# BÀI 2.
# CÁC HỆ MẬT MÃ (NHẮC LẠI)

Bùi Trọng Tùng,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội



## Nội dung

- Mật mã (cipher) là gì?
- Nguyên tắc chung của các hệ mật mã
- Hệ mật mã khóa đối xứng
- Hệ mật mã khóa bất đối xứng

# 1. MẬT MÃ LÀ GÌ?

Bùi Trọng Tùng,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội



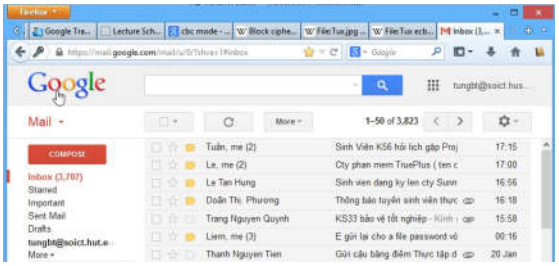
## 1.1. Khái niệm mật mã

- Mã hóa (code): biến đổi cách thức biểu diễn thông tin
- Mật mã (cipher): mã hóa để che giấu, giữ mật thông tin
  - Lưu trữ
  - Truyền tin
- Mật mã học (cryptography): ngành khoa học nghiên cứu các phương pháp toán học để mã hóa giữ mật thông tin
- Thám mã (cryptoanalysis): nghiên cứu các phương pháp toán học để phá vỡ hệ mật mã
- Là công cụ hiệu quả giải quyết bài toán ATBM, là cơ sở cho nhiều cơ chế khác (xác thực, nhận dạng)
  - ✓Nhưng không vạn năng

# Truyền tin bí mật

- Bước 1: Trao đổi khóa
- Bước 2: Mã hóa dữ liệu






5

# Lưu trữ thông tin mật


Thiết bị lưu trữ

Alice “hôm nay” truyền tin bí mật cho Alice “ngày mai”

6

## Xây dựng mô hình (mật mã khóa đối xứng)

- Alice và Bob đã chia sẻ thông tin bí mật K gọi là khóa
- Alice cần gửi cho Bob một thông điệp M (bản rõ). Nội dung thông điệp cần giữ bí mật trước quan sát của Eve (kẻ tấn công, thám mã)

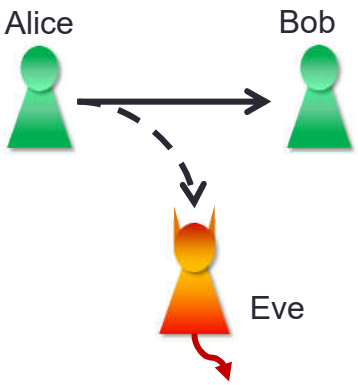
Mã hóa: C = E(K, M)

C: bản mã

- Alice gửi bản mã lên kênh truyền. Bob và Eve đều thu được thông điệp này. Chỉ có Bob giải mã để thu được bản rõ

Giải mã: M = D(K, C)

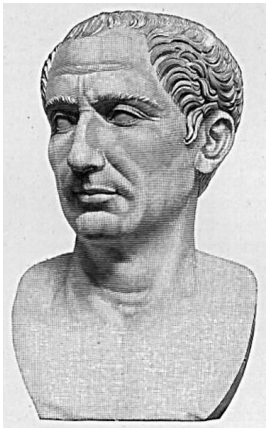
- Mật mã khóa đối xứng: dùng khóa K trong cả hai quá trình mã hóa và giải mã




7

## Một ví dụ - Mật mã Caesar

- Julius Caesar đưa ra vào thế kỷ thứ 1 trước CN, sử dụng trong quan sự
- Ý tưởng: thay thế một ký tự (bản rõ) trong bảng chữ cái bằng ký tự (bản mật) đứng sau nó 3 (khóa) vị trí.
  - Sử dụng bảng chữ cái vòng
  - A → D, B → E, C → F,..., X → A, Y → B, Z → C
- Mô hình hóa bằng toán học:
  - Khóa K = 3
  - Mã hóa: C = (M+3) mod 26
  - Giải mã: M = (C − 3) mod 26
- Dễ dàng bị phá ngay cả khi K thay đổi các giá trị khác



*Gaius Julius Caesar*

8

## 1.2. Một số nguyên lý chung của các hệ mật mã

- Định luật Kerckhoffs: “Một hệ mật mã cần an toàn ngay cả khi mọi thông tin về hệ, trừ khóa bí mật, là công khai”
- Tại sao?

9

## Lý thuyết Shannon

- Hệ mật hoàn hảo: độ an toàn của hệ mật không phụ thuộc vào số bản mã và thời gian kẻ tấn công sử dụng để thám mã (An toàn vô điều kiện)
- Lý thuyết Shannon: Một hệ mật mã là hoàn hảo thì
  - Độ dài của khóa tối thiểu bằng độ dài bản tin rõ
  - Khóa chỉ sử dụng một lần
  - → Tại sao khó đạt được trên thực tế?

Điều kiện cần

- An toàn theo tính toán: thỏa mãn đồng thời 2 điều kiện
  - Thời gian để thám mã thành công lớn hơn thời gian cần giữ mật thông tin
  - Chi phí để thám mã thành công lớn hơn giá trị thông tin thu được

10

# Lý thuyết Shannon (tiếp)

- Độ dư thừa của ngôn ngữ: Sự xuất hiện của n ký tự (n-gram) cho phép đoán nhận đúng các ký tự xuất hiện tiếp theo với xác xuất *p* nào đó.
  - Nếu *p = 0 ∀n:* ngôn ngữ không có dư thừa
  - Nếu *p > 0*: ngôn ngữ có dư thừa (một số ký tự là không cần thiết sau khi *n ký tự đã xuất hiện*)
  - Định lượng: sử dụng lý thuyết thông tin
  - Ví dụ: tiếng Việt
- Đối với thám mã: sử dụng phương pháp vét cạn, cần phải thu được tối thiểu *u* ký tự mật mã để tìm được chính xác khóa.

*u*: khoảng cách unicity (unicity distance)

→ *u* càng lớn độ an toàn của hệ càng cao

11

# Lý thuyết Shannon (tiếp)

- Tính toán khoảng cách unicity

$$u = \frac{l_K H(k)}{H(c) - H(m)}$$

$l_K$: Kích thước khóa

$H(k), H(m), H(c)$: entropy của ký tự. Ví dụ

$H(m) = -\sum p(m_i) \times log_2(p(m_i))$: entropy của ký tự bản rõ

$p(m_i)$: xác suất xuất hiện của ký tự trong không gian bản rõ

- Nếu khóa và bản mật xuất hiện hoàn toàn ngẫu nhiên, và chung bảng chữ cái:

$$u = \frac{l_K log_2(N)}{log_2(N) - H(m)}$$

N: số ký tự của bảng chữ cái

- Làm thế nào để tăng độ an toàn khi sử dụng mật mã?

12

## Thông tin tham khảo – Kích thước khóa

- Khóa có kích thước bao nhiêu?
  - Mật mã được coi là an toàn khi phương pháp vét cạn (brute-force) là cách nhanh nhất để bẻ khóa
  - Mục tiêu: giảm thiểu nguy cơ bị tấn công vét cạn (đạt độ an toàn theo tính toán)
- Bạn nghe ở đâu đó, "dễ dàng" bẻ khóa mật mã DES có kích thước khóa 64 bit?
  - Năm 1999, hệ thống phá mã EFF DES (trị giá 250K$) bẻ khóa DES trong khoảng 1 ngày
  - Năm 2008, hệ thống phá mã COPACOBANA (trị giá 10K$) bẻ khóa DES trong 6,4 ngày

  Sử dụng định luật Moore để tính thời gian bẻ khóa trong năm 2016 với chi phí 10K$?

13

## Thông tin tham khảo – Kích thước khóa

- Chi phí để bẻ khóa DES (năm 2008)
  - 64 bit: $10.000
  - 87 bit: $10.000.000.000 (thời gian bẻ khóa không đổi)
- Cần giữ thông tin mật trong bao lâu khi hệ thống phá mã là COPACOBANA? (năm 2008)
  - 64 bit: 6.4 ngày
  - 128 bit: ?
- Tuy nhiên, vét cạn là phương pháp tấn công tầm thường.
- Tham khảo kích thước khóa nên sử dụng trong tương lai tại địa chỉ
  http://csrc.nist.gov/groups/ST/toolkit/key_management.html

14

# Thông tin tham khảo – Kích thước khóa

| Date | Minimum of Strength | Symmetric Algorithms | Factoring Modulus | Discrete Logarithm Key | Discrete Logarithm Group | Elliptic Curve | Hash (A) | Hash (B) |
|---|---|---|---|---|---|---|---|---|
| 2010 (Legacy) | 80 | 2TDEA* | 1024 | 160 | 1024 | 160 | SHA-1**<br>SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 | SHA-1<br>SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 |
| 2011 - 2030 | 112 | 3TDEA | 2048 | 224 | 2048 | 224 | SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 | SHA-1<br>SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 |
| > 2030 | 128 | AES-128 | 3072 | 256 | 3072 | 256 | SHA-256<br>SHA-384<br>SHA-512 | SHA-1<br>SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 |
| >> 2030 | 192 | AES-192 | 7680 | 384 | 7680 | 384 | SHA-384<br>SHA-512 | SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 |
| >>> 2030 | 256 | AES-256 | 15360 | 512 | 15360 | 512 | SHA-512 | SHA-256<br>SHA-384<br>SHA-512 |

http://www.keylength.com

15

# Thông tin tham khảo – Thời hạn khóa

| Key Type<br>*Move the cursor over a type for description* | Cryptoperiod<br>Originator Usage Period (OUP) | Recipient Usage Period |
|---|---|---|
| Private Signature Key | 1-3 years | |
| Public Signature Key | Several years (depends on key size) | |
| Symmetric Authentication Key | <= 2 years | <= OUP + 3 years |
| Private Authentication Key | 1-2 years | |
| Public Authentication Key | 1-2 years | |
| Symmetric Data Encryption Key | <= 2 years | <= OUP + 3 years |
| Symmetric Key Wrapping Key | <= 2 years | <= OUP + 3 years |
| Symmetric and asymmetric RNG Keys | Upon reseeding | |
| Symmetric Master Key | About 1 year | |
| Private Key Transport Key | <= 2 years (1) | |
| Public Key Transport Key | 1-2 years | |
| Symmetric Key Agreement Key | 1-2 years | |
| Private Static Key Agreement Key | 1-2 years (2) | |
| Public Static Key Agreement Key | 1-2 years | |
| Private Ephemeral Key Agreement Key | One key agreement transaction | |
| Public Ephemeral Key Agreement Key | One key agreement transaction | |
| Symmetric Authorization Key | <= 2 years | |
| Private Authorization Key | <= 2 years | |
| Public Authorization Key | <= 2 years | |

16

# 2. Hệ mật mã khóa đối xứng

- Symmetric cryptography, Secret-key cryptography: sử dụng cùng một khóa khi mã hóa và giải mã.
- Được phát triển từ rất sớm
- Thuật toán mã hóa: phối hợp các toán tử
  - Thay thế
  - Đổi chỗ
  - XOR ⊕ (Tại sao dùng XOR?)
- Tốc độ thực hiện các thuật toán nhanh, có thể thực hiện bằng dễ dàng bằng phần cứng
- Một số hệ mật mã khóa đối xứng hiện đại: DES, 2DES, 3DES, AES, RC4, RC5

Việc tìm hiểu đặc điểm những hệ mật mã này
coi như một bài tập!!!

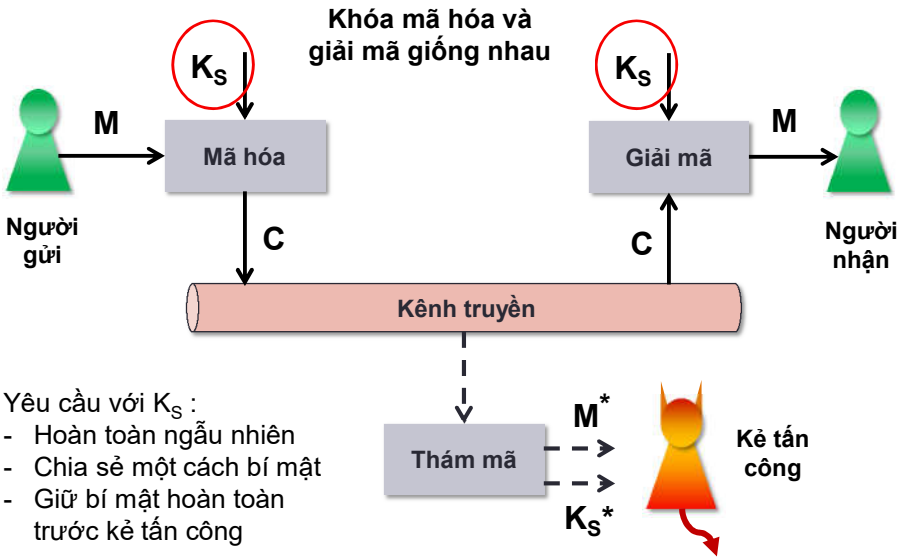
17

# 2.1. Sơ đồ chung

- Hệ mật mã gồm:
  - Bản rõ (plaintext-M): bản tin được sinh ra bởi bên gửi
  - Bản mật (ciphertext-C): bản tin che giấu thông tin của bản rõ, được gửi tới bên nhận qua một kênh không bí mật
  - Khóa (key-$K_S$): giá trị ngẫu nhiên và bí mật được chia sẻ giữa các bên trao đổi thông tin
    - ✓Bên thứ 3 được tin cậy tạo và phân phối tới bên gửi và bên nhận
    - ✓Hoặc, bên nguồn tạo và chuyển cho bên nhận
  - Mã hóa (encrypt-E): $C = E(K_S, M)$
  - Giải mã (decrypt): $M = D(K_S, C) = D(K_S, E(K_S, M))$

18

# Sơ đồ chung




Yêu cầu với $K_S$ :
- Hoàn toàn ngẫu nhiên
- Chia sẻ một cách bí mật
- Giữ bí mật hoàn toàn trước kẻ tấn công

19

# Thám mã

- Nhắc lại định luật Kerckhoffs "Một hệ mật mã cần an toàn ngay cả khi mọi thông tin về hệ, trừ khóa bí mật, là công khai"
  - Kẻ thám mã đã biết giải thuật mã hóa, giải mã
- Tấn công chỉ biết bản mật:
  - Kẻ thám mã có thêm các bản mật C (ciphertext-only attack)
  - Phương pháp phá mã: thử tất cả các tổ hợp khóa có thể để tìm ra tổ hợp khóa thích hợp. Trong trường hợp không gian khóa lớn thì phương pháp này không thực hiện được.
  - Đối phương cần phải phân tích văn bản mật, thực hiện các kiểm nghiệm thống kê để giảm số lượng trường hợp cần thử.

20

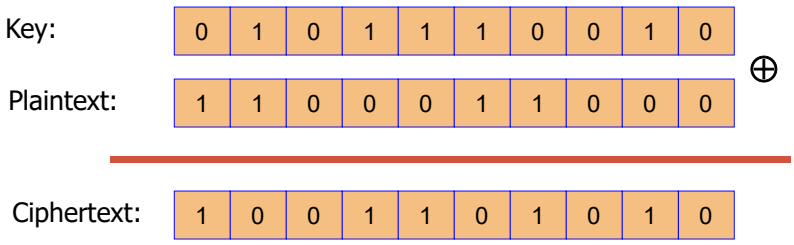
# Thám mã (tiếp)

- Tấn công đã biết bản rõ (known-plaintext attack):
  - Kẻ thám mã đã có một số cặp (M,C) của những phiên truyền tin trước đó. Mục đích: đoán khóa mật K.
  - Phương pháp tấn công: phân tích thuộc tính thống kê của ngôn ngữ trên văn bản gốc
- Tấn công chọn trước bản rõ (chosen-plaintext attack): kẻ thám mã lừa người gửi mã hóa một số bản tin đặc biệt do hắn chọn
- Tấn công chọn trước bản mật(chosen-ciphertext attack): kẻ thám mã lừa người nhận giải mã một số bản tin đặc biệt do hắn chọn
- Tấn công chọn trước bản rõ, bản mật
  - Thuật toán được thiết kế để chống lại dạng tấn công này



# Mật mã one-time-pad

- Vernam (1917)




- Kích thước của khóa bằng kích thước của bản rõ
- Khóa chỉ dùng 1 lần
- Shannon : mật mã one-time-pad là hệ mật hoàn hảo

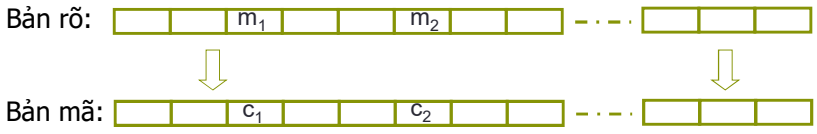
## 2.2. Mật mã dòng (stream cipher)

- Xử lý văn bản rõ theo dòng byte, thời gian thực
  - RC4 (900 Mbps), SEAL (2400 Mbps), RC5(450 Mbps)
- Phù hợp với các hệ thống truyền dữ liệu thời gian thực trên môi trường mạng máy tính
- An toàn nếu khóa hoàn toàn ngẫu nhiên, chỉ dùng 1 lần (one-time-pad)
  - Trên thực tế: khóa giả ngẫu nhiên dùng nhiều lần (n-time-pad)
  - → một số phương pháp mật mã dòng không còn an toàn (RC4)

23

## 2.3. Mật mã khối

- Chia văn bản gốc thành các khối có kích thước như nhau
- Xử lý mã hóa và giải mã từng khối độc lập
- ECB - Electronic Code Book (Chế độ mã từ điển)

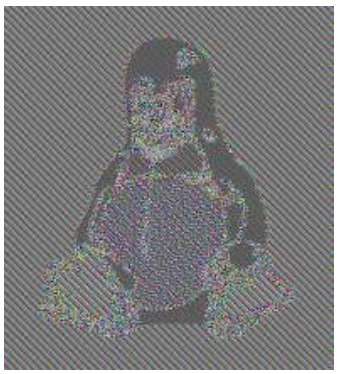



- Hạn chế?

24

# Ví dụ về mật mã khối chế độ ECB

**Ảnh gốc**

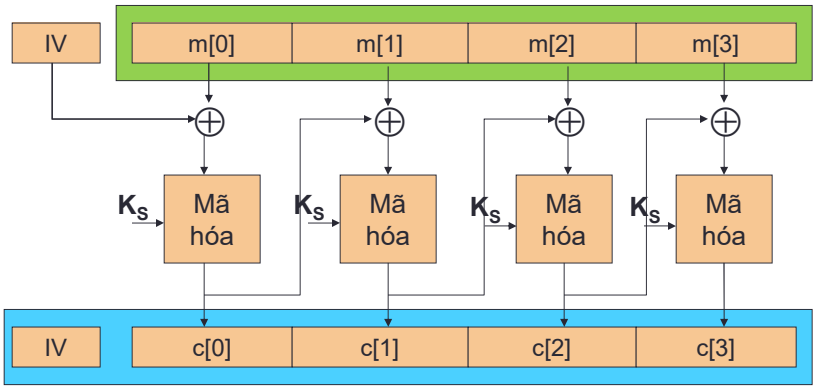


**Ảnh được mã hóa ở chế độ ECB**

25

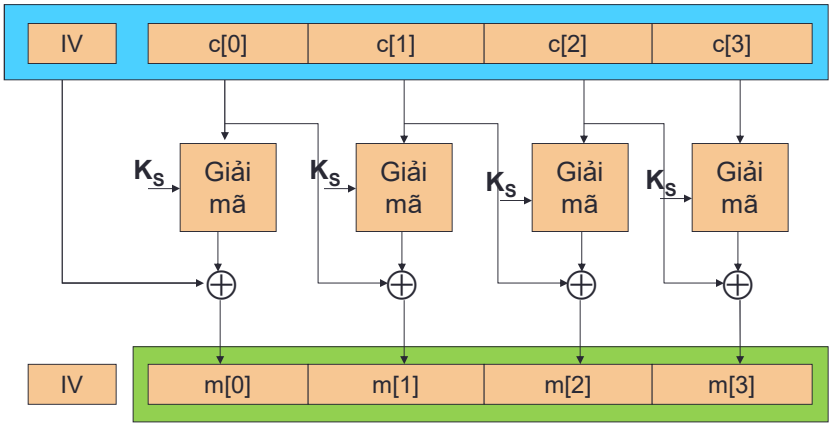
# Chế độ CBC - Cipher Block Chaining

- Chế độ mã móc xích
- Có thể thay thế mã dòng: Ưu điểm? Hạn chế?



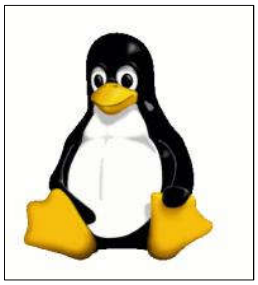

26

# CBC – Giải mã

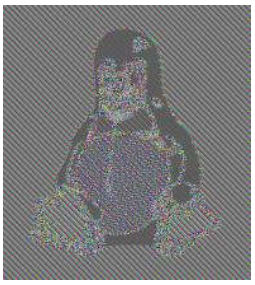



27

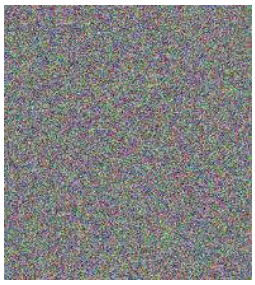
# CBC – So sánh với ECB



**Ảnh gốc**



**Mã hóa ở chế độ ECB**



**Mã hóa ở chế độ CBC**

28

## 2.4. Những hạn chế của mật mã khóa đối xứng

- Cần kênh mật để chia sẻ khóa bí mật giữa các bên
  - Làm sao để chia sẻ một cách an toàn cho lần đầu tiên
- Số lượng khóa lớn: n(n-1)/2
- Khó ứng dụng trong các hệ thống mở (E-commerce)
- Không dễ dàng để xác thực đối với thông tin quảng bá (Chúng ta sẽ quay trở lại vấn đề này trong những bài sau)

29

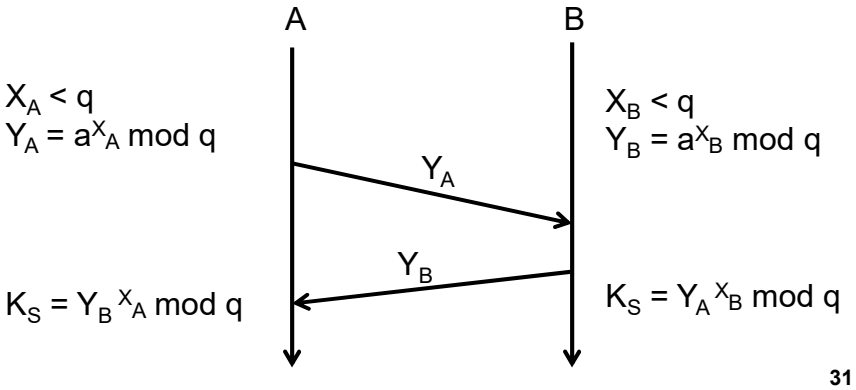
## 3. Hệ mật mã khóa bất đối xứng

- Asymmetric key cryptography, Public key cryptography
- Tháng 11/1976, Diffie và Hellman giới thiệu ý tưởng về một kịch bản chia sẻ khóa bí mật (của hệ mật mã khóa đối xứng) mới mà không truyền trực tiếp giá trị của khóa.
- Độ an toàn dựa trên độ khó khi giải một số bài toán:
  - Phân tích một số thành thừa số nguyên tố
  - Tính logarit rời rạc
- Các thuật toán dựa trên các hàm toán học
- Một số hệ mật mã khóa công khai: RSA, El-Gamal

30

## 3.1. Kịch bản trao đổi khóa bí mật của Diffie-Hellman

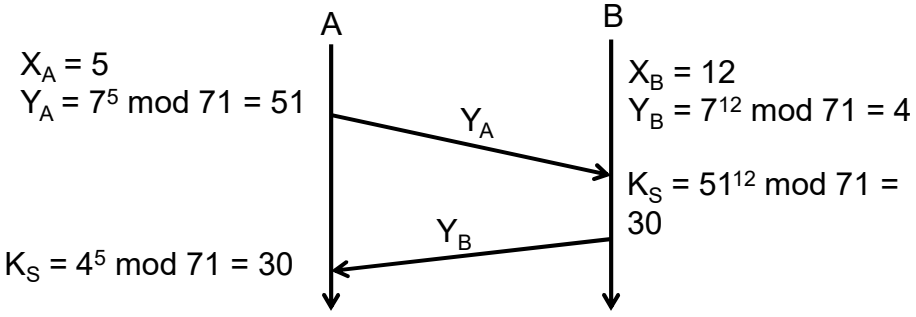
- Alice và Bob cùng chia sẻ một khóa nhóm (q, a). Trong đó
  - q là một số nguyên tố
  - 1< a < q thỏa mãn: $(a^i \bmod q) \neq a^j \bmod q \ \forall\ 1 < i \neq j < q$




31

## Ví dụ

- Khóa chung của nhóm q = 71, a = 7
  - Hãy tự kiểm tra điều kiện thỏa mãn của a
- A chọn $X_A = 5$, tính $Y_A = 7^5 \bmod 71 = 51$




- Vấn đề an toàn của sơ đồ này sẽ được xem xét đến sau.

Rút ra được điều gì từ sơ đồ này?

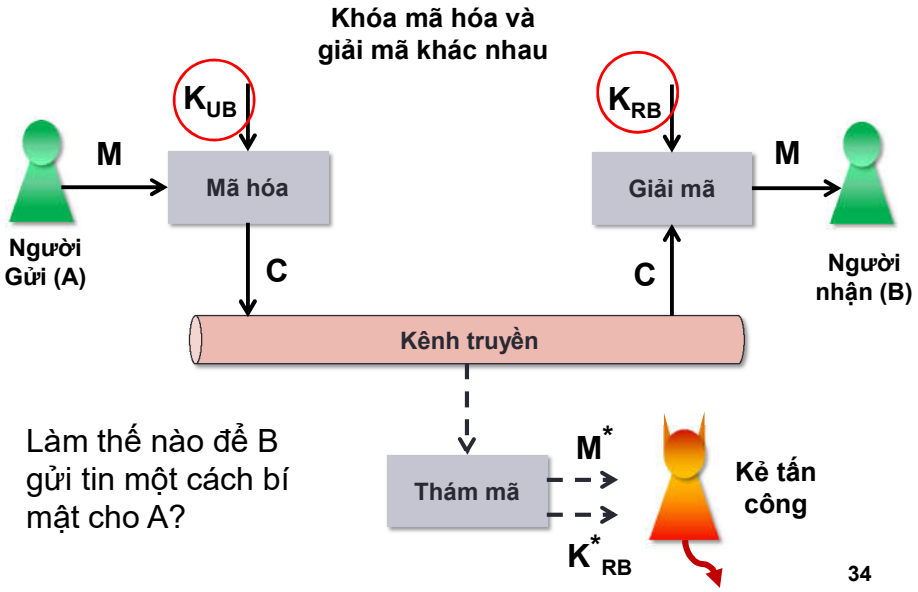
32

## 3.2. Sơ đồ chung

- Hệ mật mã gồm:
  - Bản rõ (plaintext-M): bản tin được sinh ra bởi bên gửi
  - Bản mật (ciphertext-C): bản tin che giấu thông tin của bản rõ, được gửi tới bên nhận qua một kênh không bí mật
  - Khóa: Bên nhận có **1 cặp** khóa:
    - ✓Khóa công khai $K_{UB}$ : công bố cho tất cả biết (trong đó có cả kẻ tấn công)
    - ✓Khóa cá nhân $K_{RB}$ : bên nhận giữ bí mật, không chia sẻ
  - Mã hóa (encrypt-E): $C = E(K_{UB}, M)$
  - Giải mã (decrypt): $M = D(K_{RB}, C) = D(K_{RB}, E(K_{UB}, M))$

33

## 3.2. Sơ đồ chung (tiếp)




34

## 3.2. Sơ đồ chung (tiếp)

- Yêu cầu đối với cặp khóa ($K_{UB}$, $K_{RB}$)
  - Hoàn toàn ngẫu nhiên
  - Có quan hệ về mặt toán học 1-1, nhưng...
  - Nếu chỉ có giá trị của $K_{UB}$ không thể tính được $K_{RB}$
  - $K_{RB}$ phải được giữ mật hoàn toàn

35

## Một ví dụ - Hệ mật RSA

- Sinh khóa:
  - Chọn p,q là hai số nguyên tố
  - Tính $n = p \times q$ , $\Phi(n) = (p-1)\times(q-1)$
  - Chọn e sao cho UCLN($\Phi(n)$, e) = 1 ;1< e < $\Phi(n)$
  - Tính d sao cho $(e \times d) \bmod \Phi(n) = 1$.
  - Khóa công khai : $K_U = (e,n)$
  - Khóa riêng : $K_R = (d,n)$
- Mã hóa : $C = M^e \bmod n$
- Giải mã: $M = C^d \bmod n$

36

# Một ví dụ - Hệ mật RSA

- Sinh khóa:
  - Chọn p = 5, q = 11
  - Tính n = p × q = 55, $\Phi(n) = (p-1)\times(q-1) = 40$
  - Chọn e sao cho UCLN($\Phi(n)$, e) = 1 và $1 < e < \Phi(n)$
    VD: e = 7
  - Tính d sao cho $(e \times d) \bmod \Phi(n) = 1$, $1 < d < \Phi(n)$
    d = 23
    Cặp khóa : $K_U = (7,55)$, $K_R = (23,55)$
- Mã hóa: M = 6 → C = 41
- Giải mã: C = 41 → M = 6

Nếu kẻ tấn công có $K_U$, làm thế nào để tính $K_R$?

37

# 3.3. Kết hợp mật mã khóa công khai và mật mã khóa đối xứng

- Ưu điểm của mật mã khóa công khai:
  - Không cần chia sẻ khóa mã hóa $K_{UB}$ một cách bí mật
    - ✓ Dễ dàng ứng dụng trong các hệ thống mở
  - Khóa giải mã $K_{RB}$ chỉ có B biết:
    - ✓ An toàn hơn
    - ✓ Có thể sử dụng $K_{RB}$ để xác thực nguồn gốc thông tin (Chúng ta sẽ quay lại vấn đề này trong bài sau)
  - Số lượng khóa để mã mật tỉ lệ tuyến tính với số phần tử (n phần tử → n cặp khóa)
- Nhưng...

38

## 3.3. Kết hợp mật mã khóa công khai và mật mã khóa đối xứng

- Vấn đề cần giải quyết trong hệ mật mã khóa công khai
  - Vẫn cần kênh an toàn để chia sẻ khóa.

  VD: Tấn công vào sơ đồ Diffie-Hellman bằng kỹ thuật man-in-the-middle

39

## 3.3. Kết hợp mật mã khóa công khai và mật mã khóa đối xứng

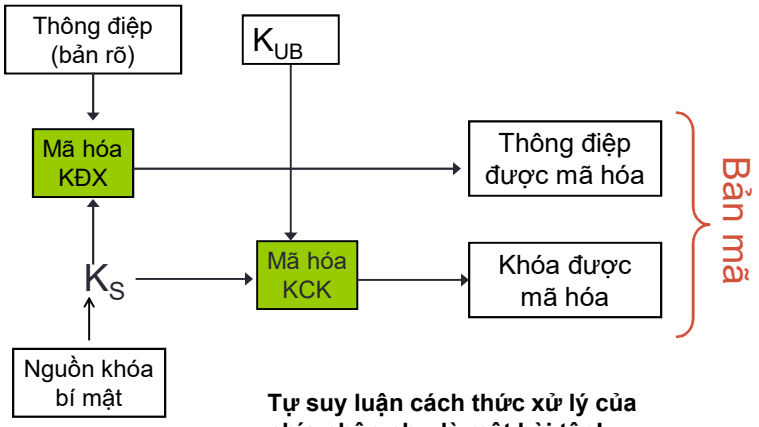
- Hạn chế của mật mã khóa công khai so với mật mã khóa đối xứng:
  - Kém hiệu quả hơn: khóa có kích thước lớn hơn, chi phí tính toán cao hơn
  - Có thể bị tấn công toán học

  → Kết hợp 2 hệ mật mã

40

# Sơ đồ "lai"

- Phía gửi




**Tự suy luận cách thức xử lý của phía nhận như là một bài tập!**

41

# 4. MỘT SỐ VẤN ĐỀ VỀ QUẢN LÝ KHÓA

Bùi Trọng Tùng,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

42

# 4.1. Kích thước và thời hạn khóa

- Khóa có kích thước bao nhiêu?
  - Mật mã được coi là an toàn khi phương pháp vét cạn (brute-force) là cách nhanh nhất để bẻ khóa
  - Mục tiêu: giảm thiểu nguy cơ bị tấn công vét cạn (đạt độ an toàn theo tính toán)
- Bạn nghe ở đâu đó, "dễ dàng" bẻ khóa mật mã DES có kích thước khóa 64 bit?
  - Năm 1999, hệ thống phá mã EFF DES (trị giá 250K$) bẻ khóa DES trong khoảng 1 ngày
  - Năm 2008, hệ thống phá mã COPACOBANA (trị giá 10K$) bẻ khóa DES trong 6,4 ngày

  Sử dụng định luật Moore để tính thời gian bẻ khóa trong năm 2016 với chi phí 10K$?

43

# Kích thước khóa

- Chi phí để bẻ khóa DES (năm 2008)
  - 64 bit: $10.000
  - 87 bit: $10.000.000.000 (thời gian bẻ khóa không đổi)
- Cần giữ thông tin mật trong bao lâu khi hệ thống phá mã là COPACOBANA? (năm 2008)
  - 64 bit: 6.4 ngày
  - 128 bit: ?
- Tuy nhiên, vét cạn là phương pháp tấn công tầm thường.
- Tham khảo kích thước khóa nên sử dụng trong tương lai tại địa chỉ
  http://csrc.nist.gov/groups/ST/toolkit/key_management.html

44

# Thông tin tham khảo – Kích thước khóa

| Date | Minimum of Strength | Symmetric Algorithms | Factoring Modulus | Discrete Logarithm Key | Discrete Logarithm Group | Elliptic Curve | Hash (A) | Hash (B) |
|---|---|---|---|---|---|---|---|---|
| 2010 (Legacy) | 80 | 2TDEA* | 1024 | 160 | 1024 | 160 | SHA-1**<br>SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 | SHA-1<br>SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 |
| 2011 - 2030 | 112 | 3TDEA | 2048 | 224 | 2048 | 224 | SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 | SHA-1<br>SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 |
| > 2030 | 128 | AES-128 | 3072 | 256 | 3072 | 256 | SHA-256<br>SHA-384<br>SHA-512 | SHA-1<br>SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 |
| >> 2030 | 192 | AES-192 | 7680 | 384 | 7680 | 384 | SHA-384<br>SHA-512 | SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 |
| >>> 2030 | 256 | AES-256 | 15360 | 512 | 15360 | 512 | SHA-512 | SHA-256<br>SHA-384<br>SHA-512 |

http://www.keylength.com

45

# Thời hạn khóa

- Thời hạn khóa (Cryptoperiod): thời gian khóa có hiệu lực sử dụng
  - Giảm xác suất bẻ khóa thành công của thám mã
  - Giảm thiệt hại khi xảy ra tình trạng khóa bị tấn công/sử dụng sai
- Các yếu tố ảnh hưởng đến thời hạn khóa:
  - Độ an toàn của hệ mật
  - Môi trường triển khai và vận hành ứng dụng
  - Lượng thông tin được mã hóa
  - Thời hạn giữ mật thông tin
  - Mục đích sử dụng khóa
  - Số phần tử chia sẻ khóa/Số bản sao được tạo ra
  - Sự phát triển của các hệ thống tính toán…

46

# Thông tin tham khảo – Thời hạn khóa

| Key Type<br>*Move the cursor over a type for description* | Cryptoperiod<br>Originator Usage Period (OUP) | Recipient Usage Period |
|---|---|---|
| Private Signature Key | 1-3 years | |
| Public Signature Key | Several years (depends on key size) | |
| Symmetric Authentication Key | <= 2 years | <= OUP + 3 years |
| Private Authentication Key | 1-2 years | |
| Public Authentication Key | 1-2 years | |
| Symmetric Data Encryption Key | <= 2 years | <= OUP + 3 years |
| Symmetric Key Wrapping Key | <= 2 years | <= OUP + 3 years |
| Symmetric and asymmetric RNG Keys | Upon reseeding | |
| Symmetric Master Key | About 1 year | |
| Private Key Transport Key | <= 2 years (1) | |
| Public Key Transport Key | 1-2 years | |
| Symmetric Key Agreement Key | 1-2 years | |
| Private Static Key Agreement Key | 1-2 years (2) | |
| Public Static Key Agreement Key | 1-2 years | |
| Private Ephemeral Key Agreement Key | One key agreement transaction | |
| Public Ephemeral Key Agreement Key | One key agreement transaction | |
| Symmetric Authorization Key | <= 2 years | |
| Private Authorization Key | <= 2 years | |
| Public Authorization Key | <= 2 years | |

http://www.keylength.com

47

# Nâng cấp thuật toán và khóa

- Thời hạn sử dụng thuật toán/khóa = Thời hạn thuật toán/khóa theo đề nghị - Thời gian cần giữ mật thông tin
  - Ví dụ: Thuật toán 3TDEA có thời hạn theo đề nghị là 2030, nếu yêu cầu giữ mật thông trong 5 năm → thời hạn sử dụng là 2025
- Nên sử dụng các thuật toán/khóa có thời hạn đủ dài, đảm bảo chỉ hết hạn khi thông tin không cần giữ mật
- Trong trường hợp không thể sử dụng các thuật toán/khóa đủ tốt, cần có kế hoạch nâng cấp thuật toán và khóa
  - Cần kiểm thử, đánh giá kế hoạch trước và sau khi nâng cấp

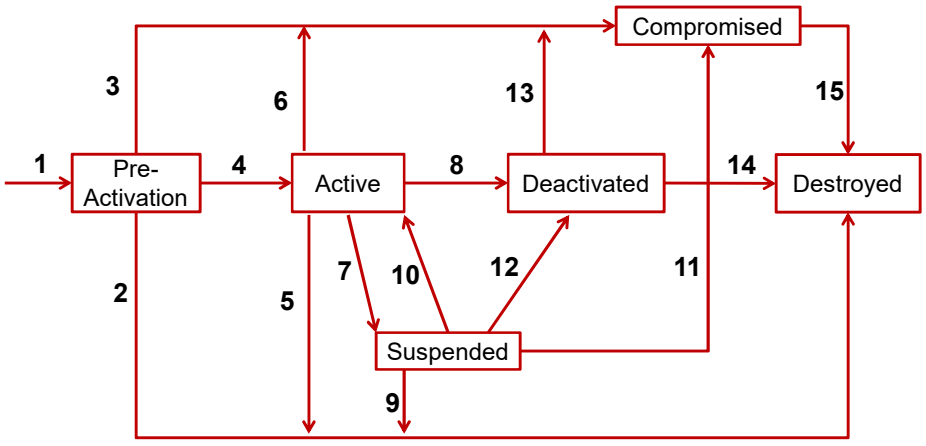
48

# 4.2. Các thông tin giao kết của khóa

- Khóa cần đặt trong mối giao kết (association) với các thực thể và thông tin khác:
  - Mục đích sử dụng
  - Thời hạn khóa
  - Trạng thái
  - Nguồn gốc
  - Chủ sở hữu khóa
  - Khóa cùng cặp (đối với khóa của hệ mật mã khóa đối xứng)
  - Khóa nhóm, tham số nhóm
  - Dữ liệu được ký/xác thực
  - Phiên sử dụng

49

# 4.3. Vòng đời của khóa




50

## Các trạng thái

- Pre-activation: Khóa đã được tạo, chờ điều kiện để sử dụng. Các chuyển trạng thái:
  - 1: Khóa được tạo
  - 2: Khóa không còn cần thiết
  - 3: Nghi ngờ khóa không đảm bảo tính bí mật hoặc toàn vẹn
  - 4: Đủ điều kiện để sử dụng. Bắt đầu tính thời hạn khóa
- Active: Khóa có thể sử dụng. Các chuyển trạng thái:
  - 5: Khóa hết hạn hoặc được thay thế
  - 6: Nghi ngờ khóa không đảm bảo tính bí mật hoặc toàn vẹn
  - 7: Khóa tạm thời không được sử dụng trong một khoảng thời gian
  - 8: Khóa không cần sử dụng trong các phiên tới

51

## Các trạng thái

- Suspended: Tạm dừng sử dụng. Các chuyển trạng thái:
  - 9: Khóa hết hạn sử dụng
  - 10: Các nguyên nhân gây tạm dừng không còn tồn tại
  - 11: Nghi ngờ khóa không đảm bảo tính bí mật hoặc toàn vẹn
  - 12: Khóa không cần sử dụng đến
  - Lưu ý: Trong một số trường hợp có thể sử dụng khóa ở tình trạng này ở phía nhận để giải mã/xác thực/…
- Deactivated: Ngừng sử dụng.
  - 13: Nghi ngờ khóa không đảm bảo tính bí mật hoặc toàn vẹn
  - 14: Khóa chắc chắn không cần sử dụng ở các bên
  - Lưu ý: Trong một số trường hợp có thể sử dụng khóa ở tình trạng này ở phía nhận để giải mã/xác thực/…

52

## Các trạng thái

- Compromised: khóa không còn đảm bảo yêu cầu bí mật/toàn vẹn. Các chuyển trạng thái:
  - 15: Khóa chắc chắn không cần sử dụng ở các bên
  - Lưu ý: đối với khóa của hệ mật bất đối xứng, trong một số trường hợp giới hạn vẫn có thể sử dụng
- Destroyed: Khóa bị hủy
  - Lưu ý: các thông tiên liên quan trong các giao kết của khóa vẫn có thể được tiếp tục lưu trữ cho mục đích quản lý

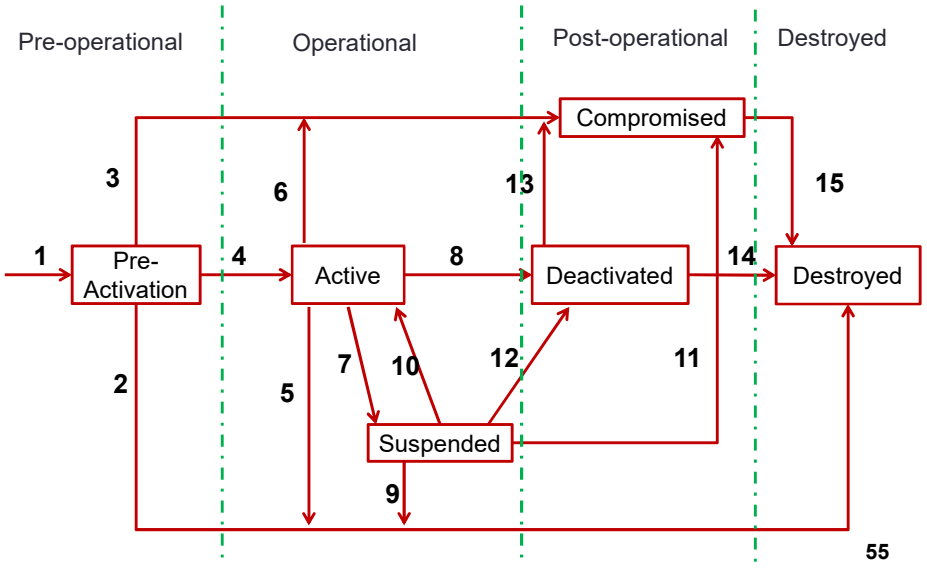
53

## 4.4. Quy trình quản lý khóa

- Gồm 4 giai đoạn:
  - Pre-operational: Khóa chưa/có thể đã được sinh nhưng chưa có đủ thông tin để phân phối, trao đổi khóa
  - Operational: Khóa có thể phân phối, trao đổi, sử dụng
  - Post-operational: Quá trình phân phối, trao đổi, sử dụng khóa kết thúc. Các thông tin và khóa được sử dụng trong một số điều kiện giới hạn
  - Destroyed: Hủy khóa

54

# 4.4. Quy trình quản lý khóa




55

# 4.4. Quy trình quản lý khóa

- Pre-operational: Thực hiện các chức năng:
  - Đăng ký thực thể
  - Khởi tạo hệ thống
  - Khởi tạo người dùng
  - Cài đặt thông tin phân phối, trao đổi khóa
  - Thiết lập khóa
  - Đăng ký, cấp phát khóa
- Operation: Thực hiện các chức năng:
  - Lưu trữ
  - Truy cập
  - Sao lưu và khôi phục
  - Thay đổi khóa

56

## 4.4. Quy trình quản lý khóa

- Post-operational: Thực hiện các chức năng:
  - Lưu kho
  - Hủy đăng ký thực thể
  - Hủy đăng ký khóa
  - Thu hồi khóa
  - Hủy khóa
- Destroyed: Thực hiện các chức năng tùy thuộc theo yêu cầu
  - Hủy bản ghi lưu trữ
  - Hủy thông tin giao kết
  - Kiểm toán

57

## Một số lưu ý khác

- Những sai lầm khi sử dụng mật mã:
  - Mật mã là giải pháp vạn năng (những bài sau chúng ta sẽ phân tích kỹ hơn)
  - Khóa có kích thước lớn, mã hóa sẽ an toàn
  - Sửa đổi/Thêm một vài yếu tố bí mật vào giải thuật, hệ mật mã sẽ an toàn hơn

58

# Một số lưu ý khác

- Chỉ sử dụng thuật toán chuẩn
- Đừng tự thiết kế hệ mật mã cho riêng mình:
  - Nếu không thể sử dụng các hệ mật mã đã có, hãy xem lại hệ thống
  - Nếu bắt buộc phải sử dụng hệ mật mã mới hoàn toàn, hãy đánh giá một cách cẩn thận
- Sử dụng khóa khác nhau cho các mục đích khác nhau
- Mật mã chưa đáp ứng yêu cầu về toàn vẹn
  - Khi sử dụng mật mã hãy thêm vào các sơ đồ đáp ứng toàn vẹn nội dung thông tin và xác thực nguồn gốc thông tin (sẽ đề cập đến trong những bài sau)
- Sinh khóa:
  - Sử dụng những hàm sinh số ngẫu nhiên cho mật mã
  - Đừng sử dụng những hàm sinh số ngẫu nhiên chuẩn rand(), srand()...

59