



BẢO VỆ DỮ LIỆU  
MÃ HÓA

# Nội dung

- Giới thiệu
- Phân tích mã
- Các thuật ngữ
- Mã hóa bất đối xứng – khóa công khai
- RSA
- Mã hóa đối xứng
- Chống tấn công

# Giới thiệu

- Nếu không có mã hóa thì bất kỳ ai cũng có thể dễ dàng truy cập vào đường truyền dữ liệu giữa các máy tính để xem, sửa chữa,...
- Bảo mật là vấn đề hết sức quan trọng trong giao dịch thương mại và nhiều kiểu trao đổi thông tin khác

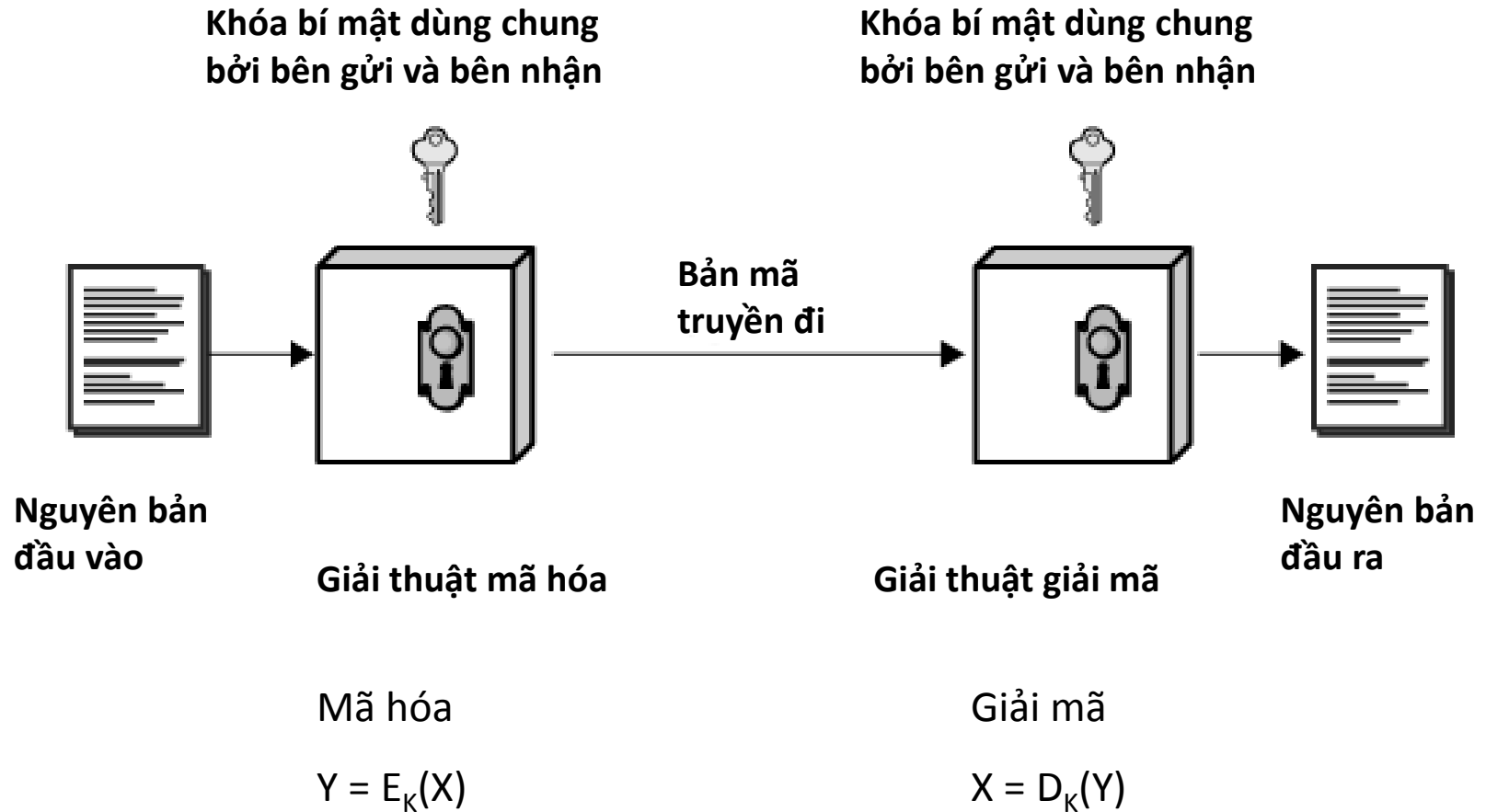
# Giới thiệu

- Nội dung gồm 3 phần:
  - Mô tả các phương pháp bề khóa bảo mật và chỉ ra bảo mật như thế nào là yếu
  - Mã hóa bất đối xứng: phương pháp được ứng dụng nhiều nhất
  - Mã hóa đối xứng: phương pháp bổ sung, kết hợp với các kiểu khác để tăng cường hiệu quả

# Các thuật ngữ

- Plain text (bản rõ): dữ liệu chưa mã hóa
- Cipher text (bản mã): dữ liệu đã được mã hóa
- Key: dữ liệu dùng để mã hóa hoặc giải mã
- Cryptographic algorithm hoặc Cipher: giải thuật mã hóa hoặc giải mã
- Strength: độ khó khi bẻ khóa

# Mô hình hệ mã hóa đối xứng



# Mô hình hệ mã hóa đối xứng

- Gồm có 5 thành phần
  - Bản rõ
  - Giải thuật mã hóa
  - Khóa bí mật
  - Bản mã
  - Giải thuật giải mã
- Độ an toàn phụ thuộc vào bí mật của khóa, không phụ thuộc vào bí mật của giải thuật

# Phá mã

- Là nỗ lực giải mã văn bản đã được mã hóa không biết trước khóa bí mật
- Có hai phương pháp phá mã
  - Vét cạn  
Thử tất cả các khóa có thể
  - Thám mã
    - ✓ Khai thác những nhược điểm của giải thuật
    - ✓ Dựa trên những đặc trưng chung của nguyên bản hoặc một số cặp nguyên bản - bản mã mẫu



# Phương pháp phá mã vét cạn

- Về lý thuyết có thể thử tất cả các giá trị khóa cho đến khi tìm thấy nguyên bản từ bản mã
- Dựa trên giả thiết có thể nhận biết được nguyên bản cần tìm
- Thực tế không khả thi nếu độ dài khóa lớn

# Chuẩn mã hóa dữ liệu

- DES (Data Encryption Standard) được công nhận chuẩn năm 1977
- Phương thức mã hóa được sử dụng rộng rãi nhất
- Tên giải thuật là DEA (Data Encryption Algorithm)
- Kích thước khối: 64 bit
- Kích thước khóa: 56 bit
- Số vòng: 16
- Từng gây nhiều tranh cãi về độ an toàn

# Phá mã DES

- Khóa 56 bit có  $2^{56} = 7,2 \times 10^{16}$  giá trị có thể
- Phương pháp vét cạn tỏ ra không thực tế
- Tốc độ tính toán cao có thể phá được khóa
  - 1997: 70000 máy tính phá mã DES trong 96 ngày
  - 1998: Electronic Frontier Foundation (EFF) phá mã DES bằng máy chuyên dụng (250000\$) trong < 3 ngày
  - 1999: 100000 máy tính phá mã trong 22 giờ
- Vấn đề còn phải nhận biết được nguyên bản
- Nếu cần an toàn hơn: 3DES hay chuẩn mới AES

# Hệ mã hóa 3DES

- Sử dụng 3 khóa và chạy 3 lần giải thuật DES
  - Mã hóa:  $C = E_{K_3}[D_{K_2}[E_{K_1}[p]]]$
  - Giải mã:  $p = D_{K_1}[E_{K_2}[D_{K_3}[C]]]$
- Độ dài khóa thực tế là 168 bit
  - Không tồn tại  $K_4 = 56$  sao cho  $C = E_{K_4}(p)$
- Vì sao 3 lần? tránh tấn công "gặp nhau ở giữa"
  - $C = E_{K_2}(E_{K_1}(p)) \Rightarrow X = E_{K_1}(p) = D_{K_2}(C)$
  - Nếu biết một cặp  $(p, C)$ 
    - ✓ Mã hóa  $p$  với  $2^{56}$  khóa và giải mã  $C$  với  $2^{56}$  khóa
    - ✓ So sánh tìm ra  $K_1$  và  $K_2$  tương ứng
    - ✓ Kiểm tra lại với 1 cặp  $(p, C)$  mới; nếu OK thì  $K_1$  và  $K_2$  là khóa

# Chuẩn mã hóa tiên tiến

- AES (Advanced Encryption Standard) được công nhận chuẩn mới năm 2001
- Tên giải thuật là Rijndael (Rijmen + Daemen)
- An toàn hơn và nhanh hơn 3DES
- Kích thước khối: 128 bit
- Kích thước khóa: 128/192/256 bit
- Số vòng: 10/12/14

# Hệ mã hóa khối khác

## ▪ RC5

- Phát triển bởi Ron Rivest
- Khối 32/64/128 bit, khóa 0-2040 bit, 0-255 vòng
- Đơn giản, thích hợp các bộ xử lý có độ rộng khác nhau

# Các phương thức mã hóa khối

- ECB (Electronic Codebook)

Mã hóa từng khối riêng rẽ

- CBC (Cipher Block Chaining)

Khối nguyên bản hiện thời được XOR với khối bản mã trước đó

- CFB (Cipher Feedback)

Mô phỏng mã hóa luồng (đơn vị  $s$  bit)

$s$  bit mã hóa trước được đưa vào thanh ghi đầu vào hiện thời

- OFB (Output Feedback)

$s$  bit trái đầu ra trước được đưa vào thanh ghi đầu vào hiện thời

- CTR (Counter)

XOR mỗi khối nguyên bản với 1 giá trị thanh đếm mã hóa

# Bố trí công cụ mã hóa

- Giải pháp hữu hiệu và phổ biến nhất chống lại các mối đe dọa đến an ninh mạng là mã hóa
- Để thực hiện mã hóa, cần xác định:
  - Mã hóa những gì
  - Thực hiện mã hóa ở đâu
- Có 2 phương án cơ bản:
  - Mã hóa liên kết
  - Mã hóa đầu cuối



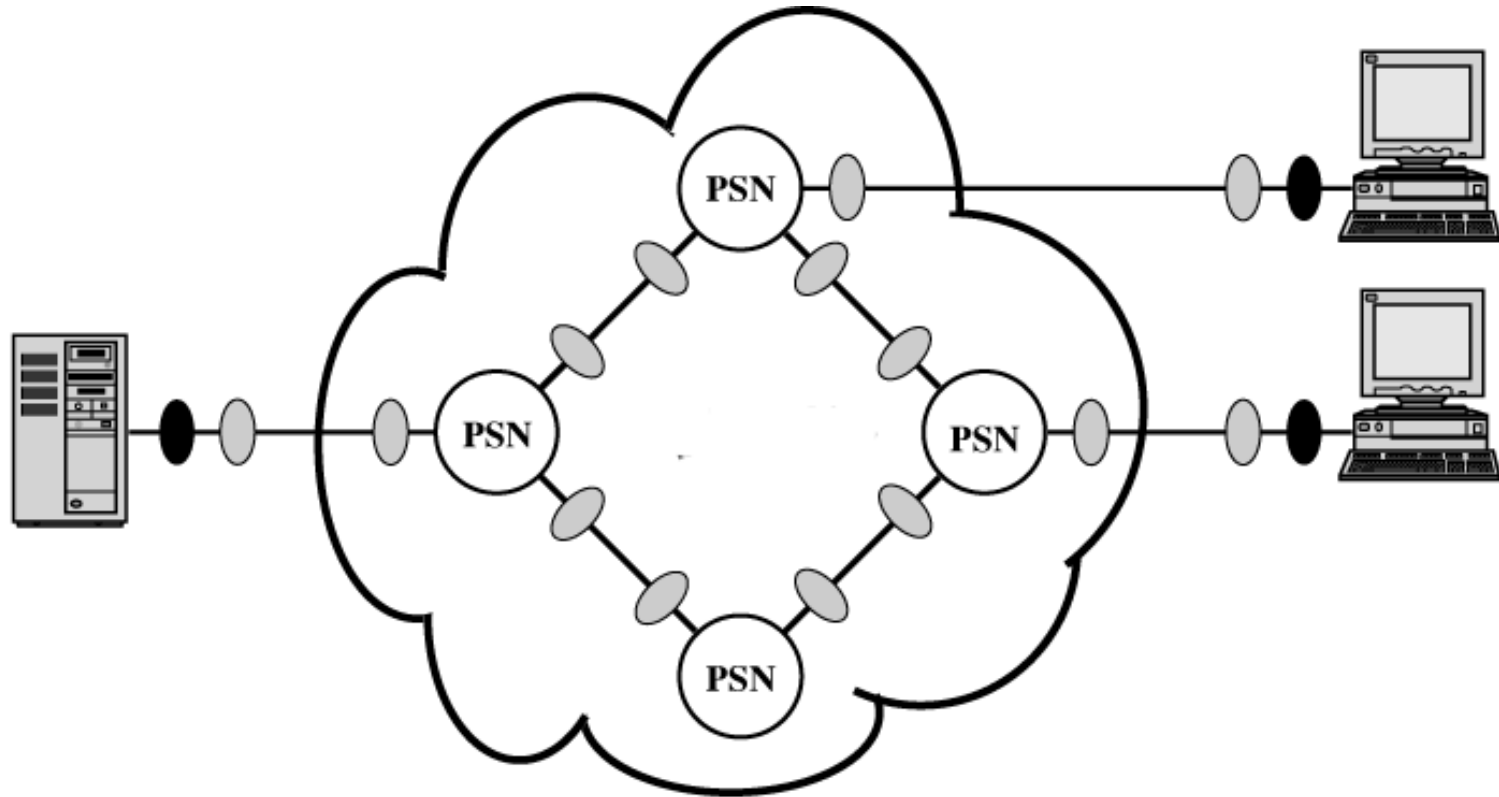
# Mã hóa liên kết

- Công cụ mã hóa được đặt ở 2 đầu của mọi liên kết có nguy cơ bị tấn công
- Đảm bảo an toàn việc chuyển thông tin trên tất cả các liên kết mạng
- Các mạng lớn cần đến rất nhiều công cụ mã hóa
- Cần cung cấp rất nhiều khóa
- Nguy cơ bị tấn công tại mỗi chuyển mạch
  - Các gói tin cần được mã hóa mỗi khi đi vào một chuyển mạch gói để đọc được địa chỉ ở phần đầu
- Thực hiện ở tầng vật lý hoặc tầng liên kết

# Mã hóa đầu cuối

- Quá trình mã hóa được thực hiện ở 2 hệ thống đầu cuối
- Đảm bảo an ninh dữ liệu người dùng
- Chỉ cần một khóa cho 2 đầu cuối

# Kết hợp các phương án mã hóa



- Công cụ mã hóa đầu cuối
- Công cụ mã hóa liên kết

PSN: Packet-switching node

# Quản lý khóa bí mật

- Vấn đề đối với mã hóa đối xứng là làm sao phân phối khóa đến các bên truyền tin

Thường hệ thống mất an ninh là do không quản lý tốt việc phân phối khóa bí mật

- Phân cấp khóa

- Khóa phiên (tạm thời)

- ✓ Dùng mã hóa dữ liệu trong một phiên kết nối

- ✓ Hủy bỏ khi hết phiên

- Khóa chủ (lâu dài)

- Dùng để mã hóa các khóa phiên, đảm bảo phân phối chúng một cách an ninh

# Các cách phân phối khóa

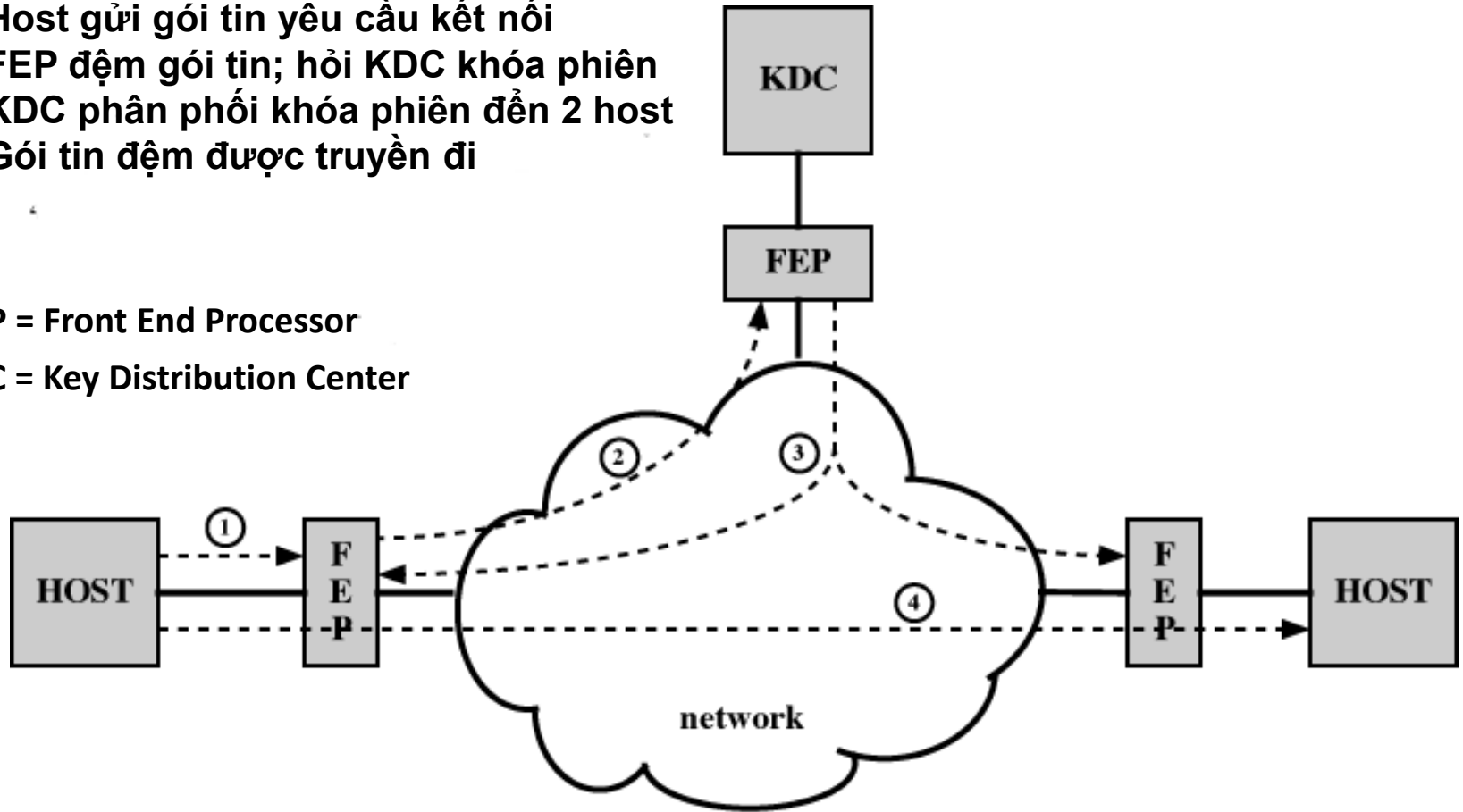
- Khóa có thể được chọn bởi bên A và gửi theo đường vật lý đến bên B
- Khóa có thể được chọn bởi một bên thứ ba, sau đó gửi theo đường vật lý đến A và B
- Nếu A và B đã có một khóa dùng chung thì một bên có thể gửi khóa mới đến bên kia, sử dụng khóa cũ để mã hóa khóa mới
- Nếu mỗi bên A và B đều có một kênh mã hóa đến một bên thứ ba C thì C có thể gửi khóa theo các kênh mã hóa đó đến A và B

# Phân phối khóa tự động

1. Host gửi gói tin yêu cầu kết nối
2. FEP đệm gói tin; hỏi KDC khóa phiên
3. KDC phân phối khóa phiên đến 2 host
4. Gói tin đệm được truyền đi

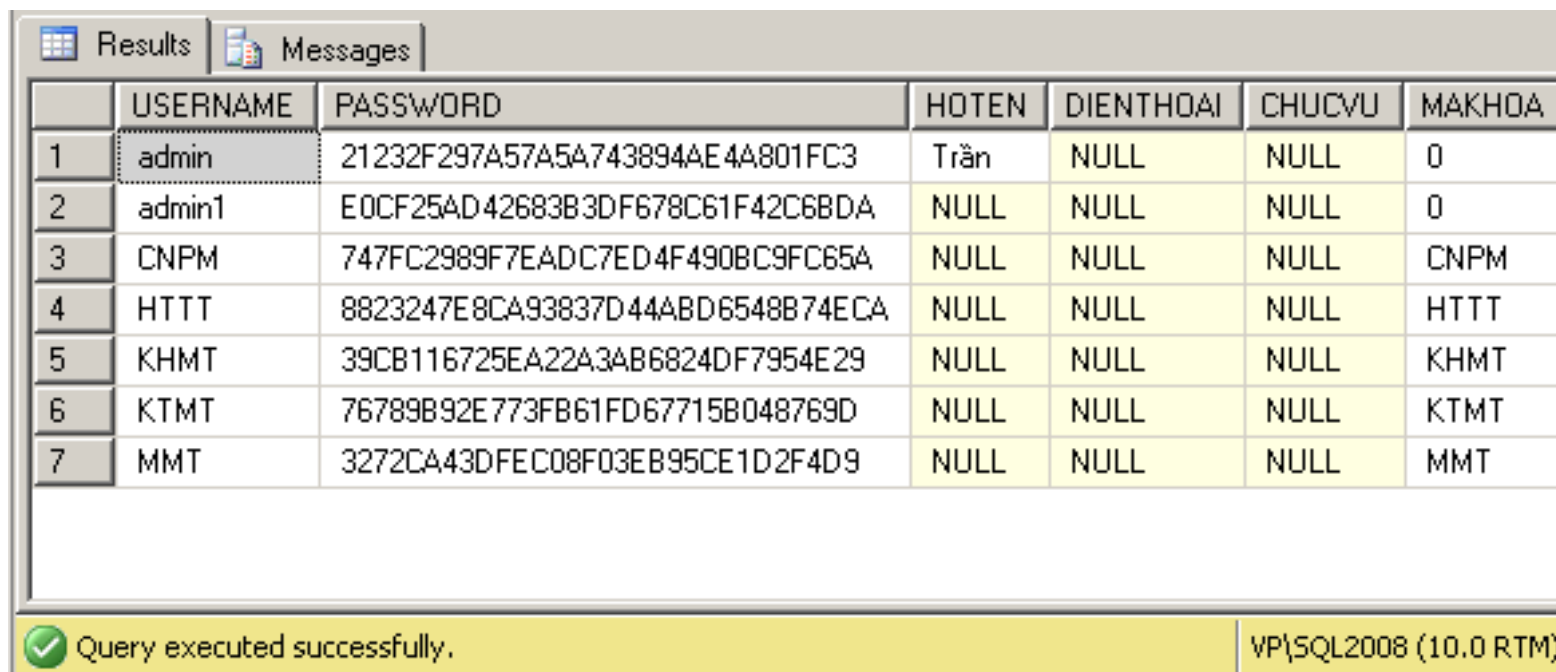
FEP = Front End Processor

KDC = Key Distribution Center



# Minh họa ứng dụng mã hóa

Không nên lưu trữ password thô vào cơ sở dữ liệu, nếu hacker tấn công thành công thì nguy cơ bị chiếm hoàn toàn hệ thống rõ ràng xảy ra



	USERNAME	PASSWORD	HOTEN	DIENHOA	CHUCVU	MAKHOA
1	admin	21232F297A57A5A743894AE4A801FC3	Trần	NULL	NULL	0
2	admin1	E0CF25AD42683B3DF678C61F42C6BDA	NULL	NULL	NULL	0
3	CNPM	747FC2989F7EADC7ED4F490BC9FC65A	NULL	NULL	NULL	CNPM
4	HTTT	8823247E8CA93837D44ABD6548B74ECA	NULL	NULL	NULL	HTTT
5	KHMT	39CB116725EA22A3AB6824DF7954E29	NULL	NULL	NULL	KHMT
6	KTMT	76789B92E773FB61FD67715B048769D	NULL	NULL	NULL	KTMT
7	MMT	3272CA43DFEC08F03EB95CE1D2F4D9	NULL	NULL	NULL	MMT

Query executed successfully. VP\SQL2008 (10.0 RTM)

# Mã hóa bất đối xứng

- Những hạn chế của mật mã đối xứng
  - Vấn đề phân phối khóa
    - ✓ Khó đảm bảo chia sẻ mà không làm lộ khóa bí mật
    - ✓ Trung tâm phân phối khóa có thể bị tấn công
  - Không thích hợp cho chữ ký số
    - ✓ Bên nhận có thể làm giả thông báo nói nhận được từ bên gửi
- Mật mã khóa công khai đề xuất bởi Whitfield Diffie và Martin Hellman vào năm 1976
  - Khắc phục những hạn chế của mật mã đối xứng
  - Có thể coi là bước đột phá quan trọng nhất trong lịch sử của ngành mật mã
  - Bổ sung chứ không thay thế mật mã đối xứng

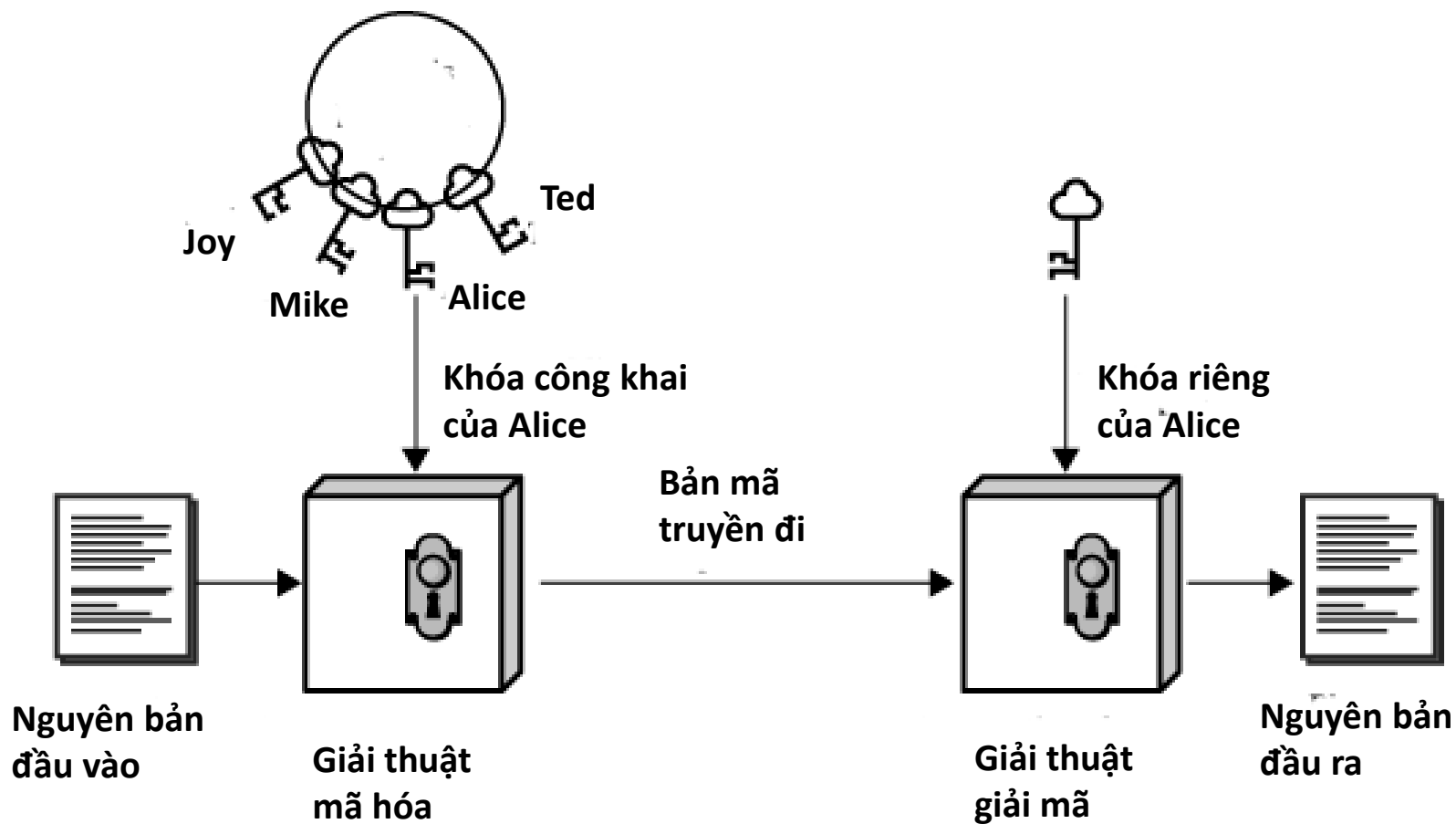


# Đặc điểm mật mã khóa công khai

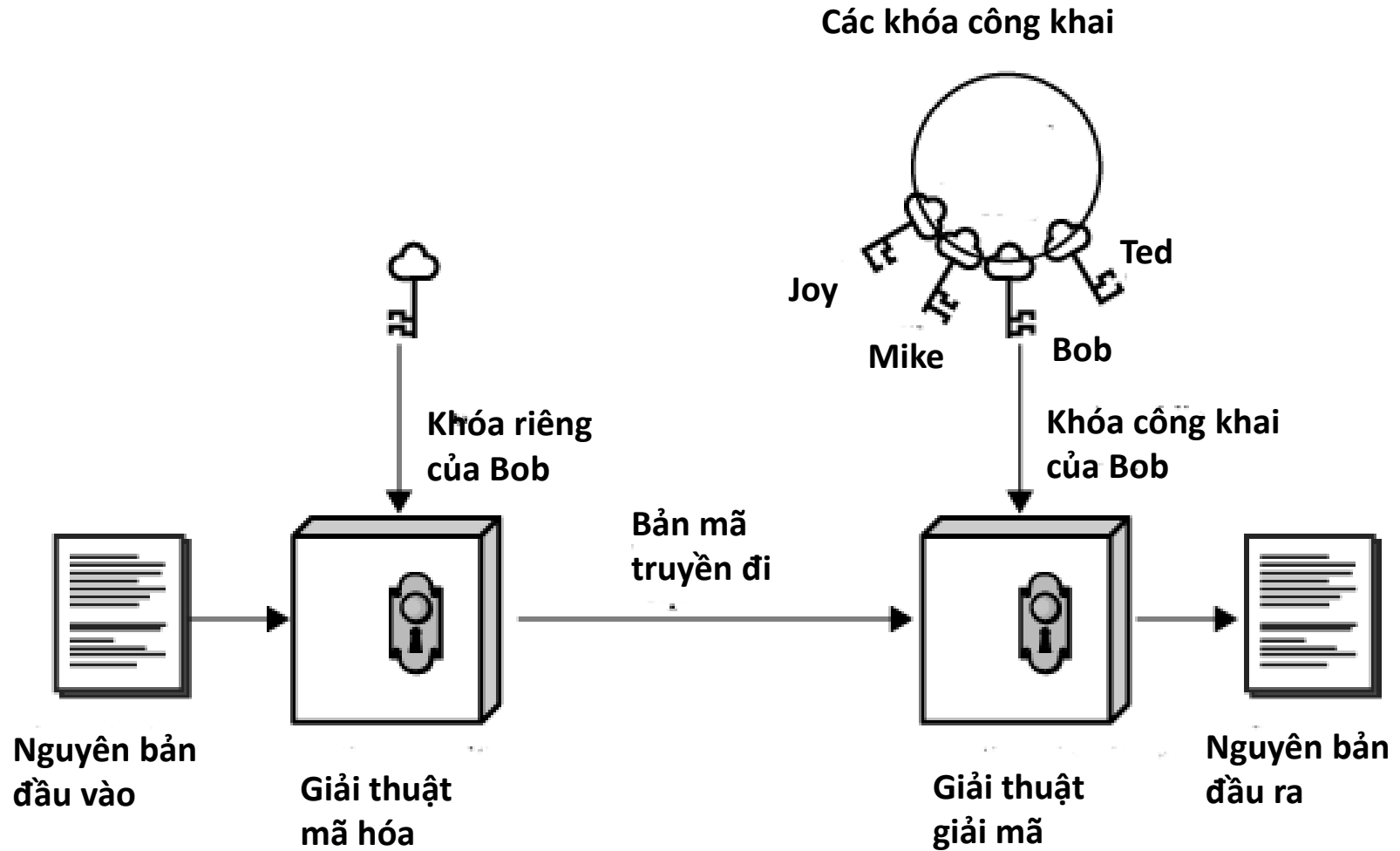
- Còn gọi là mật mã hai khóa hay bất đối xứng
- Các giải thuật khóa công khai sử dụng 2 khóa
  - Một khóa công khai
    - ✓ Ai cũng có thể biết
    - ✓ Dùng để mã hóa thông báo và thẩm tra chữ ký
  - Một khóa riêng
    - ✓ Chỉ nơi giữ được biết
    - ✓ Dùng để giải mã thông báo và ký (tạo ra) chữ ký
- Có tính bất đối xứng
  - Bên mã hóa không thể giải mã thông báo
  - Bên thẩm tra không thể tạo chữ ký

# Mã hóa khóa công khai

Các khóa công khai



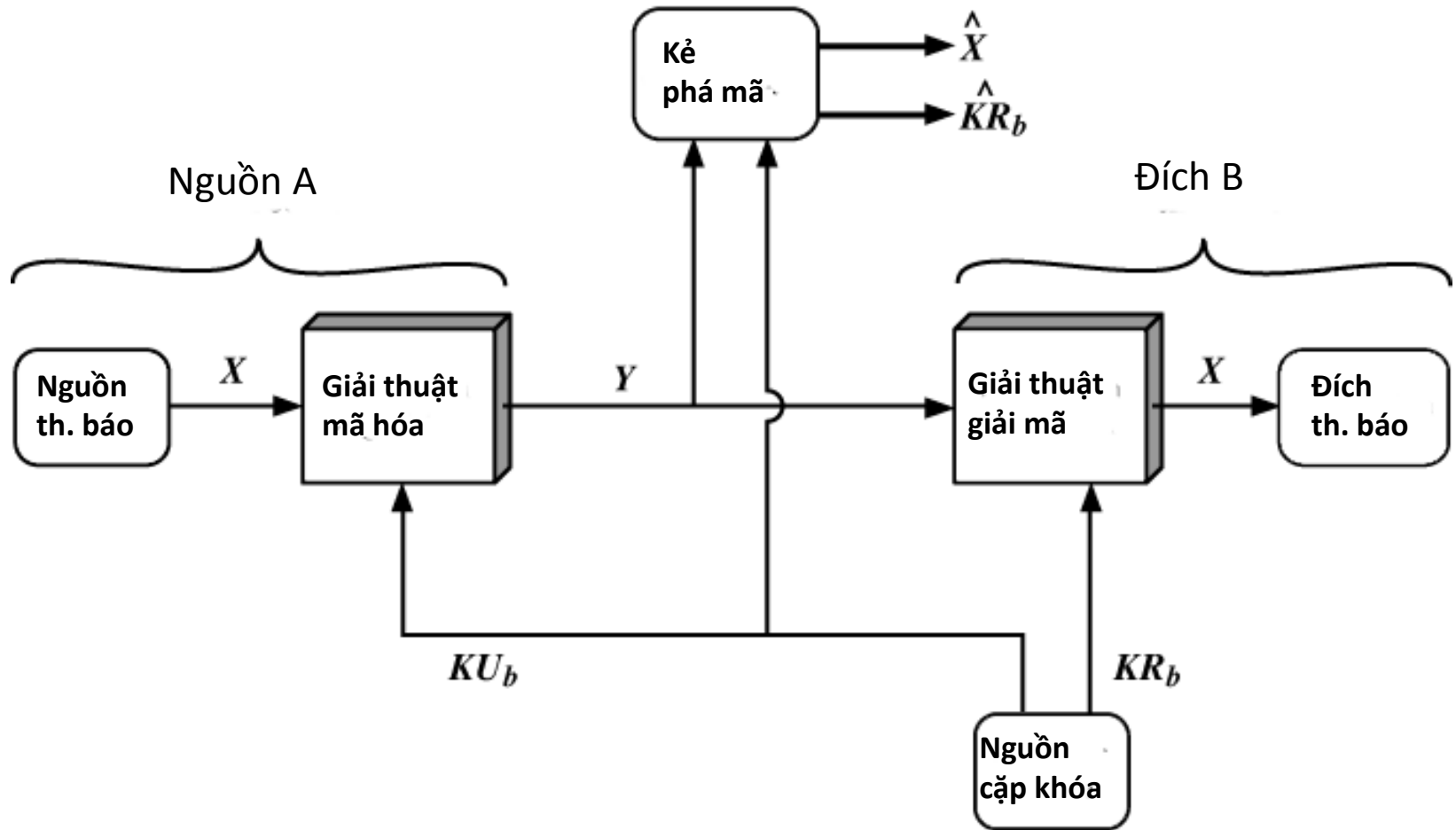
# Xác thực



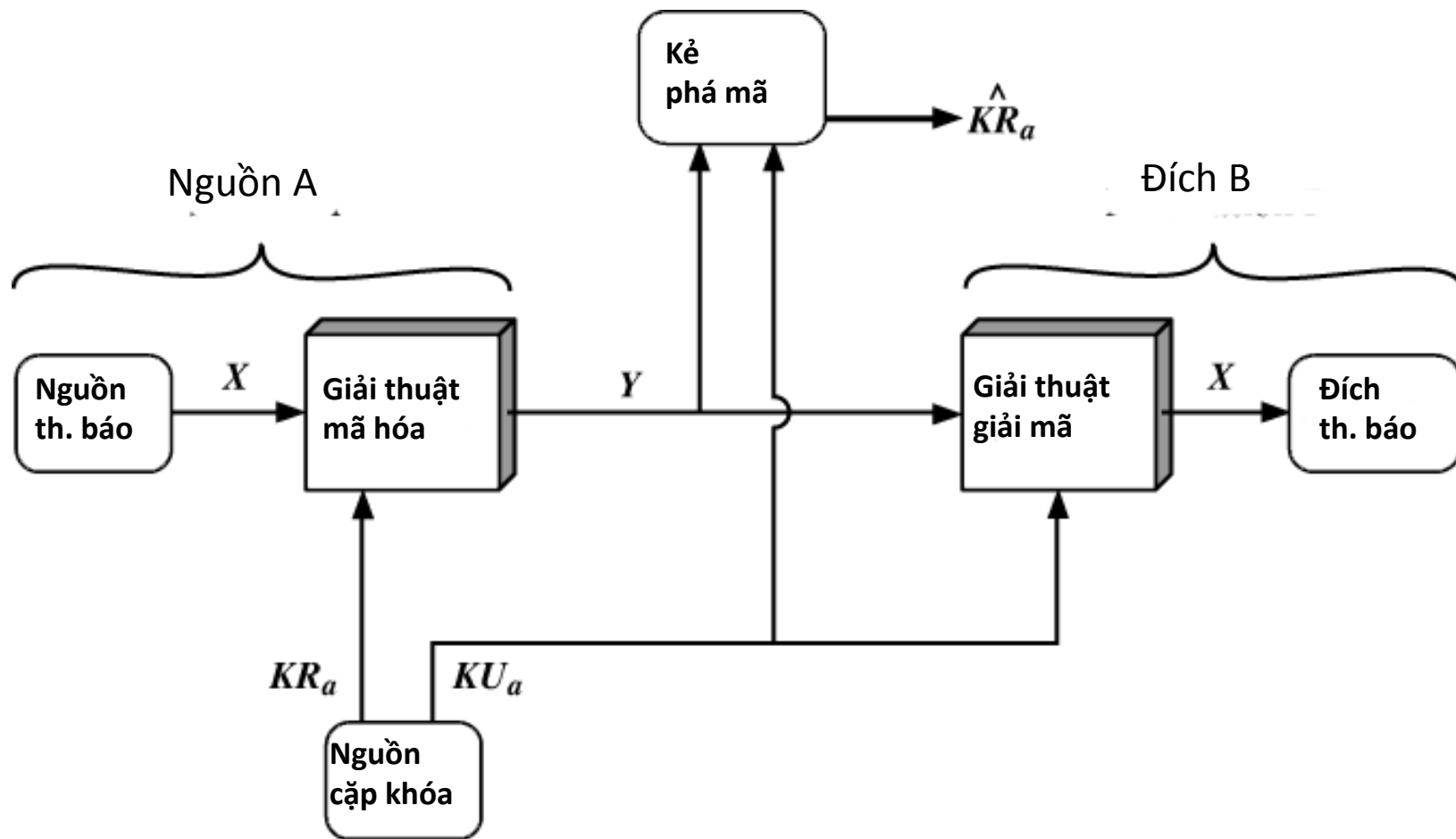
# Ứng dụng mật mã khóa công khai

- Có thể phân ra 3 loại ứng dụng
  - Mã hóa/giải mã  
Đảm bảo sự bí mật của thông tin
  - Chữ ký số  
Hỗ trợ xác thực văn bản
  - Trao đổi khóa  
Cho phép chia sẻ khóa phiên trong mã hóa đối xứng
- Một số giải thuật khóa công khai thích hợp cho cả 3 loại ứng dụng; một số khác chỉ có thể dùng cho 1 hay 2 loại

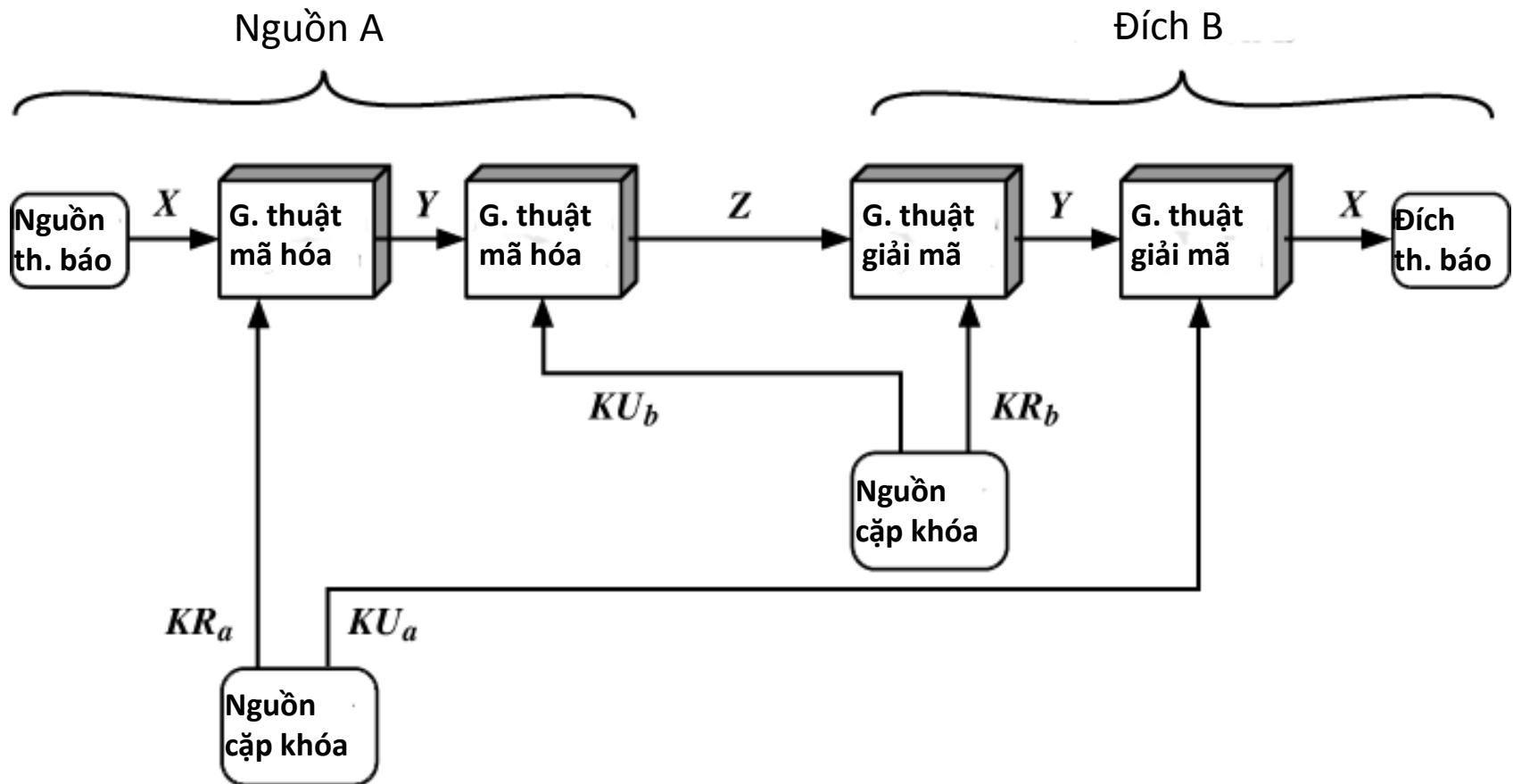
# Mô hình đảm bảo bí mật



# Mô hình xác thực



# Mô hình kết hợp



# Hệ mã hóa RSA

- Đề xuất bởi Ron Rivest, Adi Shamir và Len Adleman (MIT) vào năm 1977
- Hệ mã hóa khóa công khai thông dụng nhất
- Mã hóa khối với mỗi khối là một số nguyên  $< n$   
Thường kích cỡ  $n$  là 1024 bit  $\approx$  309 chữ số thập phân
- An toàn vì chi phí phân tích thừa số của một số nguyên lớn là rất lớn



# Sự hỗ trợ giữ 2 hệ thống mã hóa

Giao thức bảo mật sử dụng cả 2:

- Đối xứng: bảo vệ trao đổi dữ liệu qua mạng, tốc độ xử lý nhanh
- Bất đối xứng: thành lập kết nối giữa 2 thực thể mạng và thành lập khóa đối xứng.

# An ninh của RSA

- Khóa 128 bit là một số giữa 1 và một số rất lớn  
340.282.366.920.938.000.000.000.000.000.000.000.000
- Có bao nhiêu số nguyên tố giữa 1 và số này  
 $\approx n / \ln(n) = 2^{128} / \ln(2^{128}) \approx$   
3.835.341.275.459.350.000.000.000.000.000.000.000
- Cần bao nhiêu thời gian nếu mỗi giây có thể tính được  $10^{12}$  số  
Hơn 121.617.874.031.562.000 năm (khoảng 10 triệu lần tuổi của vũ trụ)

# Phá mã RSA

- Phương pháp vét cạn

  - Thử tất cả các khóa riêng có thể

  - Phụ thuộc vào độ dài khóa

- Phương pháp phân tích toán học

  - Phân  $n$  thành tích 2 số nguyên tố  $p$  và  $q$
  - Xác định trực tiếp  $\Phi(n)$  không thông qua  $p$  và  $q$
  - Xác định trực tiếp  $d$  không thông qua  $\Phi(n)$

- Phương pháp phân tích thời gian

  - Dựa trên việc đo thời gian giải mã
  - Có thể ngăn ngừa bằng cách làm nhiều

# Phân tích thừa số RSA

- An toàn của RSA dựa trên độ phức tạp của việc phân tích thừa số  $n$
- Thời gian cần thiết để phân tích thừa số một số lớn tăng theo hàm mũ với số bit của số đó  
Mất nhiều năm khi số chữ số thập phân của  $n$  vượt quá 100 (giả sử làm 1 phép tính nhị phân mất 1  $\mu$ s)
- Kích thước khóa lớn đảm bảo an toàn cho RSA
  - Từ 1024 bit trở lên
  - Gần đây nhất năm 1999 đã phá mã được 512 bit (155 chữ số thập phân)

# Bảo vệ tác quyền phần mềm

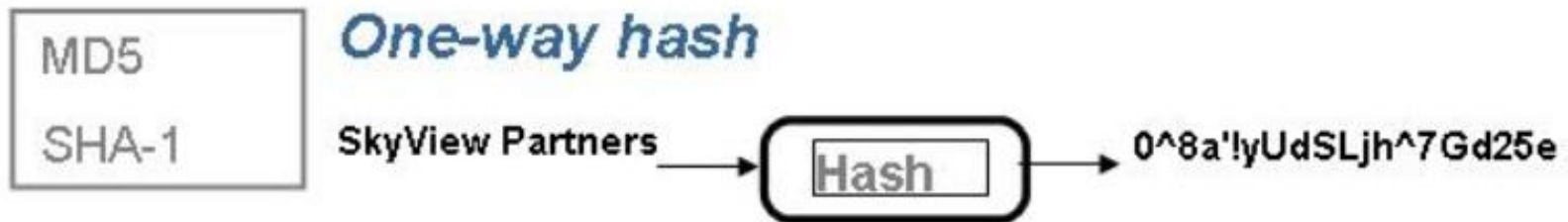
- Giữ key trên mạng cục bộ hoặc server trung tâm để bảo đảm key duy nhất
- Kẻ tấn công rất khó khăn tìm được key hợp lệ thứ hai
- Phương pháp khác là giả sử phần mềm sinh ra một số ngẫu nhiên lớn  $n$  tại thời điểm mua

# Bảo vệ tác quyền phần mềm

- Hacker có thể dùng phần mềm tự động lặp hàng triệu lần các tổ hợp key để giả lập người dùng nhập thông tin vào cửa sổ “enter license key”
- Vì vậy phần mềm của chúng ta nên kết thúc nếu sau 3 lần thử bị sai và xóa chính nó nếu sai 100 lần.

# Hash functions

- Băm từng chuỗi bit một
- Không thể khôi phục dữ liệu ban đầu từ cipher text
- 1 bit khác nhau sẽ tạo ra ít nhất nửa số bit kết quả khác nhau



# Hash functions

- MD (Message Digest) functions  
MD2, MD4, MD5: 16 bytes fingerprint
- SHA-1 (Secure Hash Algorithm-1): 20 bytes
- SHA-256: 256 bytes
- SHA-224: 224 bytes
- SHA-512: 512 bytes

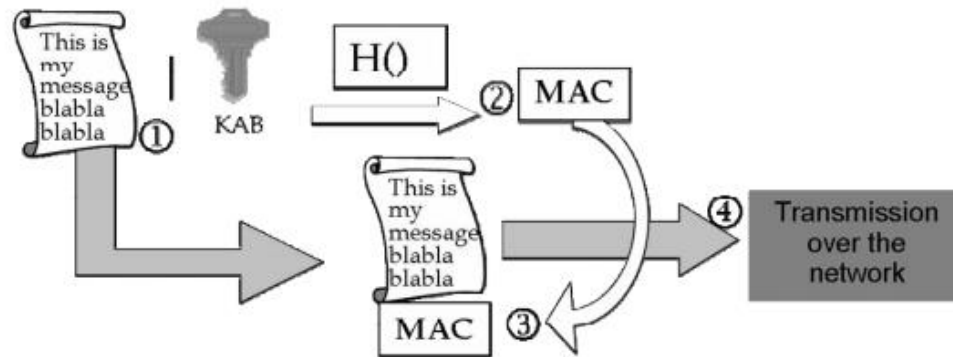


# Chữ ký điện tử

- MAC (Message Authentication Code), chữ ký điện tử có hai mục tiêu:
  - Kiểm tra tính nguyên thủy (nguồn gốc)
  - Kiểm tra toàn vẹn của dữ liệu
- Sử dụng hash functions, symmetric hoặc asymmetric keys

# Chữ ký điện tử

Source A



Receiver B

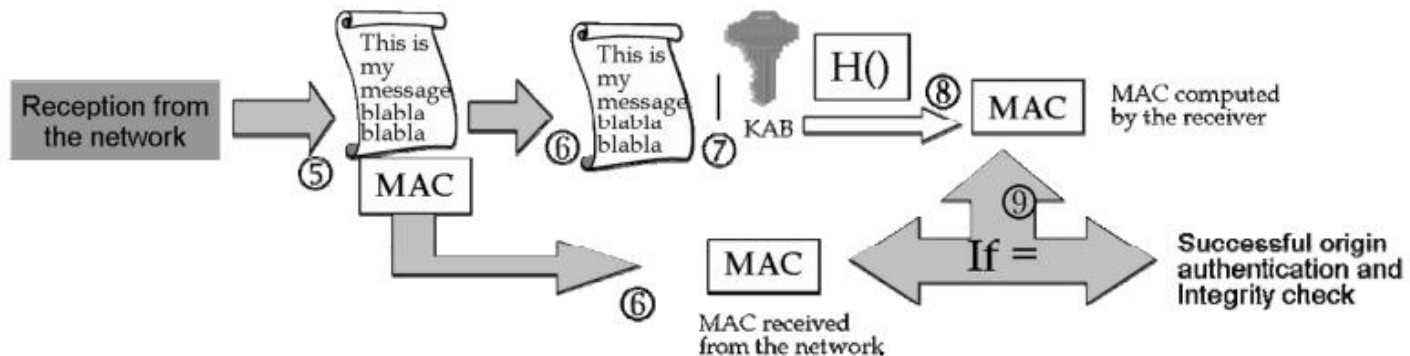
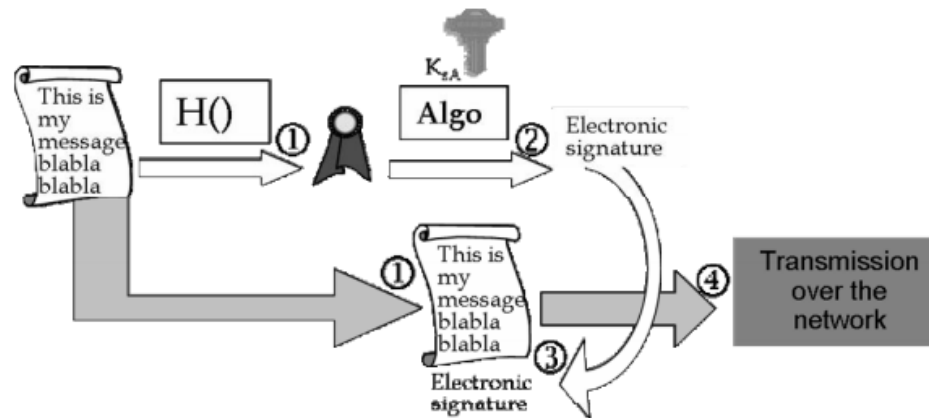


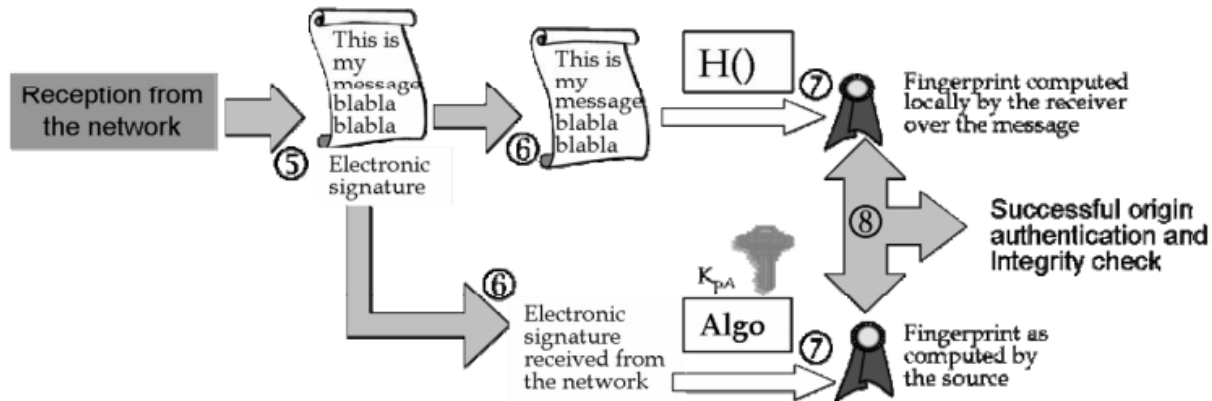
Figure 3.1. Generation and verification of a MAC (symmetric cryptography)

# Chữ ký điện tử

Source A



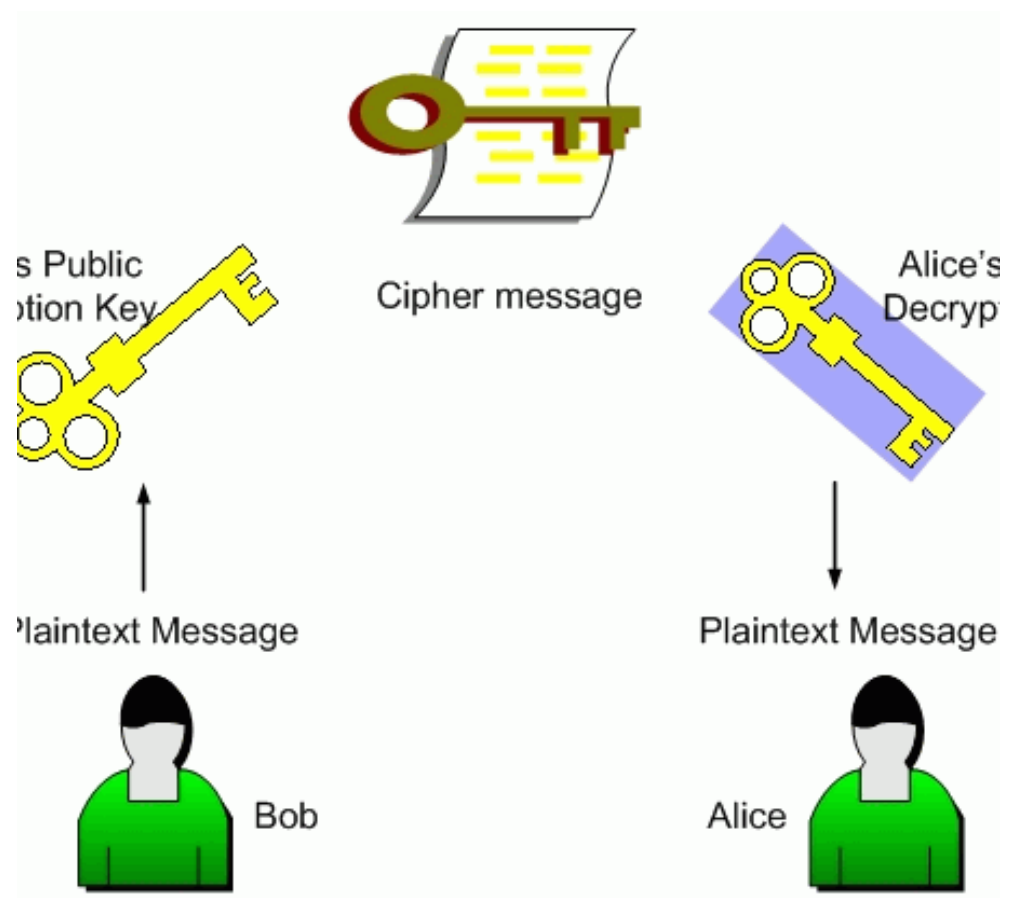
Receiver B



**Figure 3.2.** Generation and verification of an electronic signature (asymmetric cryptography)

# PKI

- Public Key Infrastructure
- Được sử dụng nhiều:  
S/MIME, SSL/TLS, IPsec  
và SSH.



# PKI

PKI phân biệt 2 vai trò:

- Certification Authority (CA):
  - Giữ private key
  - Phát hành và hủy chứng thực số
- Registration Authority:
  - Đăng kí với CA
  - Xử lý yêu cầu chứng thực từ người dùng