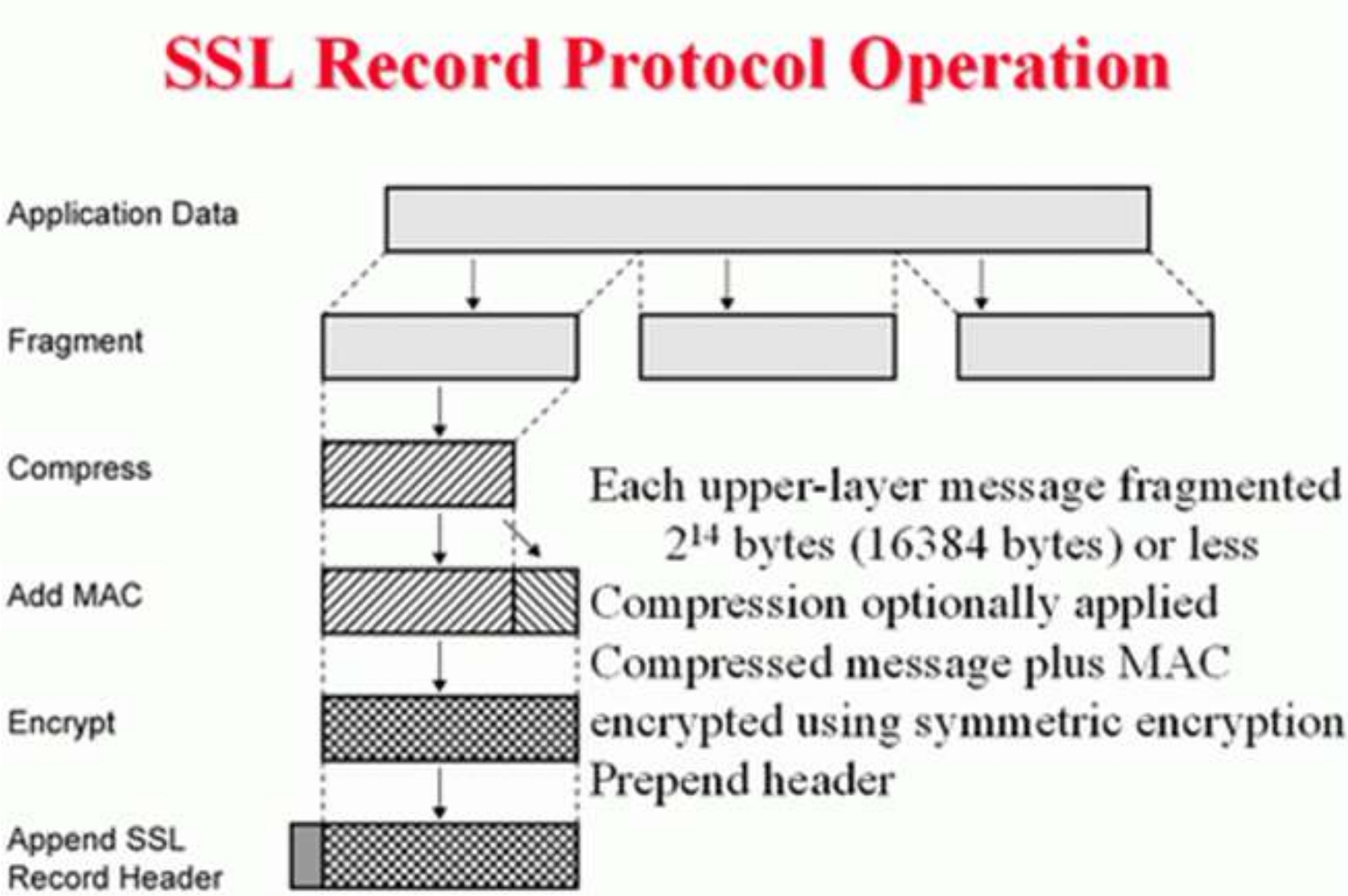
# 4.7 Các giao thức đảm bảo ATTT – SSL/TLS

- SSL Handshake Protocol:



# 4.7 Các giao thức đảm bảo ATTT – SSL/TLS

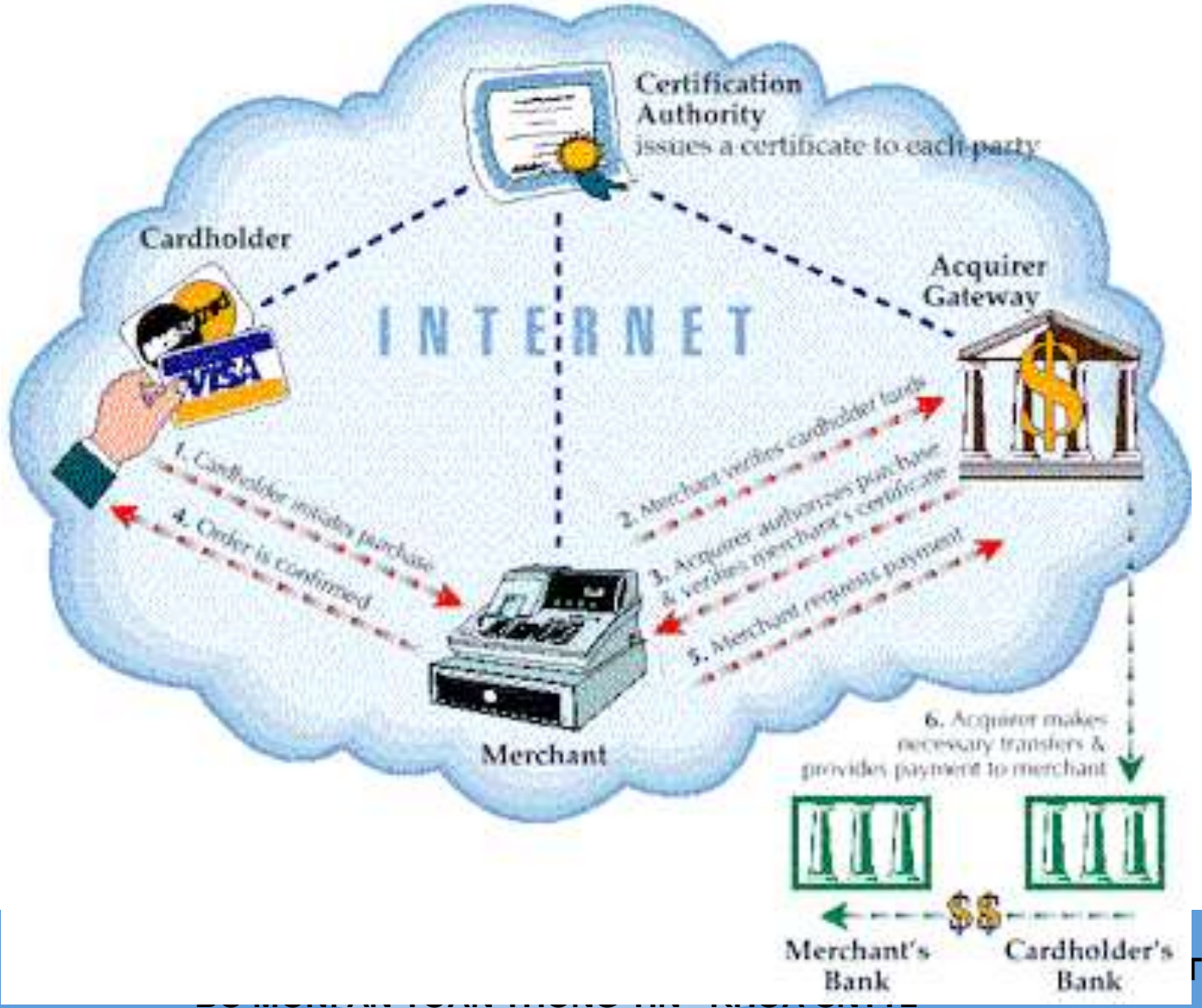
- SSL Record Protocol:



# 4.7 Các giao thức đảm bảo ATTT – SET

- SET là giao thức cho phép thanh toán điện tử an toàn, sử dụng thẻ tín dụng;
- 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

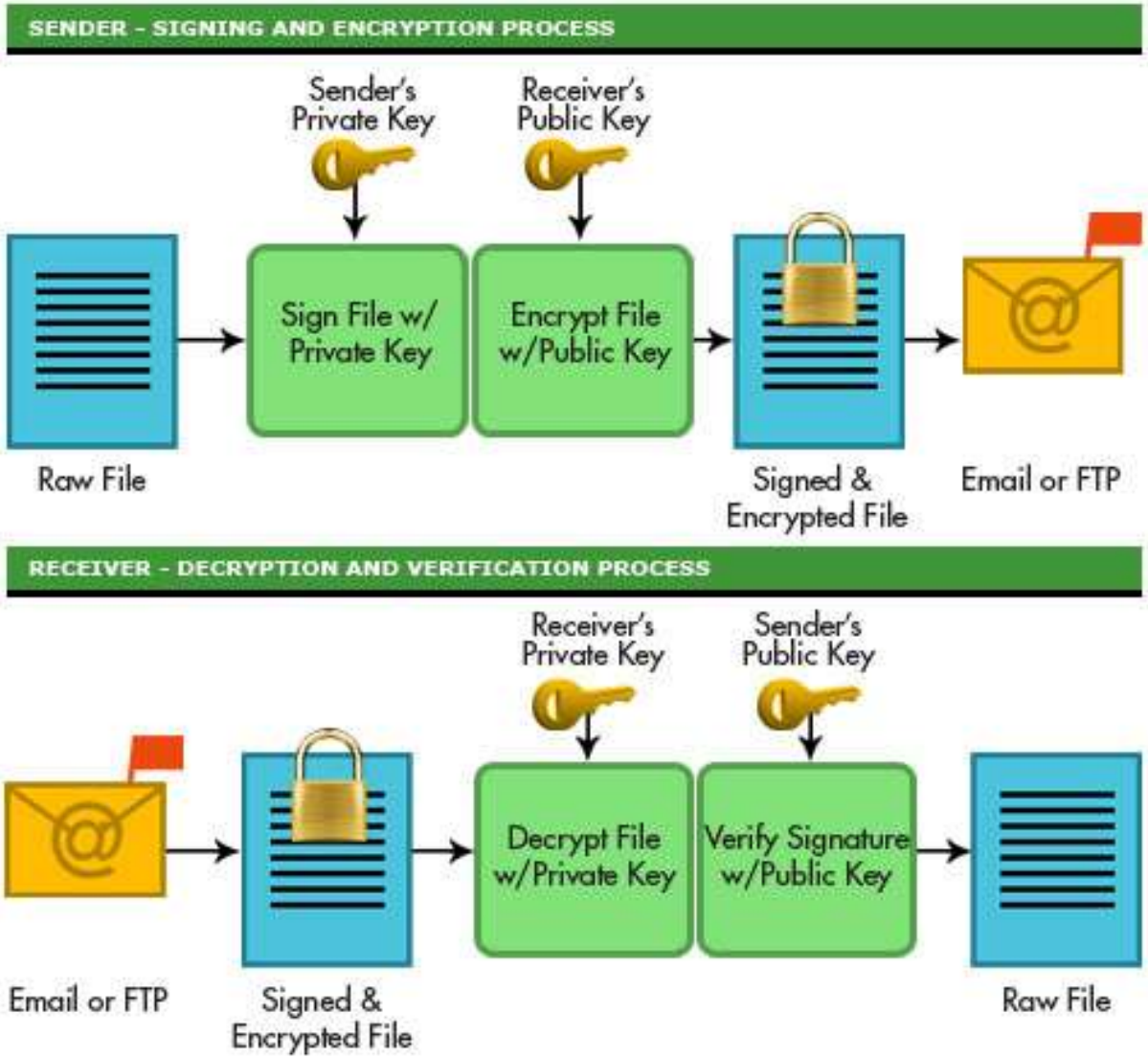
# 4.7 Các giao thức đảm bảo ATTT – SET



# 4.7 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.

# 4.7 Các giao thức đảm bảo ATTT – PGP



# Tổng kết các PP đảm bảo ATTT dựa trên mã hóa