

# An toàn và bảo mật thông tin

*Giáo viên:* TS. Lê Thị Anh

Sdt/zalo: 0976621138

# Tổng quan môn học

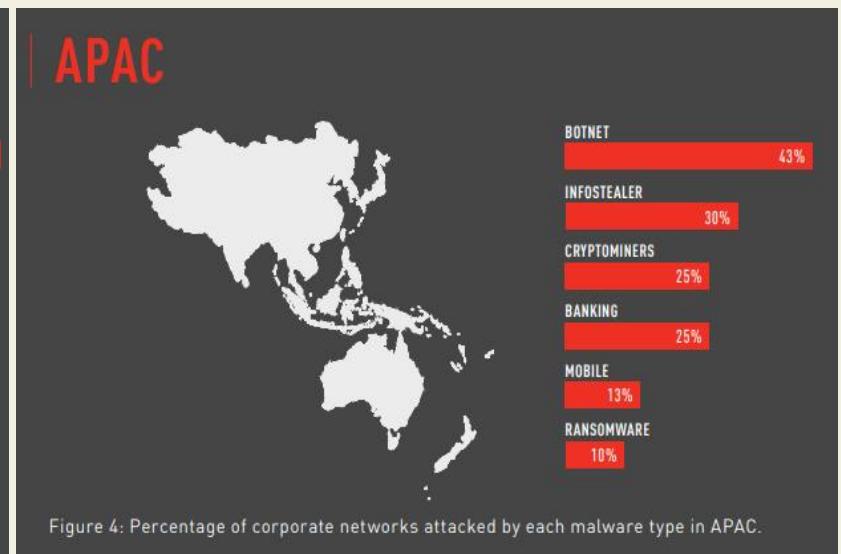
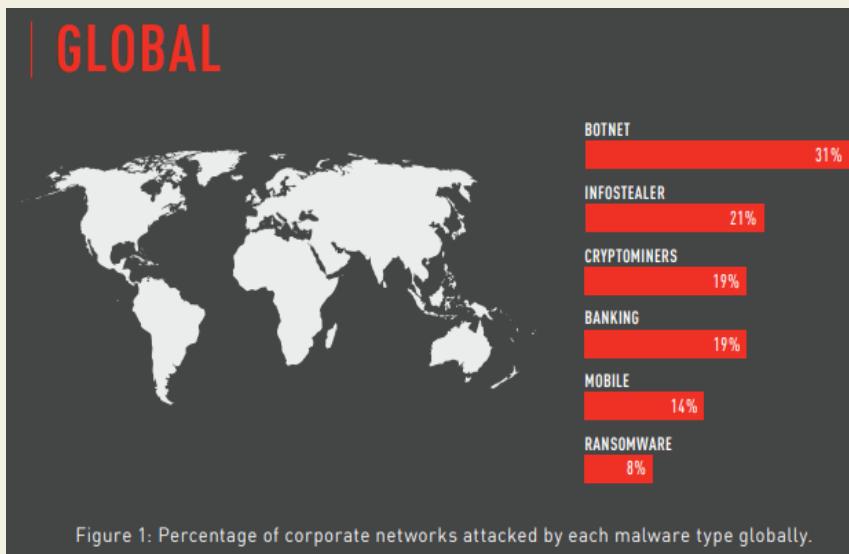
- Mã học phần: IT6001
- Số tín chỉ: 3(2.5;0.5;0)
- Bộ môn phụ trách: Kỹ thuật và mạng máy tính
- Đánh giá: 02 bài kiểm tra thường xuyên 1, 2;  
01 bài tập lớn thi hết môn.
- Tài liệu học tập:
  - Tài liệu chính: Giáo trình bảo mật an toàn thông tin – Khoa CNTT, Đại học Công nghiệp Hà Nội

# I. Tổng quan về an toàn thông tin

## Một số thống kê về tình hình an toàn thông tin

Một số thống kê về An ninh mạng trong báo cáo “Security Report 01/24/22” của hãng bảo mật Checkpoint.

Năm 2021, tổng các cuộc tấn công vào các mạng doanh nghiệp tăng 50% mỗi tuần so với năm 2020.



## 0.1. Một số thống kê về tình hình an toàn thông tin

Một số thống kê về An ninh mạng trong báo cáo “Security Report 01/24/22” của hãng bảo mật Checkpoint.

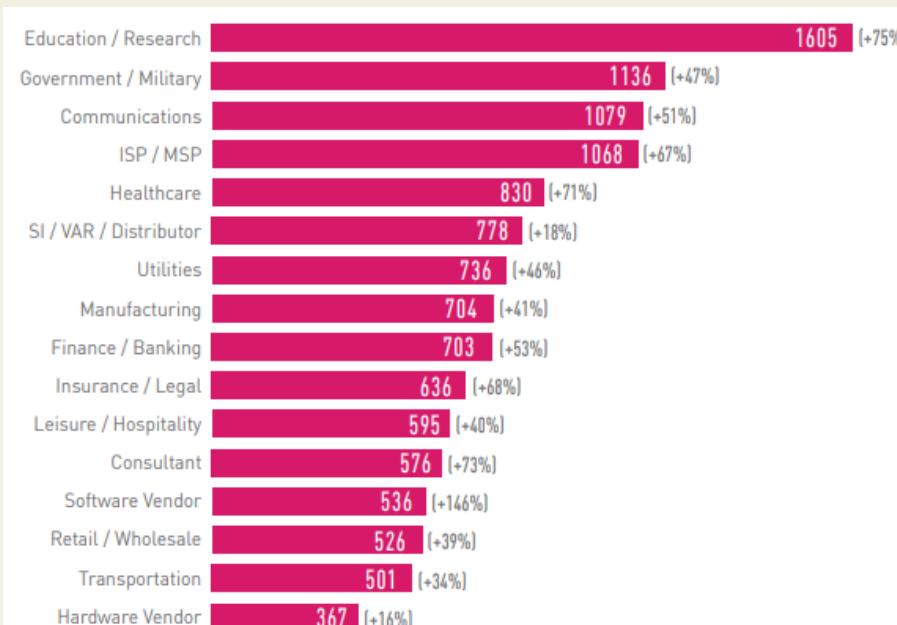
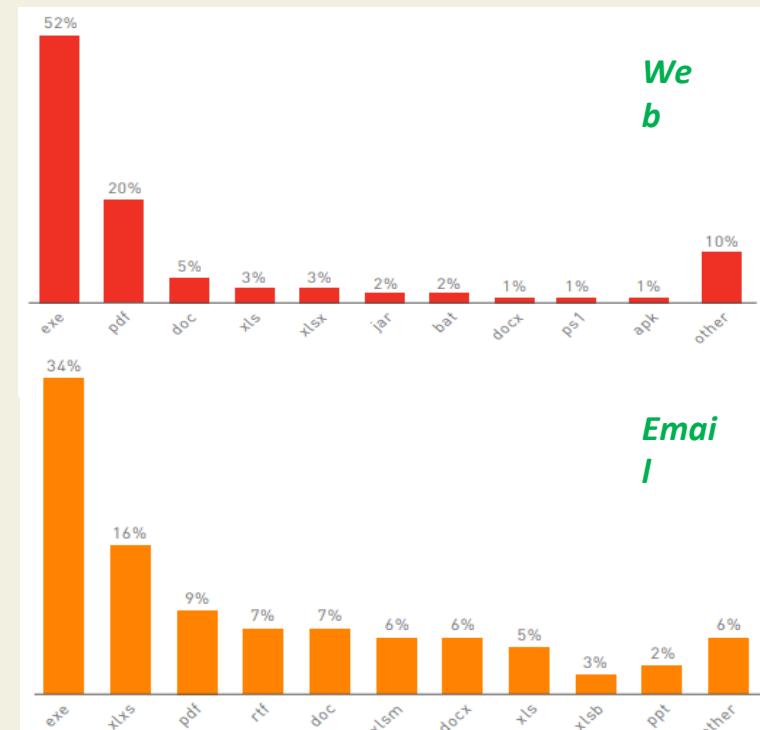
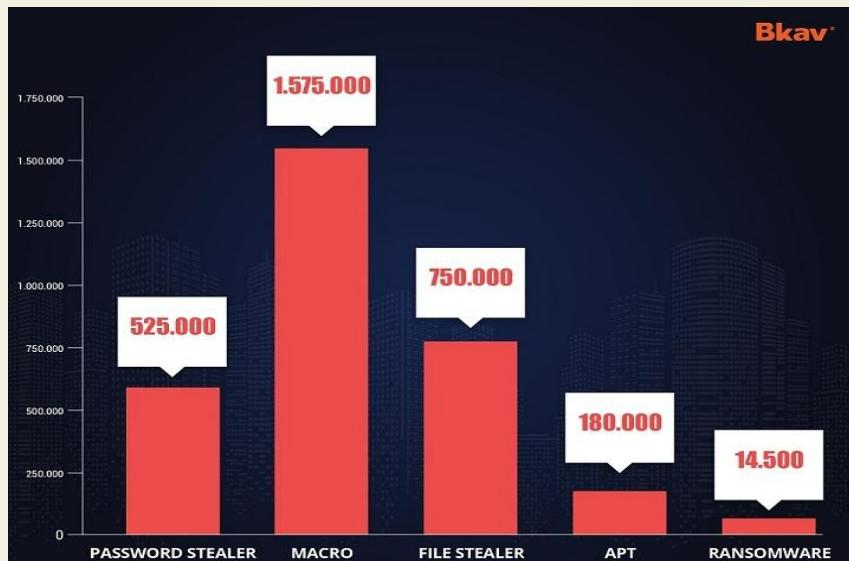


Figure 6: Average weekly attacks per organization by Industry 2021, compared to 2020.



## 0.1. Một số thống kê về tình hình an toàn thông tin

**Việt Nam:** Luật An ninh mạng được Quốc hội thông qua năm 2018 và chính thức có hiệu lực từ 01/01/2019, với 7 chương, 43 điều quy định về hoạt động bảo vệ an ninh quốc gia, bảo đảm trật tự, an toàn xã hội trên không gian mạng, bên cạnh đó là trách nhiệm của cơ quan, tổ chức, cá nhân có liên quan. ([tập trung điều 2, 8, 19, 41, 42](#))



Số máy tính Việt Nam bị nhiễm 5 dòng mã độc phổ biến năm 2022

Năm 2022, thiệt hại do mã độc máy tính gây ra đối với người dùng Việt Nam ở mức 21,2 nghìn tỷ (tương đương 883 triệu USD) → Mức thiệt hại nhóm thấp so với thế giới (tổng cầu 1000 tỷ USD).

Lần đầu tiên sau hơn 10 năm Bkav thực hiện thống kê, con số thiệt hại ghi nhận giảm so với các năm trước đó.

Việt Nam tăng 25 bậc về chỉ số an toàn an ninh mạng GCI, cho thấy nỗ lực của Chính phủ và giới an ninh mạng trong nước.

## Một số thống kê về tình hình toàn thông tin

Dữ liệu về mã độc được lấy từ bản đồ mối đe dọa trên mạng toàn cầu của Checkpoint từ tháng 1 đến tháng 12 năm 2021 của hãng: <https://threatmap.checkpoint.com/>

### TOP MALWARE FAMILIES

#### ■ GLOBAL

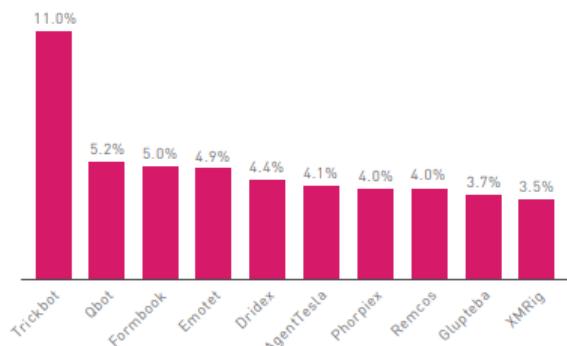


Figure 10: Most prevalent malware globally.

Percentage of corporate networks attacked by each malware family.

### ■ ASIA PACIFIC (APAC)

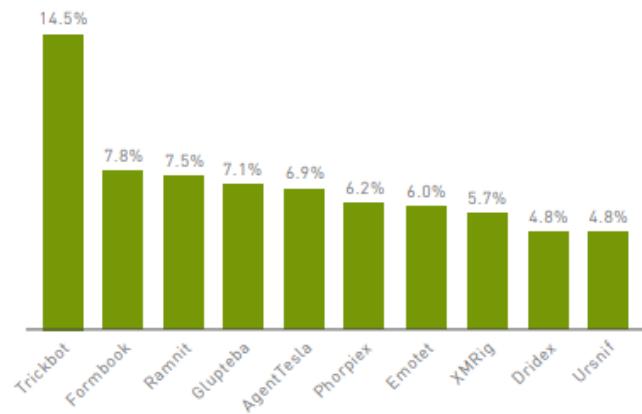


Figure 13: Most prevalent malware in APAC.

# 1. Tại sao phải bảo vệ thông tin

- ✓ Thông tin là một bộ phận quan trọng và là tài sản thuộc quyền sở hữu của các tổ chức
- ✓ Sự thiệt hại và lạm dụng thông tin không chỉ ảnh hưởng đến người sử dụng hoặc các ứng dụng mà nó còn gây ra các hậu quả tai hại cho toàn bộ tổ chức đó
- ✓Thêm vào đó sự ra đời của Internet đã giúp cho việc truy cập thông tin ngày càng trở nên dễ dàng hơn

## 2. Khái niệm hệ thống và tài sản của hệ thống

- **Khái niệm hệ thống** : Hệ thống là một tập hợp các máy tính bao gồm các thành phần, phần cứng, phần mềm và dữ liệu làm việc được tích luỹ qua thời gian.
- **Tài sản của hệ thống bao gồm:**
  - ✓ Phần cứng
  - ✓ Phần mềm
  - ✓ Dữ liệu
  - ✓ Các truyền thông giữa các máy tính của hệ thống
  - ✓ Môi trường làm việc
  - ✓ Con người

### 3. Các mối đe dọa đối với một hệ thống và các biện pháp ngăn chặn

- **Có 3 hình thức chủ yếu đe dọa đối với hệ thống:**

- ✓ **Phá hoại:** kẻ thù phá hỏng thiết bị phần cứng hoặc phần mềm hoạt động trên hệ thống.
- ✓ **Sửa đổi:** Tài sản của hệ thống bị sửa đổi trái phép. Điều này thường làm cho hệ thống không làm đúng chức năng của nó. Chẳng hạn như thay đổi mật khẩu, quyền người dùng trong hệ thống làm họ không thể truy cập vào hệ thống để làm việc.
- ✓ **Can thiệp:** Tài sản bị truy cập bởi những người không có thẩm quyền. Các truyền thông thực hiện trên hệ thống bị ngăn chặn, sửa đổi.

### 3. Các mối đe dọa đối với một hệ thống và các biện pháp ngăn chặn

- **Các đe dọa đối với một hệ thống thông tin có thể đến từ ba loại đối tượng như sau:**

Các đối tượng từ ngay bên trong hệ thống (insider), đây là những người có quyền truy cập hợp pháp đối với hệ thống.

Những đối tượng bên ngoài hệ thống (hacker, cracker), thường các đối tượng này tấn công qua những đường kết nối với hệ thống như Internet chẳng hạn.

Các phần mềm (chẳng hạn như spyware, adware ...) chạy trên hệ thống.

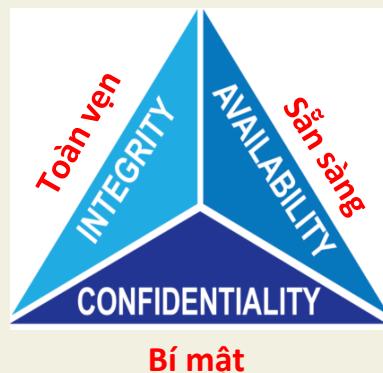
### 3. Các mối đe dọa đối với một hệ thống và các biện pháp ngăn chặn

- **Các biện pháp ngăn chặn:**

- ✓ **Điều khiển thông qua phần mềm:** dựa vào các cơ chế an toàn bảo mật của hệ thống nền (hệ điều hành), các thuật toán mật mã học
- ✓ **Điều khiển thông qua phần cứng:** các cơ chế bảo mật, các thuật toán mật mã học được cứng hóa để sử dụng
- ✓ **Điều khiển thông qua các chính sách của tổ chức:** ban hành các qui định của tổ chức nhằm đảm bảo tính an toàn bảo mật của hệ thống.

## 4. Mục tiêu của an toàn thông tin

- **Computer Security:** The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).
- **An toàn mạng máy tính:** Sự bảo vệ dành cho hệ thống thông tin tự động nhằm đạt được các mục tiêu đó là duy trì tính toàn vẹn, tính sẵn sàng (tính khả dụng) và tính bí mật của tài nguyên hệ thống thông tin (bao gồm phần cứng, mềm, phần sụn, thông tin/dữ liệu và viễn thông).



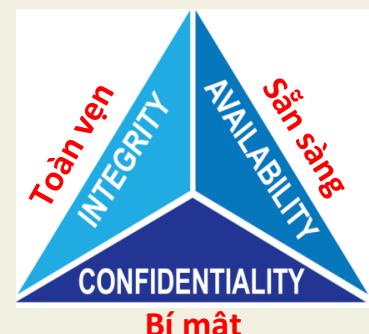
Hình 1. Tam giác CIA

Ba nguyên tắc cốt lõi này  
phải dẫn đường cho tất cả  
các hệ thống an ninh mạng

## 4. Mục tiêu An toàn thông tin

**Tính bí mật:** là sự ngăn ngừa việc tiết lộ trái phép những thông tin quan trọng, nhạy cảm. Gồm 2 nội dung là Bí mật về dữ liệu và Quyền riêng tư.

→ Đối với an ninh mạng thì tính bí mật rõ ràng là điều đầu tiên được nói đến và nó thường xuyên bị tấn công nhất.



**Tính toàn vẹn:** Là sự phát hiện và ngăn ngừa việc sửa đổi trái phép về dữ liệu, thông tin và hệ thống, do đó Bảo đảm sự chính xác về dữ liệu và hệ thống. Gồm có tòan vẹn về dữ liệu và tòan vẹn của hệ thống:

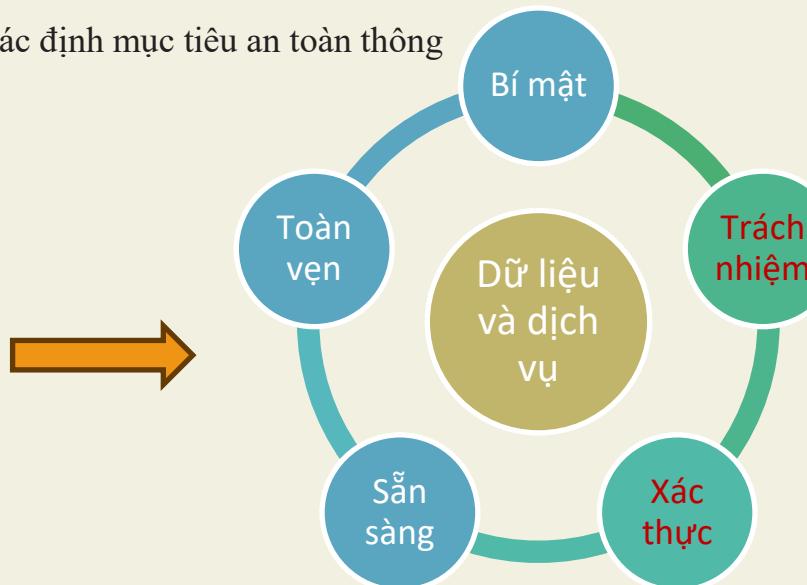
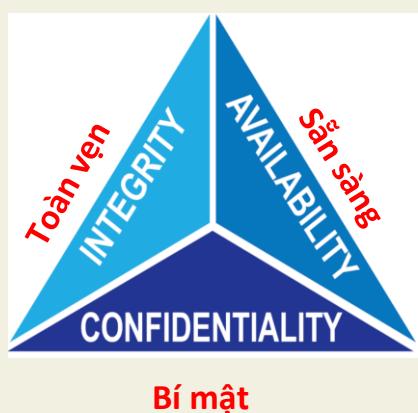
→ Tòan vẹn dữ liệu: Đảm bảo rằng dữ liệu và các chương trình chỉ được thay đổi theo bởi người được cấp quyền.

→ Tính toàn vẹn của hệ thống: Đảm bảo rằng một hệ thống thực hiện chức năng dự kiến của nó một cách nguyên vẹn, không bị thao túng trái phép một cách có chủ ý hoặc vô ý.

**Tính sẵn sàng:** Đảm bảo truy cập và sử dụng thông tin kịp thời và đáng tin cậy. Mất tính sẵn sàng là sự gián đoạn truy cập hoặc gián đoạn sử dụng thông tin hoặc gián đoạn sử dụng hệ thống thông tin.

## 4. Mục tiêu của an toàn thông tin

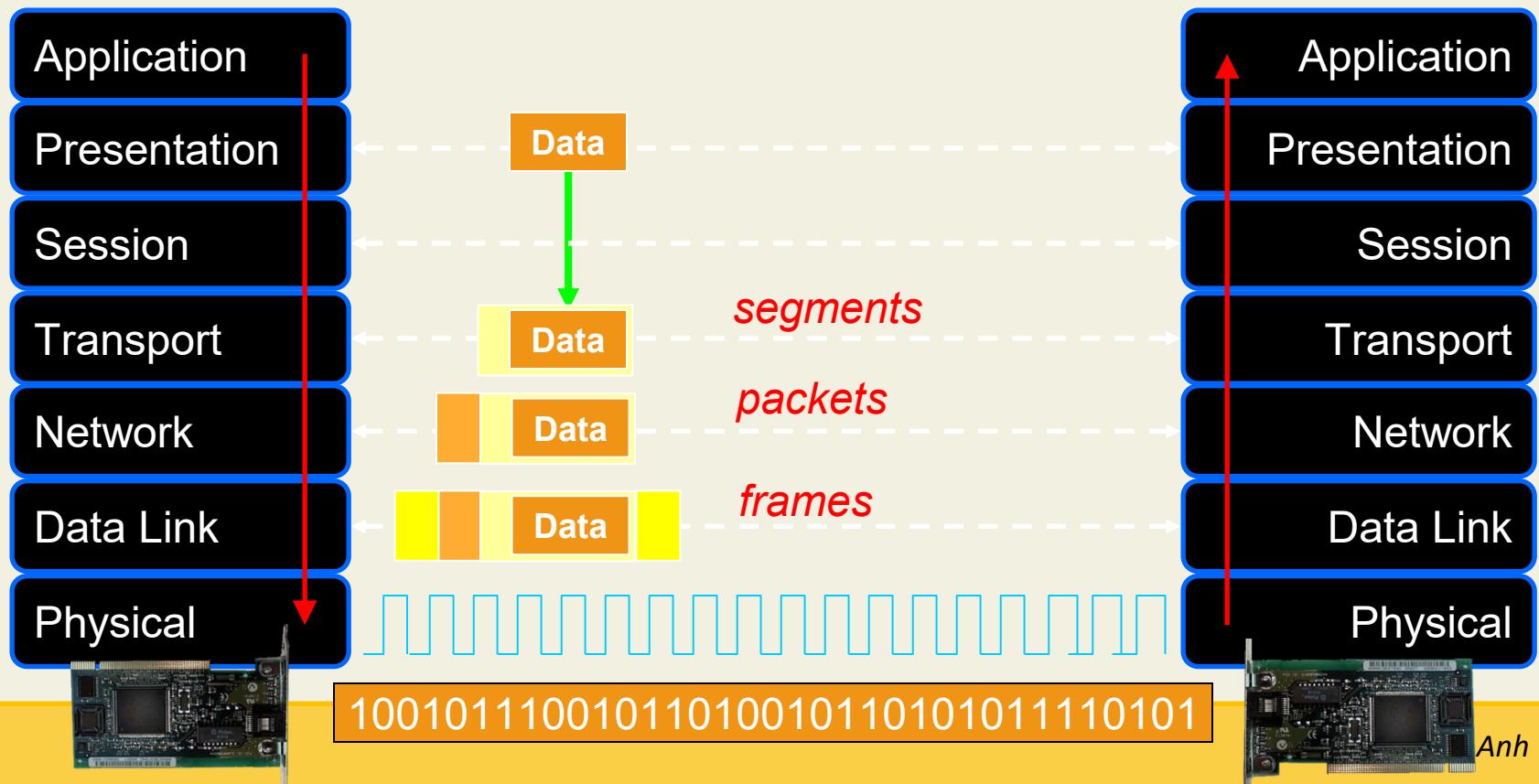
Một bức tranh hoàn chỉnh để xác định mục tiêu an toàn thông tin được đề xuất gồm 5 yếu tố:



**Trách nhiệm:** Mục tiêu an ninh quy định các hành động của một thực thể phải được quy một cách duy nhất về thực thể đó. Điều này hỗ trợ chống từ chối, ngăn chặn, cách ly lối, phát hiện và ngăn chặn xâm nhập, phục hồi sau hành động và hành động pháp lý.

**Tính xác thực:** Thể hiện thuộc tính được xác minh và có độ tin cậy; độ tin cậy vào tính hợp lệ của việc truyền thông, tin nhắn hoặc người khởi tạo tin nhắn

# 5. Mô hình OSI



# 6. Các loại tấn công an toàn thông tin

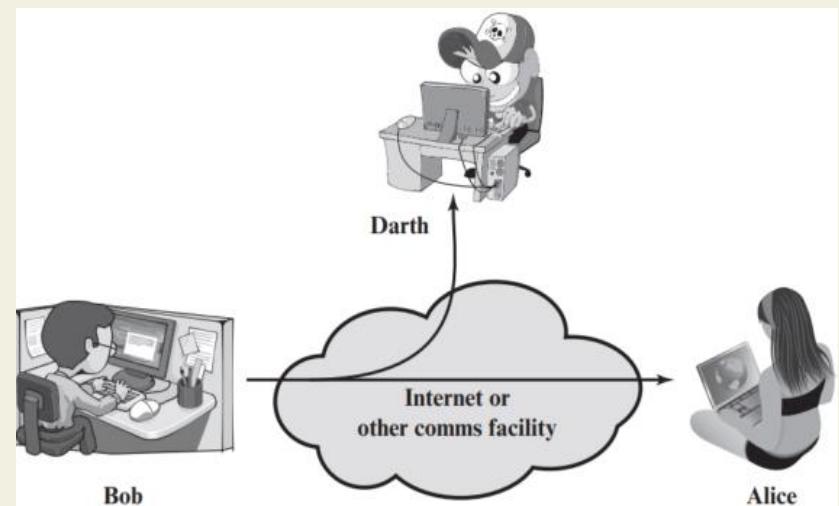
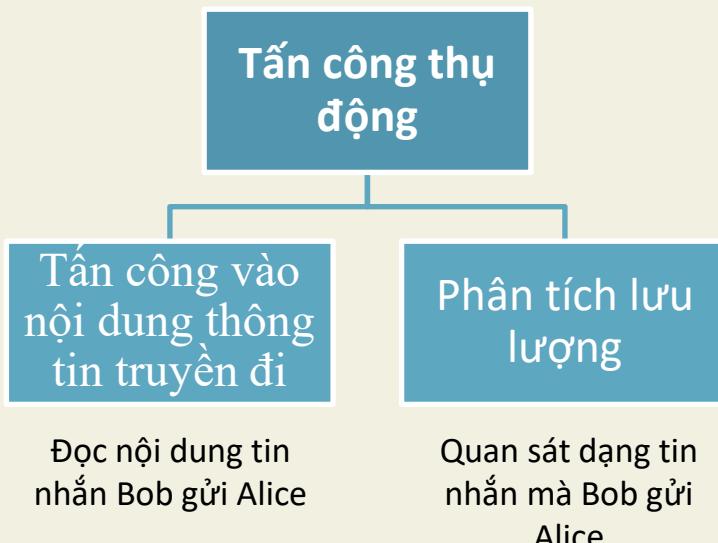
## Tấn công an toàn

Có 2 loại hình tấn công an ninh chính được sử dụng trong cả X.800 (đây là kiến trúc bảo mật cho hệ thống OSI được ITU quy định), tiêu chuẩn RFC 4949 (RFC viết tắt của Request for comment, bao gồm các thuật ngữ bảo mật Internet).

- **Tấn công thụ động:** là cuộc tấn công cố gắng tìm hiểu hoặc sử dụng thông tin từ hệ thống nhưng không ảnh hưởng đến tài nguyên của hệ thống.
- **Tấn công chủ động:** là cuộc tấn công mà attacker cố gắng thay đổi tài nguyên hệ thống hoặc ảnh hưởng đến hoạt động của các hệ thống đó

# 6. Các loại tấn công an toàn thông tin

**Tấn công thụ động:** Các cuộc tấn công bị động có bản chất là nghe lén hoặc giám sát đường truyền dữ liệu. Mục tiêu là lấy được thông tin đang được truyền đi.

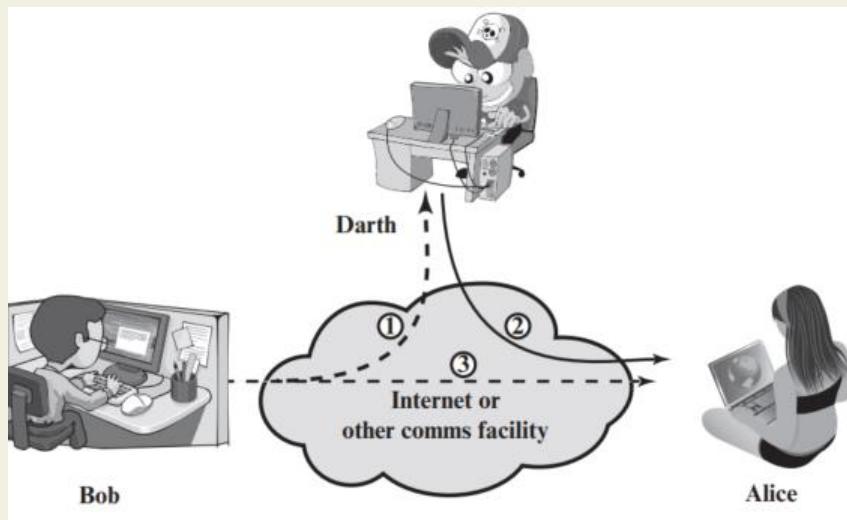


Các cuộc tấn công thụ động rất khó phát hiện do không tạo ra sự thay đổi gì về dữ liệu

Để đối phó với các cuộc tấn công này, chúng ta phải có các kỹ thuật phòng ngừa (như mã hóa) hơn là phát hiện.

## 6. Các loại tấn công an toàn thông tin

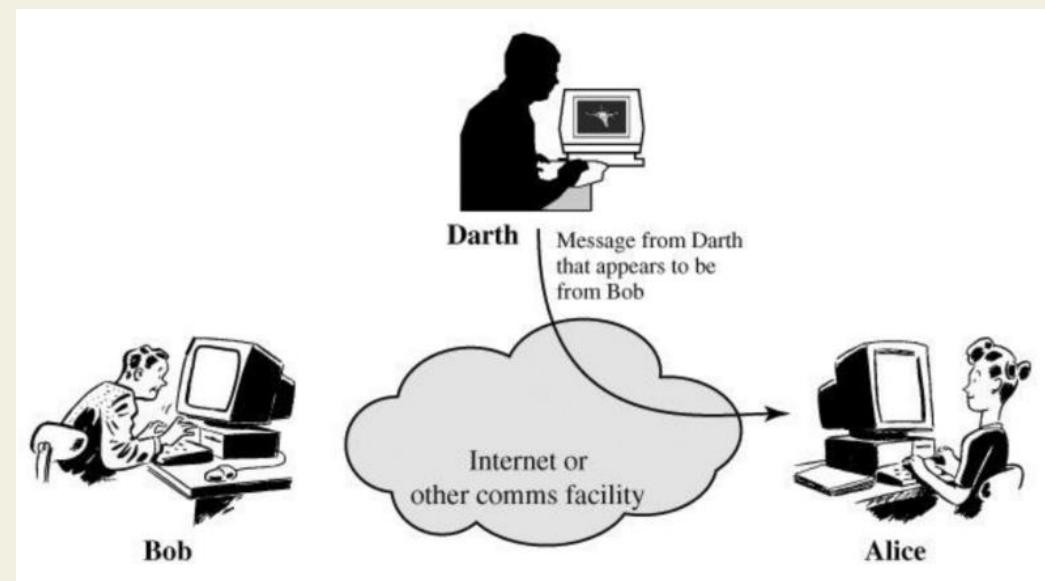
**Tấn công chủ động:** liên quan đến việc sửa đổi luồng dữ liệu hoặc tạo luồng giả và có thể được chia thành 4 loại: giả mạo, phát lại, sửa đổi thông tin, và từ chối dịch vụ.



# 6. Các loại tấn công an toàn thông tin

## Tấn công chủ động

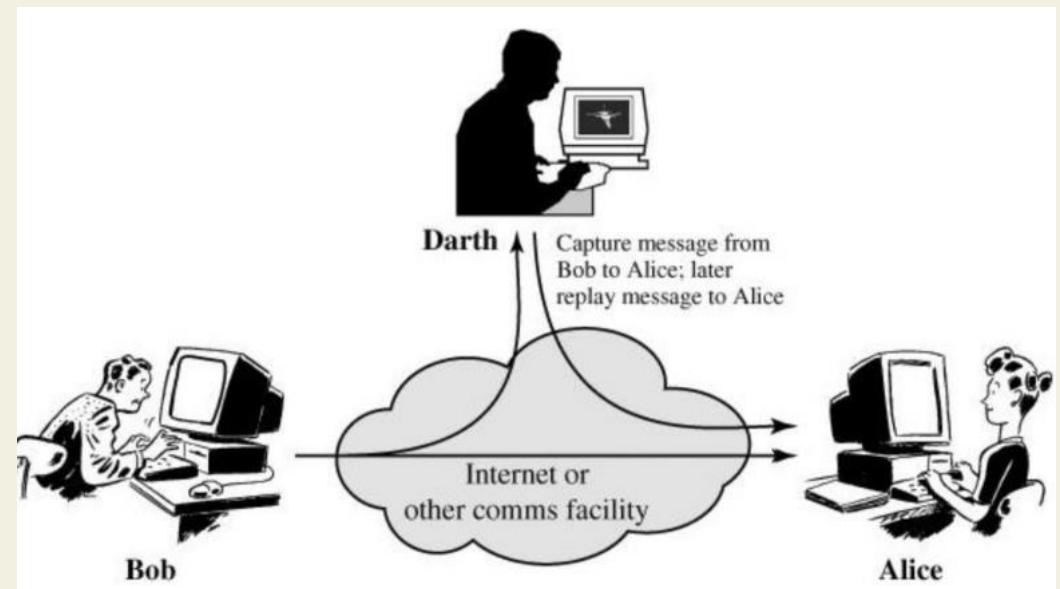
Giả mạo: Diễn ra khi một thực thể giả vờ một thực thể khác (2) – tin nhắn từ Darth tới Alice nhưng lại giả vờ là từ Bob



# 6. Các loại tấn công an toàn thông tin

## Tấn công chủ động

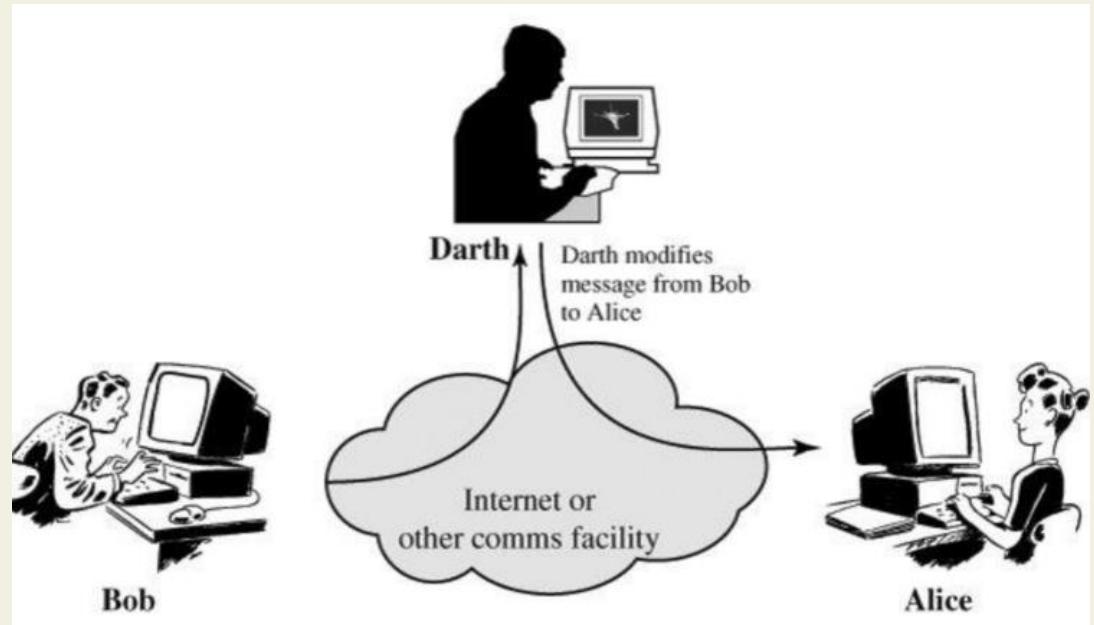
- Phát lại: Liên quan đến việc nắm bắt thụ động dữ liệu và truyền lại sau đó tạo ra hiệu ứng không xác thực (1,2,3) – Darth bắt gói tin từ Bob tới Alice; sau đó phát lại tin nhắn tới Alice



## 6. Các loại tấn công an toàn thông tin

### Tấn công chủ động

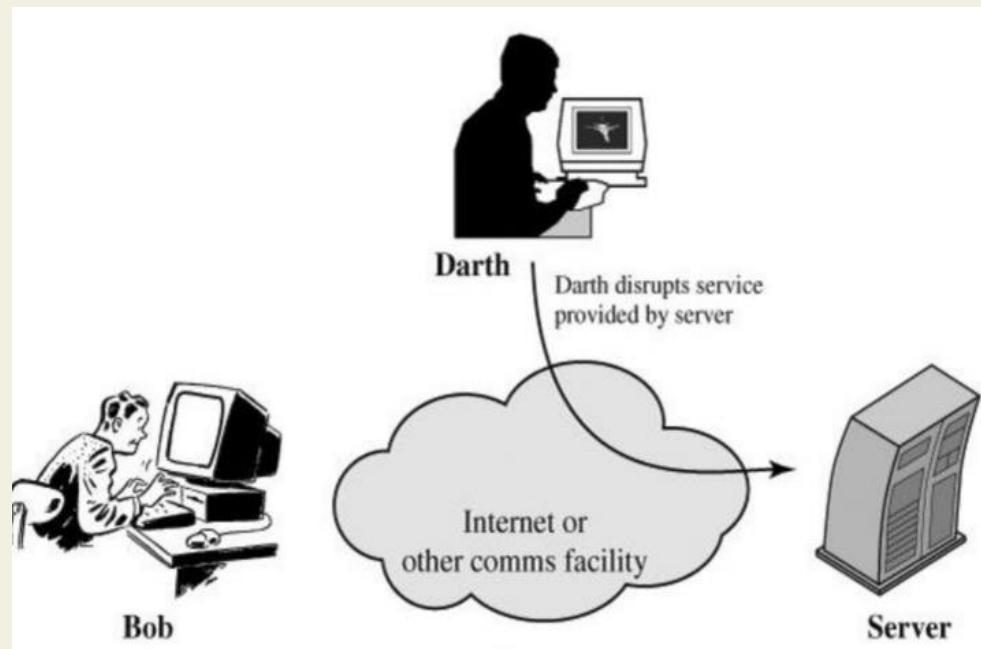
- Sửa đổi: tin nhắn bị sửa lại một phần hoặc tin nhắn gửi đi bị trễ để tạo ra hiệu ứng không xác thực (1, 2) – Darth sửa tin nhắn mà Bob gửi cho Alice.



## 6. Các loại tấn công an toàn thông tin

### Tấn công chủ động

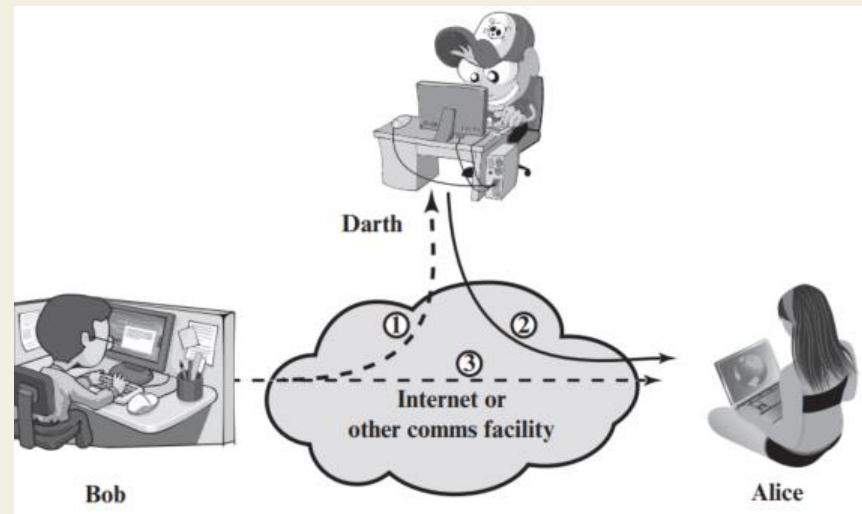
- **Tùy chỉnh dịch vụ:** là ngăn chặn hoặc cản trở việc sử dụng hoặc quản lý các phương tiện truyền thông (3 – Darth sẽ ngắt dịch vụ được cung cấp bởi máy chủ).



# 6. Các loại tấn công an toàn thông tin

## Tấn công chủ động:

- Các cuộc tấn công chủ động thể hiện các đặc điểm ngược lại của các tấn công bị động.
- Có rất nhiều lỗ hổng vật lý, phần mềm và mạng tiềm ẩn → rất khó để ngăn chặn hoàn toàn tấn công chủ động
- Mục tiêu là phát hiện các cuộc tấn công chủ động và khắc phục sự cố



## 7. An toàn thông tin bằng mật mã

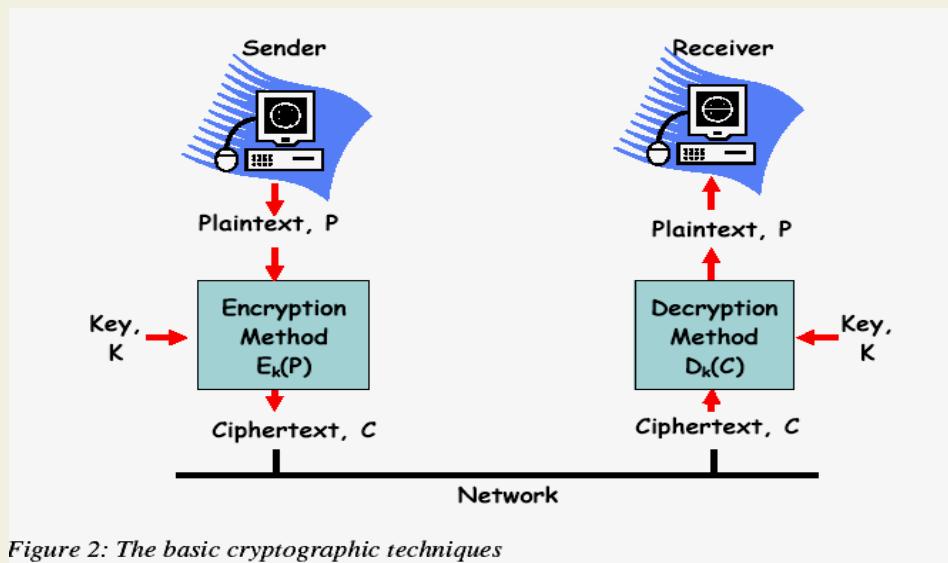
Mật mã là một ngành khoa học chuyên nghiên cứu các phương pháp truyền tin bí mật.

Mật mã bao gồm : Lập mã và phá mã.

- **Lập mã hay** mã hóa và giải mã.
- Các sản phẩm của lĩnh vực này là các hệ mã mật , các hàm băm, các hệ chữ ký điện tử, các cơ chế phân phối, quản lý khóa và các giao thức mật mã.
- **Phá mã:** Nghiên cứu các phương pháp phá mã hoặc tạo mã giả. Sản phẩm của lĩnh vực này là các phương pháp phá mã , các phương pháp giả mạo chữ ký, các phương pháp tấn công các hàm băm và các giao thức mật mã

## 7. An toàn thông tin bằng mật mã

- Một trong những nghệ thuật để bảo vệ thông tin là biến đổi nó thành một định dạng mới khó đọc.
- Viết mật mã có liên quan đến việc mã hoá các thông báo trước khi gửi chúng đi và tiến hành giải mã chúng lúc nhận được



## 7. An toàn thông tin bằng mật mã

- ✓ **Phương thức mã hoá thay thế:** là phương thức mã hoá mà từng ký tự gốc hay một nhóm ký tự gốc của bản rõ được thay thế bởi các từ, các ký hiệu khác hay kết hợp với nhau cho phù hợp với một phương thức nhất định và khoá.
  
- ✓ **Phương thức mã hoá hoán vị:** là phương thức mã hoá mà các từ mã của bản rõ được sắp xếp lại theo một phương thức nhất định.

## 8. Hệ mật mã

- ✓ Hệ mật mã phải che dấu được nội dung của văn bản rõ (PlainText).
- ✓ Tạo các yếu tố xác thực thông tin, đảm bảo thông tin lưu hành trong hệ thống đến người nhận hợp pháp là xác thực (Authenticity).
- ✓ Tổ chức các sơ đồ chữ ký điện tử, đảm bảo không có hiện tượng giả mạo, mạo danh để gửi thông tin trên mạng.

## 8. Hệ mật mã

- **Khái niệm cơ bản**

**Bản rõ** X được gọi là bản tin gốc. Bản rõ có thể được chia nhỏ có kích thước phù hợp.

**Bản mã** Y là bản tin gốc đã được mã hoá. Ở đây ta thường xét phương pháp mã hóa mà không làm thay đổi kích thước của bản rõ, tức là chúng có cùng độ dài.

**Mã** là thuật toán E chuyển bản rõ thành bản mã. Thông thường chúng ta cần thuật toán mã hóa mạnh, cho dù kẻ thù biết được thuật toán, nhưng không biết thông tin về khóa cũng không tìm được bản rõ.

## 8. Hệ mật mã

Một hệ mật mã là bộ  $(P, C, K, E, D)$  thoả mãn các điều kiện sau:

- **P** là không gian bản rõ: là tập hữu hạn các bản rõ có thể có.
- **C** là không gian bản mã: là tập hữu hạn các bản mã có thể có.
- **K** là không gian khoá: là tập hữu hạn các khoá có thể có.

$$d_K(e_K(x)) = x \text{ với mọi bản rõ } x \in P.$$

Hàm giải mã  $d_k$  chính là ánh xạ ngược của hàm mã hóa  $e_k$

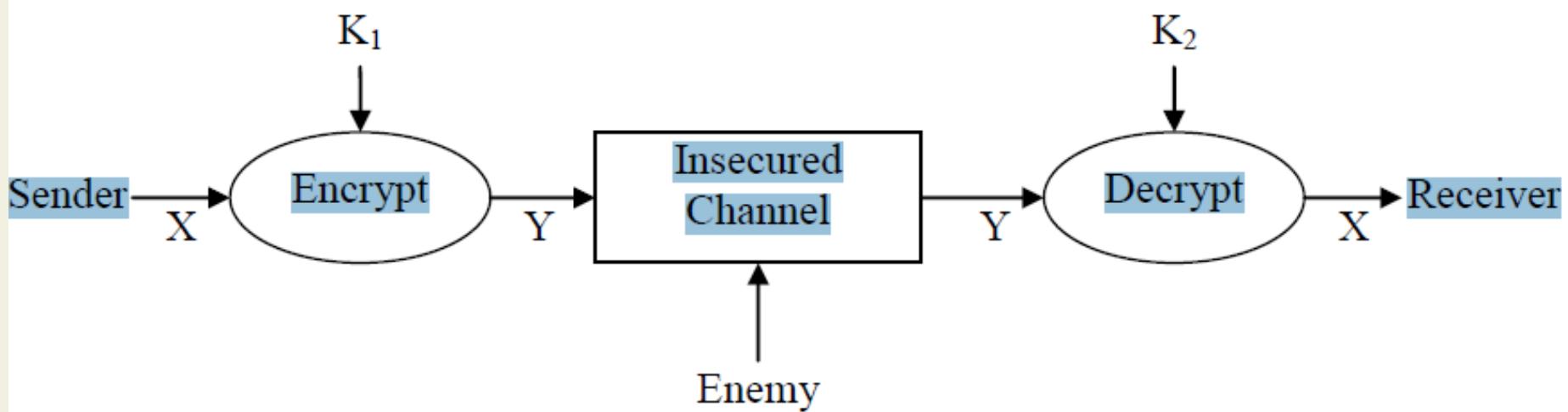
## 9. Tiêu chuẩn đánh giá hệ mật mã

- **Độ an toàn:** Một hệ mật được đưa vào sử dụng điều đầu tiên phải có độ an toàn cao.
  - Chúng phải có phương pháp bảo vệ mà chỉ dựa trên sự bí mật của các khoá, còn thuật toán thì công khai. Tại một thời điểm, độ an toàn của một thuật toán phụ thuộc:
    - Nếu chi phí hay phí tổn cần thiết để phá vỡ một thuật toán lớn hơn giá trị của thông tin đã mã hóa thuật toán thì thuật toán đó tạm thời được coi là an toàn.
    - Nếu thời gian cần thiết dùng để phá vỡ một thuật toán là quá lâu thì thuật toán đó tạm thời được coi là an toàn.
    - Nếu lượng dữ liệu cần thiết để phá vỡ một thuật toán quá lớn so với lượng dữ liệu đã được mã hoá thì thuật toán đó tạm thời được coi là an toàn
  - Bản mã C không được có các đặc điểm gây chú ý, nghi ngờ.

## 9.Tiêu chuẩn đánh giá hệ mật mã

- **Tốc độ mã và giải mã:** Khi đánh giá hệ mật mã chúng ta phải chú ý đến tốc độ mã và giải mã. Hệ mật tốt thì thời gian mã và giải mã nhanh.
- **Phân phối khóa:** Một hệ mật mã phụ thuộc vào khóa, khóa này được truyền công khai hay truyền khóa bí mật. Phân phối khóa bí mật thì chi phí sẽ cao hơn so với các hệ mật có khóa công khai. Vì vậy đây cũng là một tiêu chí khi lựa chọn hệ mật mã.

# 10. Mô hình truyền tin cơ bản của mật mã học và luật Kirchoff



Hình 1.1: Mô hình cơ bản của truyền tin bảo mật

## 10. Mô hình truyền tin cơ bản của mật mã học và luật Kirchoff

- **Theo luật Kirchoff (1835 - 1903)** (một nguyên tắc cơ bản trong mã hóa) thì: *toàn bộ cơ chế mã/giải mã trừ khoá là không bí mật đối với kẻ địch.*
- **Ý nghĩa của luật Kirchoff:** sự an toàn của các hệ mã mật không phải dựa vào sự phức tạp của thuật toán mã hóa sử dụng.

## 11. Một số ứng dụng của mã hóa trong security

Một số ứng dụng của mã hóa trong đời sống hằng ngày nói chung và trong lĩnh vực bảo mật nói riêng. Đó là:

Securing Email

Authentication System

Secure E-commerce

Virtual Private Network

Wireless Encryption

## Câu hỏi ôn tập: (20 phút)

1. Tìm hiểu luật An ninh mạng 2018: tập trung điều 2, 8, 19, 41, 42
2. Lấy ví dụ về các tấn công thụ động và chủ động?
3. Kể tên các ứng dụng của mã hóa?

# CƠ SỞ TOÁN HỌC CHO MẬT MÃ

## Chương 2: Cơ sở toán học

### • Số học đồng dư (modulo):

- Cho một số nguyên  $a$  và số nguyên dương  $n$  bất kỳ, thực hiện phép chia  $a$  cho  $n$  thì thu được thương số  $q$  và phần dư  $r$  thỏa mãn mối quan hệ sau:

$$a = q * n + r, \quad 0 \leq r < n$$

Bảng 2. 1 Minh họa thương số và phần dư khi thực hiện phép chia  $a$  cho  $n$

$a = 13$	$n = 4$	$13 = 3 \times 4 + 1$	$q = 3$	$r = 1$
$a = -13$	$n = 4$	$-13 = (-4) \times 4 + 3$	$q = -4$	$r = 3$

Tóm lại, cho một số nguyên  $a$  và số nguyên dương  $n$  thì ta định nghĩa  $a \bmod n$  là phần dư của phép chia  $a$  cho  $n$ . Ví dụ:  $13 \bmod 4 = 1$

Hai số nguyên  $a$  và  $b$  được gọi là đồng dư modulo với  $n$  nếu  $(a \bmod n) = (b \bmod n)$  và được ký hiệu như sau:  $a \equiv b \pmod{n}$ . Ví dụ  $13 \equiv 5 \pmod{4}$ .

# Chương 2: Cơ sở toán học

- Số học đồng dư (modulo): Các tính chất của đồng dư trên  $Z_n$**

Tính chất	Biểu thức
Giao hoán	$(x + y) \text{ mod } n = (y + x) \text{ mod } n$ $(x \times y) \text{ mod } n = (y \times x) \text{ mod } n$
Kết hợp	$[(x + y) + z] \text{ mod } n = [x + (y + z)] \text{ mod } n$ $[(x \times y) \times z] \text{ mod } n = [x \times (y \times z)] \text{ mod } n$
Phân phối	$[x \times (y + z)] \text{ mod } n = [(x \times y) + (x \times z)] \text{ mod } n$
Số đối ( $-x$ )	Với mỗi số nguyên $x \in Z_n$ tồn tại số $y$ sao cho $x + y \equiv 0 \text{ (mod } n)$
Identities	$(0 + x) \text{ mod } n = x \text{ mod } n$ $(1 \times x) \text{ mod } n = x \text{ mod } n$

# Chương 2: Cơ sở toán học

## • Ước số chung lớn nhất:

- Ước số chung lớn nhất của 2 số nguyên  $a$  và  $b$  là số nguyên dương lớn nhất vừa là ước của  $a$  và của  $b$ , được ký hiệu là  $\text{gcd}(a,b)$
- Hai số  $a$  và  $b$  được gọi là nguyên tố cùng nhau nếu  $\text{gcd}(a,b) = 1$
- Thuật toán Oclit tìm ước số chung lớn nhất dựa vào định lý sau: Ứng với số nguyên không âm  $a$  và số nguyên dương  $b$  bất kỳ thì:  
$$\text{gcd}(a, b) = \text{gcd} (b, a \bmod b )$$

Ví dụ:  $\text{gcd}(55,22)?$

## Chương 2: Cơ sở toán học

Đoạn chương trình sau minh họa cài đặt thuật toán Oclit để tìm ước số chung lớn nhất bằng ngôn ngữ lập trình Java.

```
int euclid(int a, int b) {  
    int r;  
    while(true) {  
        if(b==0) return a;  
        r = a%b;  
        a = b;  
        b = r;  
    }  
}
```

# Chương 2: Cơ sở toán học

**Số nguyên tố:** Số nguyên  $p > 1$  được gọi là số nguyên tố nếu nó chỉ có ước số là  $\pm 1$  và  $\pm p$ . Ví dụ 2 là số nguyên tố vì nó chỉ có các ước số là  $\pm 1$  và  $\pm 2$ .

73
79
83
89
97

2	101	211	307	401
3	103	223	311	409
5	107	227	313	419
7	109	229	317	421
11	113	233	331	431
13	127	239	337	433
17	131	241	347	439
19	137	251	349	443
23	149	257	353	449
29	151	263	359	457
31	157	269	367	461
37	163	271	373	463
41	167	277	379	467
43	173	281	383	479
47	179	283	389	487
53	181	293	397	491
59	191			499
61	193			
67	197			
71	199			

## Một số thuật toán trên Zn

### • *Tìm phần tử nghịch đảo*

Phần tử nghịch đảo của số nguyên  $a \in \mathbb{Z}_n$  là số nguyên  $x \in \mathbb{Z}_n$  sao cho:

$$a \times x \equiv 1 \pmod{n}$$

Nếu tồn tại  $x$  thì nó là duy nhất và  $a$  được gọi là khả nghịch

Để tìm phần tử nghịch đảo của  $a$  với  $n$  nhỏ thì ta có thể sử dụng bảng nhân để tìm trực tiếp. Tuy nhiên, với  $n$  lớn thì phương pháp này không khả thi

Nếu  $\gcd(a, n) = 1$  thì  $a$  là khả nghịch modulo  $n$  có nghĩa là  $a \times a^{-1} \equiv 1 \pmod{n}$

Như vậy, ta có thể mở rộng thuật toán Oclit để tìm ước số chung lớn nhất của  $a$  và  $n$ .

Nếu  $\gcd(a, n) = 1$  thì thuật toán sẽ trả về phần tử nghịch đảo của  $a$

## Một số thuật toán trên Zn

### • *Tìm phần tử nghịch đảo*

Phần tử nghịch đảo của số nguyên  $a \in \mathbb{Z}_n$  là số nguyên  $x \in \mathbb{Z}_n$  sao cho:

$$a \times x \equiv 1 \pmod{n}$$

$x$  là nghịch đảo của  $a$ , ký hiệu là  $a^{-1} \bmod n$

• Ví dụ: Tính  $6^{-1} \bmod 11 = ?$

Phải đi tìm phần tử nghịch đảo của  $a$  là  $x$ ?  $a \cdot x = 1 \pmod{11}$

Lập bảng:

x	$x \cdot 6$	$x \cdot 6 \bmod 11$
1	6	6
2	12	1

Vậy  $x=2=6^{-1} \bmod 11$

# Một số thuật toán trên Zn

- *Tìm phần tử nghịch đảo*

Ví dụ: Tính  $6^{-1} \text{ mod } 13 = ?$

Vậy  $x = ??? = 6^{-1} \text{ mod } 13$

x	$x * 6$	$x * 6 \text{ mod } 13$
1	6	6
2	12	12
3	18	5
4	24	11
5	30	4
6	36	10
7	42	3
8	48	9
9	54	2
10	60	8
11	66	1
12	72	7

## Một số thuật toán trên Zn

- *Tìm phần tử nghịch đảo*

Ví dụ: Tính  $6^{-1} \text{ mod } 8 = ?$

Vậy  $x = ??? = 6^{-1} \text{ mod } 8$

x	$x * 6$	$x * 6 \text{ mod } 8$
1	6	6
2	12	4
3	18	2
4	24	0
5	30	6
6	36	4
7	42	2

Kết luận: Không có nghịch đảo của 6 mod 8.

*Để tồn tại nghịch đảo thì a và n phải là 2 số nguyên tố cùng nhau*

Tính  $550^{-1} \text{ mod } 1759 = ?$

# Một số thuật toán trên Zn

Ta có:  $550^{-1} \bmod 1759 = ?$  Tức là tìm x sao cho:  $550x \bmod 1759 = 1$

Hay  $550x = 1759*k + 1$  hay  $550x + 1759y = 1$

Tìm x và y sao cho:  $550x + 1759y = 1$

## Thuật toán Oclit tìm USCLN

$$1) 1759 = 3 * 550 + 109$$

$$2) 550 = 5 * 109 + 5$$

$$3) 109 = 21 * 5 + 4$$

$$4) 5 = 1 * 4 + 1$$

$$5) 4 = 4 * 1 + 0$$

## Mở rộng để tìm x, y

$$1) 109 = 550 * (-3) + 1759$$

$$2) 550 = 5 * (109 = 550 * (-3) + 1759) + 5$$

$$\rightarrow 5 = 550 * 16 + 1759 * (-5)$$

$$3) 550 * (-3) + 1759 = 21 * (550 * 16 + 1759 * (-5)) + 4$$

$$\rightarrow 4 = 550 * (-339) + 1759 * 106$$

$$4) 550 * 16 + 1759 * (-5) = 1 * 550 * (-339) + 1759 * 106 + 1$$

$$\rightarrow 1 = 550 * 355 + 1759 * (-111)$$

Như vậy: nghịch đảo của  $550 \bmod 1759$  là 355

Các bước mở rộng luôn thỏa  
 $r_i = 550x_i + 1759y_i$

## Thuật toán Oclit mở rộng

Ta có:  $550^{-1} \bmod 1759 = ?$  Tức là tìm x sao cho:  $550x \bmod 1759 = 1$

Hay  $550x = 1759*k + 1$  hay  $550x + 1759y = 1$

Tìm x và y sao cho:  $550x + 1759y = 1$

Các bước mở rộng luôn thỏa

$$r_i = 550x_i + 1759y_i$$

→ Lập bảng để tìm  $r_i$ ,  $x_i$  và  $y_i$

→ i được khởi gán từ giá trị -1, ...

$$r_i = r_{i-2} \bmod r_{i-1}$$

$$q_i = \lfloor r_{i-2} / r_{i-1} \rfloor$$

$$x_i = x_{i-2} - q_i x_{i-1}$$

$$y_i = y_{i-2} - q_i y_{i-1}$$

i	$r_i$	$q_i$	$x_i$	$y_i$
-1	1759		0	1
0	550		1	0
1	109	3	-3	1
2	5	5	16	-5
3	4	21	-339	106
4	1	1	355	-111
5	0	4	-1759	550

# Một số thuật toán trên Zn

- **Tính  $a^b \ mod \ n$**
- Để tính  $a^b$  với  $a$  và  $b$  là các số nguyên dương. Nếu ta biểu diễn  $b$  thành số nhị phân  $b_k b_{k-1} \dots b_0$  thì  $b = \sum 2^i$  (với  $b_i \neq 0$ ) nên ta có:

$$a^b = a^{\sum_{b_i \neq 0} 2^i} = \prod_{b_i \neq 0} a^{2^i}$$

$$a^b \ mod \ n = \left[ \prod_{b_i \neq 0} a^{2^i} \right] \ mod \ n = \left( \left[ \prod_{b_i \neq 0} a^{2^i} \ mod \ n \right] \right) \ mod \ n$$

.

## Một số thuật toán trên Zn

**Tính  $a^b \text{ mod } n$**  : Do đó, ta có thể phát triển thuật toán bình phương và nhân để tính giá trị của  $a^b \text{ mod } n$  như đoạn mã giả sau đây:

```
MODULE calcExponent(a,b,n)
    Biểu diễn b dưới dạng nhị phân: bkbk-1...b0
    f = 1
    FOR i = k DOWNTO 0 DO
        f = (f*f) mod n
        IF bi = 1 THEN
            f = (f*a) mod n
        END_IF
    END_FOR
    RETURN f
END MODULE
```

## Một số thuật toán trên Zn

- *Tính  $a^b \text{ mod } n$*
- Ví dụ: minh họa các bước trong thuật toán bình phương và nhân để tính  $a^b$  với  $a = 7, b = 560 = 1000110000, n = 561$

$i$	9	8	7	6	5	4	3	2	1	0
$b_i$	1	0	0	0	1	1	0	0	0	0
$f$	7	49	157	526	160	241	298	166	67	1

**Số nguyên tố:** Số nguyên  $p > 1$  được gọi là số nguyên tố nếu nó chỉ có ước số là  $\pm 1$  và  $\pm p$

- **Phân tích một số ra thừa số nguyên tố**

Phân tích một số ra thừa số nguyên tố tức là viết nó dưới dạng tích lũy thừa của các số nguyên tố. Với mọi số nguyên  $a > 1$ , ta có thể phân tích nó thành tích sau:

$$a = p_1^{a_1} \times p_2^{a_2} \times \dots \times p_k^{a_k}$$

- Trong đó,  $p_1 < p_2 < \dots < p_k$  là các số nguyên tố dương.

Ví dụ:  $91 = 3 \times 17$ ;  $3600 = 2^4 \times 3^2 \times 5^2$

Ta có thể dễ dàng tìm được ước số chung lớn nhất của 2 số nguyên dương bằng cách phân tích chúng ra thừa số nguyên tố, bởi vì nếu  $k = \gcd(a, b)$  thì

$k_p = \min(a_p, b_p)$  với mọi giá trị của  $p$ .

- Ví dụ: Tìm  $\gcd(300, 18)$ ?

# Một số thuật toán trên Zn

- **Định lý Fermat**

Định lý Fermat phát biểu như sau: Nếu p là số nguyên tố và a là số nguyên dương không chia hết cho p thì:

$$a^{p-1} \equiv 1 \pmod{p}$$

- Ví dụ p = 3 là số nguyên tố, a = 5 không chia hết cho 3. Ta có  $a^{p-1} \pmod{p} = 5^2 \pmod{3} = 25 \pmod{3} = 1$ .
- Định lý Fermat có thể phát biểu cách khác như sau: Nếu p là số nguyên tố và a là số nguyên dương thì:

$$a^p \equiv a \pmod{p}$$

- Ví dụ: p = 5, a = 3 khi đó:  $a^p \pmod{p} = 3^5 \pmod{5} = 243 \pmod{5} = 3 \pmod{5} = a \pmod{p}$

# Một số thuật toán trên Zn

## *Định lý phân dư Trung Hoa*

- Trong nhiều trường hợp ta muốn tìm cách để tăng tốc độ tính toán Modulo → Các phép toán trên modulo các số nhỏ tính nhanh nhiều so với các số lớn.
- Chính vì vậy nếu số lớn phân tích được thành tích của các số nhỏ, từng cặp là nguyên tố cùng nhau .
- Giả sử ta cần tính  $A \ mod \ M$ , trong đó  $M$  là một số lớn và có thể phân tích thành tích các số nhỏ như công thức sau:

$$M = \prod_{i=1}^k m_i$$

- Trong đó  $m_i$  là cặp các số nguyên tố cùng nhau, tức là  $\gcd(m_i, m_j) = 1$ , với mọi  $i \neq j$  và  $1 \leq i, j \leq k$ .

# Một số thuật toán trên Zn

## *Định lý phần dư Trung Hoa*

- Ta có thể biểu diễn số nguyên  $A$  bất kỳ trong  $Z_M$  bởi một bộ  $k$  thành phần và các thành phần nằm trong  $Z_{m_i}$  như sau:

$$A \leftrightarrow (a_1, a_2, a_3, \dots, a_k)$$

- Trong đó:  $A \in Z_M$ ,  $a_i \in Z_{m_i}$  và  $a_i = A \bmod m_i$  với  $1 \leq i \leq k$ .

- Đặt  $M_i = \frac{M}{m_i}$  và  $c_i = M_i \times (M_i^{-1} \bmod m_i)$  với  $1 \leq i \leq k$ .

- Định lý phần dư trung hoa xác định:

$$A \bmod M = (\sum_{i=1}^k a_i \times c_i) \bmod M.$$

## Một số thuật toán trên Zn

### *Định lý phần dư Trung Hoa*

- Định lý phần dư trung hoa áp dụng cho các phép toán số học. Nếu ta có:

$$A \leftrightarrow (a_1, a_2, a_3, \dots, a_k)$$

$$B \leftrightarrow (b_1, b_2, b_3, \dots, b_k)$$

$$(A + B) \text{ mod } M \leftrightarrow ((a_1 + b_1) \text{ mod } m_1, (a_2 + b_2) \text{ mod } m_2, \dots, (a_k + b_k) \text{ mod } m_k)$$

$$(A - B) \text{ mod } M \leftrightarrow ((a_1 - b_1) \text{ mod } m_1, (a_2 - b_2) \text{ mod } m_2, \dots, (a_k - b_k) \text{ mod } m_k)$$

$$(A \times B) \text{ mod } M \leftrightarrow ((a_1 \times b_1) \text{ mod } m_1, (a_2 \times b_2) \text{ mod } m_2, \dots, (a_k \times b_k) \text{ mod } m_k)$$

m1 = 5	M	Mod 7						
m2 = 7		0	1	2	3	4	5	6
Mod 5	0	0	15	30	10	25	5	20
	1	21	1	16	31	11	26	6
	2	7	22	2	17	32	12	27
	3	28	8	23	3	18	33	13
	4	14	29	9	24	4	19	34

$$A = 16 \Leftrightarrow (1, 2)$$

$$B = 29 \Leftrightarrow (4, 1)$$

$$(A + B) \text{ MOD } 35 \Leftrightarrow (1+4, 2 + 1)$$

- $(A + B) \text{ MOD } 35 = (16 + 29) \text{ mod } 35 = 10$

- $(1+4, 2 + 1) = (0, 3)$

$$10 \Leftrightarrow (0, 3)$$

# Một số thuật toán trên Zn

## Định lý phần dư Trung Hoa

- Áp dụng định lý phần dư trung hoa để tính  $101^{59} \text{ mod } 323$

Đặt

$$A = 101^{59},$$

$$M = 323 = 17 * 19;$$

$$\rightarrow m_1 = 17, m_2 = 19.$$

Tính theo các modul thành phần

$$a_1 = 101^{59} \text{ mod } 17 = 16$$

$$a_2 = 101^{59} \text{ mod } 19 = 5$$

Tính kết quả A mod M;

Khi đó  $M_1 = 19, M_2 = 17$  và

$$M_1^{-1} \text{ mod } m_1 = 19^{-1} \text{ mod } 17 = 9,$$

$$M_2^{-1} \text{ mod } m_2 = 17^{-1} \text{ mod } 19 = 9,$$

Tính  $c_i = M_i \times (M_i^{-1} \text{ mod } m_i)$

$$c_1 = 19.9 = 171$$

$$c_2 = 17.9 = 153$$

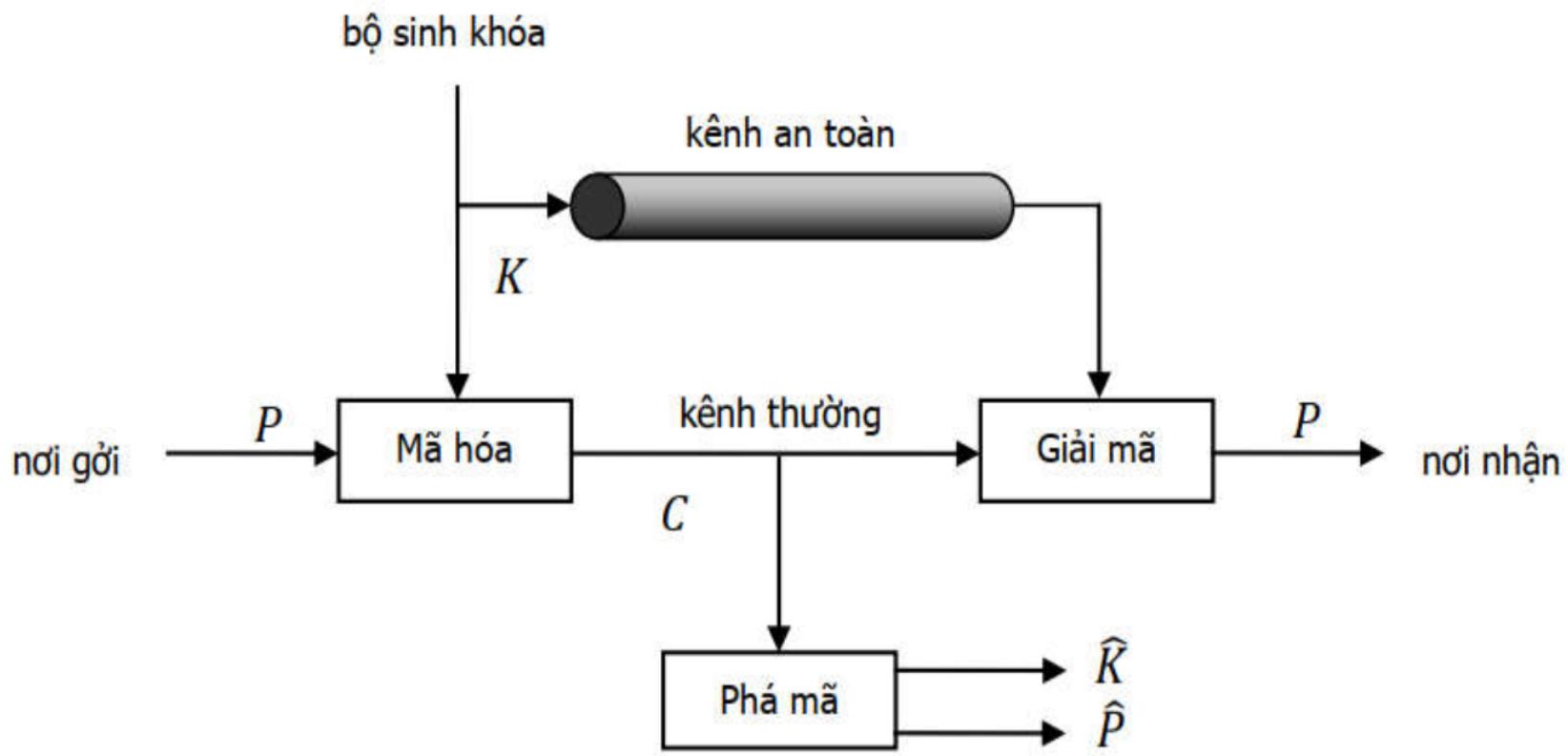
$$A = (a_1 c_1 + a_2 c_2) \text{ mod } M$$

$$= (16 * 171 + 5 * 153) \text{ mod } 323$$

$$= 3501 \text{ mod } 323 = 271$$

# Chương 3: Các hệ mã khóa bí mật

## Mô hình



# Chương 3: Các hệ mã khóa bí mật

## I. Hệ mã hóa cổ điển:

### 1. Hệ mã hóa thay thế :

Hệ mã hóa thay thế là hệ mã hóa trong đó mỗi ký tự của bản rõ được thay thế bằng ký tự khác trong bản mã (có thể là một chữ cái, một số hoặc một ký hiệu).

Có 4 kỹ thuật thay thế sau đây:

*Thay thế đơn*

*Thay thế đồng âm*

*Thay thế đa mẫu tự*

*Thay thế đa sơ đồ*

# I. Hệ mã hóa cổ điển:

- a. **Thay thế đơn**: là hệ trong đó một ký tự của bản rõ được thay bằng một ký tự tương ứng trong bản mã. Một ánh xạ 1-1 từ bản rõ tới bản mã được sử dụng để mã hoá toàn bộ thông điệp.
  
- b. **Thay thế đồng âm**: giống như hệ thống mã hoá thay thế đơn, ngoại trừ một ký tự của bản rõ có thể được ánh xạ tới một trong số một vài ký tự của bản mã: sơ đồ ánh xạ 1-n (one-to-many). Ví dụ, “A” có thể tương ứng với 5, 13, 25, hoặc 56, “B” có thể tương ứng với 7, 19, 31, hoặc 42, v.v.

# I. Hệ mã hóa cổ điển:

- c. **Thay thế đa mẫu tự**: được tạo nên từ nhiều thuật toán mã hoá thay thế đơn. Ánh xạ 1-1 như trong trường hợp thay thế đơn, nhưng có thể thay đổi trong phạm vi một thông điệp. Ví dụ, có thể có năm thuật toán mã hoá đơn khác nhau được sử dụng; đặc biệt thuật toán mã hoá đơn được sử dụng thay đổi theo vị trí cù
- d. **Thay thế đa số đồ**: là thuật toán trong đó các khối ký tự được mã hoá theo nhóm. Đây là thuật toán tổng quát nhất, cho phép thay thế các nhóm ký tự của văn bản gốc. Ví dụ, “ABA” có thể tương ứng với “RTQ”, “ABB” có thể tương ứng với “SLL”, v.v

# I. Hệ mã hóa cổ điển:

## 2. Hệ mã Caesar:

- Hệ mã Caesar là một hệ mã hoá thay thế đơn âm làm việc trên bảng chữ cái tiếng Anh 26 ký tự (A, B, ..., Z).
- Không gian các bản rõ  $P$  là *các thông điệp được tạo từ bảng chữ cái A*, không gian các bản mã  $C \equiv P$ . Giả sử số phần tử của bảng chữ cái  $|A| = N$ .
- Để mã hóa người ta đánh số các chữ cái từ 0 tới  $N-1$ .
- Không gian khóa  $k = Z_N$ . *Với mỗi khóa  $K \in k$  hàm mã hóa và giải mã một ký tự có số thứ tự là  $i$  sẽ được thực hiện như sau:*

# I. Hệ mã hóa cổ điển:

## 2. Hệ mã Caesar:

Mã hóa:  $E_k(i) = (i + k) \bmod N$ .

Giải mã:  $D_k(i) = (i - k) \bmod N$ .

- Hệ mã Caesar với bảng chữ cái tiếng Anh sẽ có  $N = 26$  chữ cái, bảng chữ cái được đánh số như sau:

Bảng 3. 1 Bảng chữ cái tiếng Anh

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

# I. Hệ mã hóa cổ điển:

## 2. Hệ mã Caesar:

**Ví dụ:** Với  $k=3$  (trường hợp đã được hoàng đế Caesar sử dụng), ký tự A được thay bằng D, B được thay bằng E, ..., W được thay bằng Z, ..., X được thay bằng A, Y được thay bằng B, và Z được thay bằng C.

Bảng chữ cái gốc:

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

Bảng chữ cái dùng để mã hóa:

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

# I. Hệ mã hóa cổ điển:

## 2. Hệ mã Caesar:

- Hệ mã Caesar sử dụng phương pháp thay thế đơn âm nên có hiện tượng gọi là phụ thuộc tần suất xuất hiện của ngôn ngữ tự nhiên.
- Trên thực tế hệ mã Caesar có số khóa ít nên hoàn toàn có thể thám mã bằng cách thử tất cả các khóa có thể (kiểu tấn công Brute force).

# I. Hệ mã hóa cổ điển:

## 3. Hệ mã Affine: cũng là hệ mã thay thế

$P = C = Z_{26}$ ,  $K = \{(a,b) \in Z_{26} \times Z_{26}, \text{ ước chung lớn nhất của } a \text{ và } 26 \text{ bằng } 1\}$ .

Với mỗi  $k \in K$  ta có:

Hàm mã hóa  $e_k(x) = ax + b \pmod{26}$

Hàm giải mã  $d_k(y) = a^{-1}(y-b) \pmod{26}$ .

Để việc giải mã có thể thực hiện được, yêu cầu cần thiết là hàm Affine phải là đơn ánh, tức là với bất kỳ  $y \in Z_{26}$ , ta muốn có đồng nhất thức sau  $ax + b \equiv y \pmod{26}$  phải có nghiệm  $x$  duy nhất

Ví dụ: Mã hóa cụm từ “HOT”

# I. Hệ mã hóa cỗ điển:

Ví dụ: Giả sử  $P = C = Z_{26}$ .

- *encryption:*  $e_k(x) = a \cdot x + b \pmod{26}$  .
- *key:*  $k = (a, b)$  where  $a, b \in Z_{26}$  .
- *decryption:*  $x = a^{-1} \cdot (y - b) \pmod{26}$  .

- $a$  và 26 nguyên tố cùng nhau:  $\gcd(a,n)=1$

# I. Hệ mã hóa cổ điển:

- Mã tuyến tính là một mã thay thế có dạng  $e(x) = ax + b \pmod{26}$ , trong đó  $a, b \in \mathbb{Z}_{26}$ .

- Giải mã: Tìm  $x$ ?

$$y = ax + b \pmod{26}$$

$$ax = y - b \pmod{26}$$

$$x = a^{-1}(y - b) \pmod{26}.$$

- Vấn đề: Tính  $a^{-1}$ .

Để có  $a^{-1}$ , đòi hỏi  $(a, 26) = 1$ .

Tính  $a^{-1}$ : Thuật toán Euclide mở rộng (lập trình để tính)

# I. Hệ mã hóa cổ điển:

## 4. Hệ mã Vigenere:

- Trong phương pháp mã hóa bằng thay thế: với một khóa  $k$  được chọn, mỗi phần tử  $x \in P$  được ánh xạ vào duy nhất một phần tử  $y \in C$ .
- Phương pháp Vigenere sử dụng khóa có độ dài  $m$ .
- Được đặt tên theo nhà khoa học Blaise de Vigenere (thế kỷ 16)
- Có thể xem phương pháp mã hóa Vigenere bao gồm  $m$  phép mã hóa bằng dịch chuyển được áp dụng luân phiên nhau theo chu kỳ
- Không gian khóa  $K$  của phương pháp Vigenere có số phần tử là  $n^m$
- Ví dụ:  $n=26$ ,  $m=5$  thì không gian khóa  $\sim 1.1 \times 10^7$

# I. Hệ mã hóa cổ điển:

## 5. Hệ mã Hill:

-Phương pháp Hill (1929)

-Tác giả: Lester S. Hill

-Ý tưởng chính:

Sử dụng  $m$  tổ hợp tuyến tính của  $m$  ký tự trong plaintext để tạo ra  $m$  ký tự trong ciphertext

-Ví dụ:

$$\begin{aligned}y_1 &= 11x_1 + 3x_2 \\y_2 &= 8x_1 + 7x_2.\end{aligned}$$

$$(y_1, y_2) = (x_1, x_2) \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$$

# I. Hệ mã hóa cổ điển:

Chọn số nguyên dương  $m$ . Định nghĩa:

$P = C = (\mathbb{Z}_n)^m$  và  $K$  là tập hợp các ma trận  $m \times m$  khả nghịch

Với mỗi khóa  $k = \begin{pmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,m} \\ k_{2,1} & \cdots & \cdots & k_{2,m} \\ \vdots & \vdots & & \vdots \\ k_{m,1} & k_{m,2} & \cdots & k_{m,m} \end{pmatrix} \in K$ , định nghĩa:

$e_k(x) = xk = (x_1, x_2, \dots, x_m) \begin{pmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,m} \\ k_{2,1} & \cdots & \cdots & k_{2,m} \\ \vdots & \vdots & & \vdots \\ k_{m,1} & k_{m,2} & \cdots & k_{m,m} \end{pmatrix}$  với  $x = (x_1, x_2, \dots, x_m) \in P$

và  $d_k(y) = yk^{-1}$  với  $y \in C$ .

Mọi phép toán số học đều được thực hiện trên  $\mathbb{Z}_n$ .

# I. Hệ mã hóa cổ điển:

Ví dụ: cho hệ mã Hill có  $M = 2$  (khóa là các ma trận vuông cấp 2) và bảng chữ cái là bảng chữ cái tiếng Anh, tức là  $N = 26$ . Cho khóa

$$K = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}$$

Hãy mã hóa xâu  $P = \text{"HELP"}$  và giải mã ngược lại bản mã thu được.

# I. Hệ mã hóa cỗ điển:

Để mã hóa chúng ta chia xâu bản rõ thành hai vecto hàng 2 chiều “HE” (7 4) và “LP” (11 15) và tiến hành mã hóa lần lượt.

$$\text{Với } P_1 = (7 \ 4) \text{ ta có } C_1 = P_1 * K = (7 \ 4) \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} = (3 \ 15) = (\text{D P})$$

$$\text{Với } P_2 = (11 \ 15) \text{ ta có } C_2 = P_2 * K = (11 \ 15) \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} = (11 \ 4) = (\text{L E})$$

Vậy bản mã thu được là C = “DPLE”.

# I. Hệ mã hóa cổ điển:

Để giải mã ta tính khóa giải mã là ma trận nghịch đảo của ma trận khóa trên  $Z_{26}$  theo công thức sau:

Với  $K = \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix}$  và  $\det(K) = (k_{11}*k_{22} - k_{21}*k_{12}) \text{ mod } N$  là một phần tử có phần tử

nghịch đảo trên  $Z_N$  (ký hiệu là  $\det(K)^{-1}$ ) thì khóa giải mã sẽ là

$$K^{-1} = \det(K)^{-1} * \begin{pmatrix} k_{22} & -k_{12} \\ -k_{21} & k_{11} \end{pmatrix}$$

Áp dụng vào trường hợp trên ta có  $\det(K) = (15 - 6) \text{ mod } 26 = 9$ .  $\text{GCD}(9, 26) = 1$  nên áp dụng thuật toán O'clit mở rộng tìm được  $\det(K)^{-1} = 3$ . Vậy  $K^{-1} = 3 * \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix}$ .

$$\begin{pmatrix} 5 & 23 \\ 24 & 3 \end{pmatrix} = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix}.$$

# I. Hệ mã hóa cổ điển:

Giải mã  $C = "DP" = (3 \ 15)$ ,  $P = C * K^{-1} = (3 \ 15) * \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix} = (3 \ 15) = "HE"$ .

Tương tự giải mã xâu  $C = "LE"$  kết quả sẽ được bản rõ  $P = "LP"$ .

Chú ý là trong ví dụ trên chúng ta sử dụng khóa  $K$  có kích thước nhỏ nên dễ dàng tìm được khóa để giải mã còn trong trường hợp tổng quát điều này là không dễ dàng.

# I. Hệ mã hóa cổ điển:

## 6. Hệ mã đổi chỗ (transposition cipher)

Một hệ mã hóa đổi chỗ là hệ mã hóa trong đó các ký tự của bản rõ vẫn được giữ nguyên, nhưng thứ tự của chúng được đổi chỗ cho nhau.

Ví dụ: một hệ mã hóa đổi chỗ cột đơn giản

Bản rõ: COMPUTER GRAPHICS MAY BE SLOW BUT AT LEAST IT'S EXPENSIVE

COMPUTERGR

APHICSMAYB

ESLOWBUTAT

LEASTITSEX

PENSIVE

Bản mã: CAELPOPSEEMHLANPIOSSUCWTITSBIUEMUTERATSGYAEERTX

# I. Hệ mã hóa cổ điển:

## Các kỹ thuật đổi chỗ:

1. *Đảo ngược toàn bộ bản rõ*: nghĩa là bản rõ được viết theo thứ tự ngược lại để tạo ra bản mã.

Ví dụ: bản rõ “TRANSPOSITION CIPHER” được mã hoá thành “REHPICNOITISOPSNART”.

Nhận xét: Đây là phương pháp mã hoá đơn giản nhất vì vậy không đảm bảo an toàn.

2. *Mã hóa theo mẫu hình học*: Bản rõ được sắp xếp lại theo một mẫu hình học nào đó, thường là một mảng hoặc một ma trận hai chiều.

# I. Hệ mã hóa cổ điển:

Ví dụ: bản rõ “LIECHTENSTEINER” được viết thành ma trận  $3 \times 5$  theo hàng như sau:

Cột	1	2	3	4	5
Bản rõ	L	I	E	C	H
	T	E	N	S	T
	E	I	N	E	R

Nếu lấy các ký tự ra theo số thứ tự cột 2, 4, 1, 3, 5 thì sẽ có bản mã “IEICSELTEENNHNTR”.

# I. Hệ mã hóa cổ điển:

3. Hoán vị các ký tự của bản rõ theo chu kỳ cố định  $d$ : Nếu hàm  $f$  là một hàm hoán vị của một khối gồm  $d$  ký tự được biểu diễn bởi  $K(d,f)$

Bản rõ:

$$M = m_1 m_2 \dots m_d m_{d+1} \dots m_{2d}$$

Với  $m_i$  là các ký tự , và bản rõ sẽ được mã hóa thành

$$Ek(M) = m_{f(1)} m_{f(2)} \dots m_{f(d)} m_{f(d)+1} \dots m_{d+f(d)}$$

Trong đó  $m_{f(1)} m_{f(2)} \dots m_{f(d)}$  là một hoán vị của  $m_1 m_2 \dots m_d$ .

# I. Hệ mã hóa cổ điển:

Ví dụ: giả sử  $d=5$  và f hoán vị dãy  $i=12345$  thành  $f(i)=35142$

Vị trí đầu	Vị trí hoán vị	Tù	Mã hóa
1	3	G	O
2	5	R	P
3	1	O	G
4	4	U	U
5	2	P	R

- **Mật mã Playfair** là một hệ mã hóa nhiều chữ, giảm bớt tương quan giữa văn bản mã hóa và nguyên bản bằng cách mã hóa đồng thời nhiều chữ cái của nguyên bản.
- Cơ chế hoạt động như sau: sử dụng một ma trận chữ cái  $5 \times 5$  trên cơ sở một từ khóa: điền các chữ cái của từ khóa (bỏ các chữ trùng), điền những vị trí còn lại của ma trận với các chữ cái khác của bảng chữ cái; I, J có thể ở trên cùng một ô của ma trận.

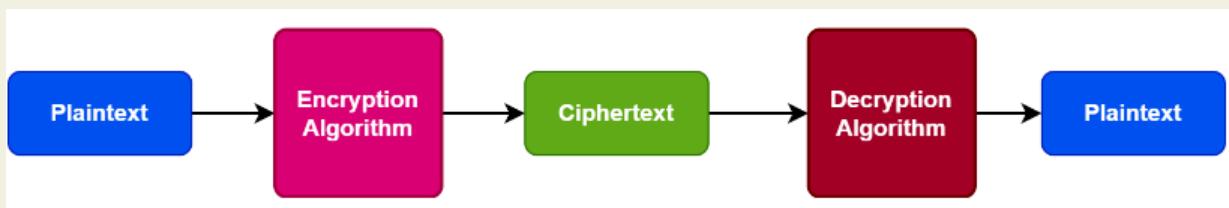
- Ví dụ ma trận với từ khóa
- MONARCHY
- M O N A R C H Y B D E F G I/J K L P Q S T U V W X Z
- • Mã hóa 2 chữ cái một lúc
  - Nếu 2 chữ giống nhau, tách ra bởi 1 chữ điền thêm thường là X hoặc Q Ví dụ: EE sẽ được thay bởi EX
  - Nếu 2 chữ nằm cùng hàng, thay bởi các chữ bên phải Ví dụ: EF sẽ thay bằng FG
  - Nếu 2 chữ nằm cùng cột, thay bởi các chữ bên dưới Ví dụ: OF thay bằng HP
  - Các trường hợp khác, mỗi chữ cái được thay bởi chữ cái khác cùng hàng, trên cột chữ cái cùng cặp Ví dụ: ET sẽ thay bằng KL

# MÃ HÓA DES

## I. Mã hóa (Nhắc lại)

### 1. Giới thiệu chung về mật mã học (Cryptography)

- Mật mã học là một lĩnh vực liên quan đến các kỹ thuật ngôn ngữ và toán học để đảm bảo an toàn thông tin, cụ thể là trong thông tin liên lạc.
- Mật mã học gắn liền với quá trình mã hóa tức là chuyển đổi thông tin từ dạng "có thể hiểu được" thành dạng "không thể hiểu được" hay chuyển đổi thông tin từ "bản rõ – plain text" sang "bản mã – cipher text" và ngược lại là quá trình giải mã



## I. Mã hóa (nhắc lại)

### 1. Giới thiệu chung về mật mã học (Cryptography)

Mật mã học giúp bảo đảm các yếu tố sau cho dữ liệu:

- **Tính bí mật (confidentiality):** thông tin chỉ được tiết lộ cho những ai được phép
- **Tính toàn vẹn (integrity):** thông tin không thể bị thay đổi mà không bị phát hiện.
- **Tính xác thực (authentication):** người gửi (hoặc người nhận) có thể chứng minh đúng họ.
- **Tính chống chối bỏ (non-repudiation):** người gửi hoặc nhận sau này không thể chối bỏ việc đã gửi hoặc nhận thông tin.

## I. Mã hóa (nhắc lại)

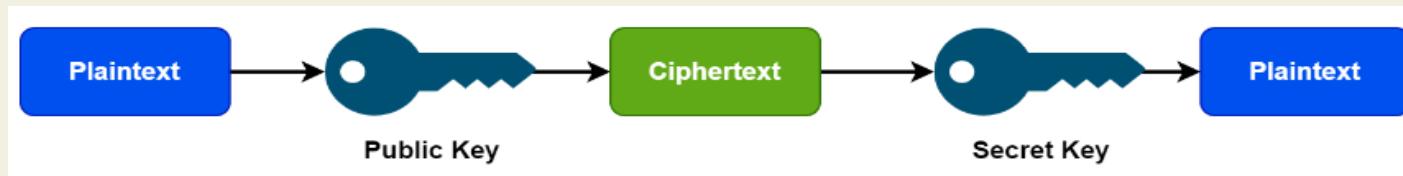
### 1. Giới thiệu chung về mật mã học (Cryptography)

- **Phân loại:**

- Loại thao tác dùng để chuyển bản rõ thành bản mã: thay thế, chuyển vị
- Số khóa sử dụng: Khóa đơn – khóa bí mật (Mã hóa đối xứng); và Hai khóa – Khóa công khai (Mã hóa bất đối xứng)
- Cách xử lý bản rõ: Mã hóa khối và mã hóa luồng



Mã hóa đối xứng



Mã hóa bất đối xứng

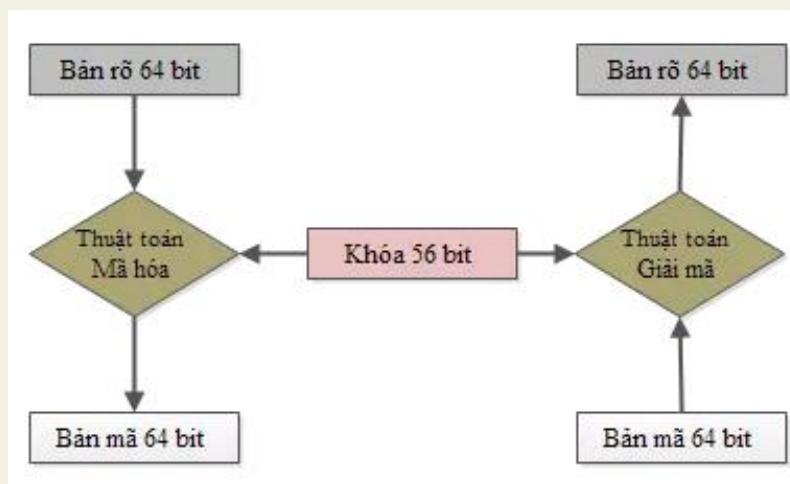
## I. Mã hóa (Nhắc lại)

### 2. Thám mã (cryptanalysis)

- Thám mã hay còn gọi là phân tích mật mã – đây là ngành học nghiên cứu các phương thức để thu được ý nghĩa của thông tin đã được mã hóa
- Các phương pháp tấn công thám mã:
  - Tìm khóa vét cạn
  - Phân tích thống kê
  - Phân tích toán học

## 2.1. Mật mã DES (Data Encryption Standard)

- Ngày 13/5/1973 ủy ban quốc gia về tiêu chuẩn của Mỹ công bố yêu cầu về mật mã áp dụng cho toàn quốc → sự ra đời của DES
- Ban đầu DES được phát triển từ hệ mã Lucifer bởi công ty IBM, năm 1975
- Sau đó DES được xem như là chuẩn mã hóa dữ liệu cho các ứng dụng



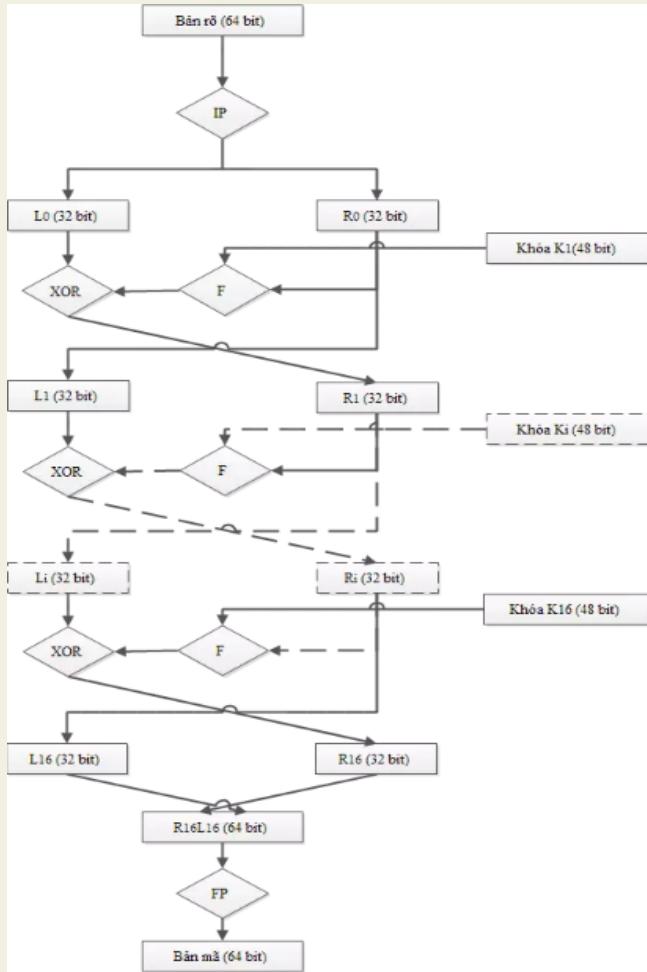
## 2.3. Mật mã DES (Data Encryption Standard)

- **Đặc điểm của thuật toán DES như sau:**
- DES là một thuật toán mã hóa khối, độ dài mỗi khối là 64 bít
- Khóa dùng trong DES có độ dài toàn bộ 64 bít. Tuy nhiên chỉ có 56 bít thực sự được sử dụng, 8 bít còn lại chỉ dùng cho việc kiểm tra
- DES xuất ra bản mã 64 bít
- Thuật toán thực hiện 16 vòng lặp, chỉ khác nhau về khóa trong mỗi vòng lặp đó
- Mã hóa và giải mã được sử dụng cùng một khóa

## 2.1. Mật mã DES (Data Encryption Standard)

- **Sơ đồ khái quát thuật toán DES**
- Với mỗi khóa K và bản rõ x, quá trình lập mã diễn ra như sau:
  - Ban đầu, dùng một phép hoán vị IP (Initial Permutation), từ x với 64 bít sẽ biến thành một từ mới  $IP(x)$ , từ này được chia thành 2 nửa  $L_0$  và  $R_0$ , mỗi nửa là một từ 32 bít
  - Từ cặp  $(L_0, R_0)$  sẽ dùng 15 lần những phép toán giống nhau để liên tiếp được các cặp  $(L_1, R_1), \dots, (L_{15}, R_{15})$ , sau đó dùng phép hoán vị nghịch đảo  $IP^{-1}$  cho từ đảo ngược  $R_{15}L_{15}$  ta sẽ được bản mã y tương ứng.

## 2.1. Mật mã DES (Data Encryption Standard)



- Thông tin đầu vào là 64 bít, được chia thành 2 khối trái (L) và phải (R)
- Từ khóa 56 bít tạo ra các khóa con (subkey) gọi là Ki.
- Hàm f là một hàm hoán vị
- Trong quá trình mã hóa, dữ liệu đầu vào phải thực hiện quá trình hoán vị đầu IP (initial permutation) và hoán vị cuối (final permutation) sau vòng thứ 16
- Hàm cơ sở f cho phép đảm bảo tính bảo mật trong DES
- Cấu trúc vòng lặp DES thực hiện theo công thức sau:

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

- Trong đó  $(L_i, R_i)$  là nửa trái và nửa phải lấy được của phép biến đổi vòng lặp thứ i

## 2.1. Mật mã DES (Data Encryption Standard)

Từ  $L_0$  và  $R_0$  sẽ lặp 16 vòng, tại mỗi vòng tính:

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

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \quad \text{với } i = 1, 2, \dots, 16$$

với:

$\oplus$  là phép XOR của hai xâu bit:

$$0 \oplus 0 = 0, \quad 1 \oplus 1 = 0$$

$$1 \oplus 0 = 1, \quad 0 \oplus 1 = 1$$

$f$  là hàm mà ta sẽ mô tả sau.

$K_i$  là các xâu có độ dài 48 bit được tính như là các hàm của khóa  $K$ .

## II. Mã hóa bí mật

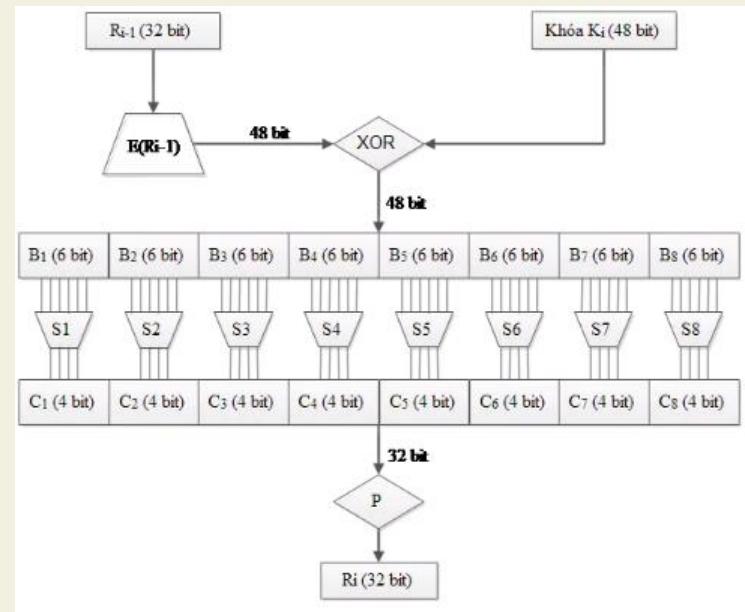
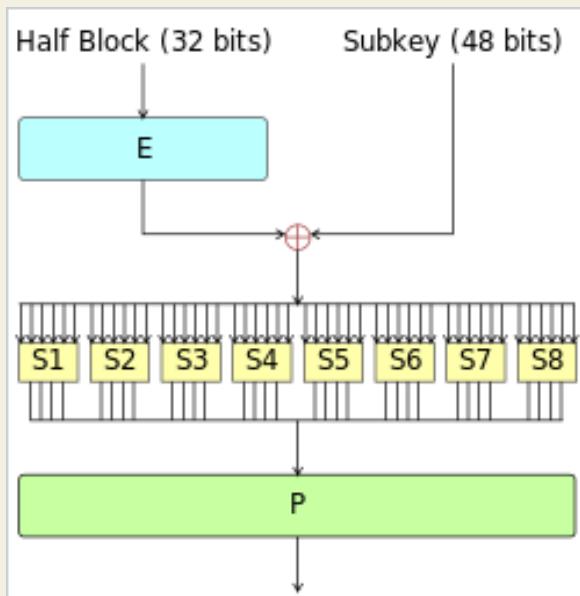
### 2.1. Mật mã DES (Data Encryption Standard)

- IP là một phép hoán vị vị trí của các ký tự trong mỗi từ 64 bít, từ vị trí thứ nhất đến vị trí thứ 64.
- Bảng dưới đây cho ta phép hoán vị IP, với cách biểu diễn là bít thứ nhất của IP(x) là bít thứ 58 của từ x (có 64 bít), bít thứ hai của IP(x) là bít thứ 50 của x,...
- Bảng của phép hoán vị  $IP^{-1}$  cũng được hiểu tương tự

IP									IP <sup>-1</sup>								
58	50	42	34	26	18	10	2		40	8	48	16	56	24	64	32	
60	52	44	36	28	20	12	4		39	7	47	15	55	23	63	31	
62	54	46	38	30	22	14	6		38	6	46	14	54	22	62	30	
64	56	48	40	32	24	16	8		37	5	45	13	53	21	61	29	
57	49	41	33	25	17	9	1		36	4	44	12	52	20	60	28	
59	51	43	35	27	19	11	3		35	3	43	11	51	19	59	27	
61	53	45	37	29	21	13	5		34	2	42	10	50	18	58	26	
63	55	47	39	31	23	15	7		33	1	41	9	49	17	57	25	

## 2.1. Mật mã DES (Data Encryption Standard)

- Sơ đồ hàm f (Feistel function):
  - Hàm f lấy đầu vào là hai từ: R có 32 bít và K có 48 bít và có kết quả ở đầu ra là từ  $f(R, K)$  có 32 bít, được xác định bởi sơ đồ sau:

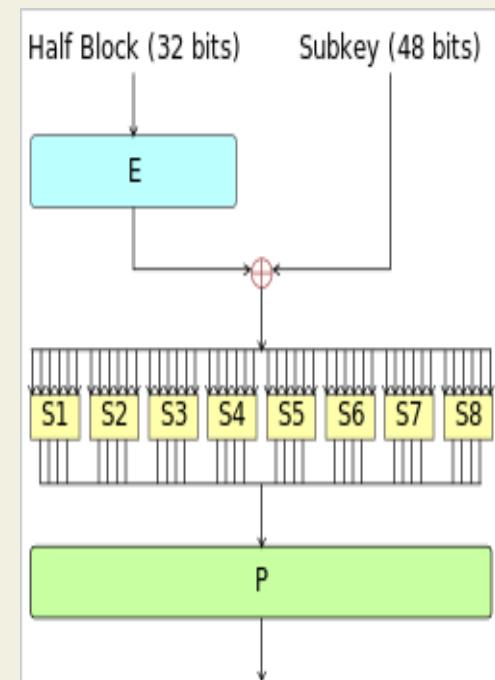


## 2.1. Mật mã DES (Data Encryption Standard)

- **Hàm E (Extension):** Là một phép hoán vị “mở rộng” theo nghĩa là nó biến mỗi từ R 32 bít thành từ E(R) bằng các hoán vị 32 bít của R nhưng có một số cặp bít được lặp lại để E(R) thành một từ có 48 bít.
- Cụ thể phép hoán vị “mở rộng” đó được cho bởi bảng sau:

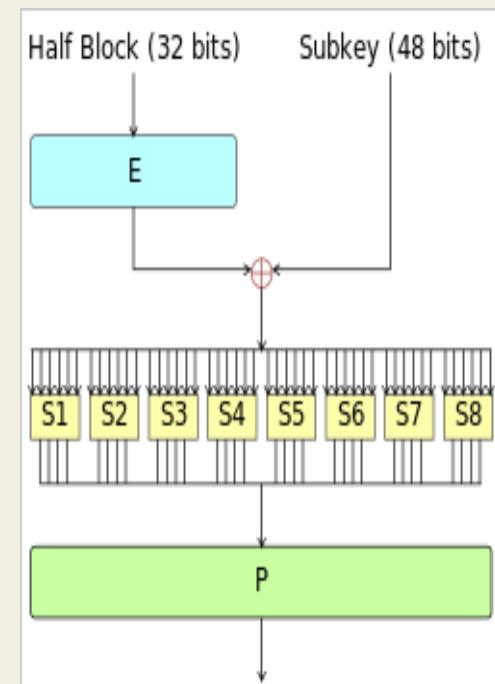
Phép hoán vị “mở rộng” E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

- Như vậy mỗi từ R= $a_1a_2\dots a_{32}$  sẽ biến thành  $E(R)=a_{32}a_1a_2a_3a_4a_5a_4a_5a_6\dots a_{30}a_{31}a_{32}a_1$



## 2.1. Mật mã DES (Data Encryption Standard)

- Sau khi thực hiện E,  $E(R)$  sẽ được cộng (từng bít theo mod2) với K, được một từ 48 bít, chia thành 8 khối (6 bít)
- Mỗi hộp  $S_i$  ( $i=1,..8$ ) là một phép thay thế, biến mỗi từ  $B_j$  6 bít thành một từ  $C_j$  4 bít; các hộp  $S_i$  được cho bởi bảng dưới đây với cách biểu diễn như sau:
  - Cụ thể phép hoán vị “mở rộng” đó được cho bởi bảng sau:
  - Mỗi từ  $B_j=b_1b_2b_3b_4b_5b_6$  ứng với một vị trí  $(r,s)$  ở hàng thứ r và cột thứ s trong bảng, các hàng được đánh số thứ tự từ 0 đến 3 với biểu diễn nhị phân  $b_1b_6$  và các cột được đánh số thứ tự từ 0 đến thứ 15 ứng với biểu diễn nhị phân  $b_2b_3b_4b_5$ .
  - Nghĩa là  $r = b_1b_6$ ;  $s = b_2b_3b_4b_5$  (từ nhị phân chuyển sang thập phân)



## 2.1. Mật mã DES (Data Encryption Standard)

Ví dụ:

$$S_1(101110) = 11_d = 1011_b \text{ (hàng } r=10_b + 1=3, \text{ cột } s=0111_b + 1 = 8\text{)}$$

$$S_2(011000) = 12_d = 1100_b \text{ (hàng } r=00_b + 1=1, \text{ cột } s=1100_b + 1 = 13\text{)}$$

$$S_3(100110) = ?$$

S5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	0	14	2	13	6	15	0	9	10	4	5	3
S6	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	S8												
1	13	0	11	0	13	2	8	4	6	15	11	1	10	9	3	14
2	1	4	11	1	1	15	13	8	10	3	7	4	12	5	6	11
3	6	11	13	2	7	11	4	1	9	12	14	2	0	6	10	13
				3	2	1	14	7	4	10	8	13	15	12	9	0
					5	6	11									

S1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	3	0	6	9	10	1	2	8	5	11	12	4	15		
1	11	5	6	15	0	3	4	7	2	12	1	10	14	9		
2	9	0	12	11	7	13	15	1	3	14	5	2	8	4		
3	0	6	10	1	13	8	9	4	5	11	12	7	2	14		

## 2.1. Mật mã DES (Data Encryption Standard)

Phép hoán vị P trong sơ đồ của hàm f được cho ở bảng dưới đây:

Mỗi 4 bít đầu ra của các hộp S-box sẽ được ghép lại, theo thứ tự các hộp và được đưa vào hộp P-box. P đơn giản chỉ là phép hoán vị các bít với nhau.

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

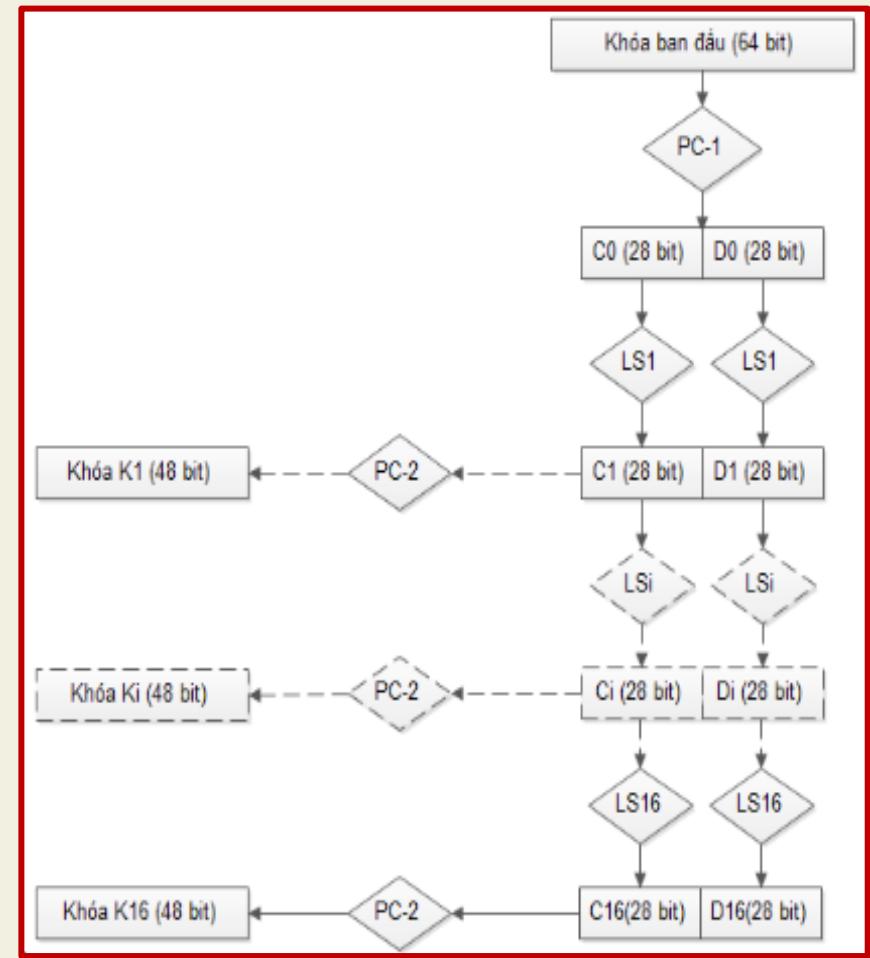
Như vậy hàm f được xác định hoàn toàn

## 2.1. Mật mã DES (Data Encryption Standard)

- Thuật toán sinh khóa  $K_i$ :

$K_1, K_2, \dots, K_{16}$

- Các khóa con đều được sinh ra từ khóa chính của DES bằng thuật toán sinh khóa con (thuật toán G)
  - LS: left shift
  - Khóa mật mã K là một từ 56 bit, ta chia thành 8 khối, mỗi khối 7 bit, ta cho thêm mỗi khối 7 bit đó một bit kiểm tra tính chẵn lẻ vào vị trí cuối để được một từ 64 bit, ta vẫn ký hiệu là K.



## 2.3. Mật mã DES (Data Encryption Standard)

- Trước tiên, thuật toán PC-1 biến K thành một từ 56 bít, ta chia thành 2 nửa C0, D0.
- Phép hoán vị PC-1 được xác định bởi bảng sau đây

57	49	41	33	25	17	9	1
58	50	42	34	25	18	10	2
59	51	43	35	27	19	11	3
60	52	44	36	63	55	47	39
31	23	15	7	62	54	46	38
30	22	14	6	61	53	45	37
29	21	13	5	28	20	12	4

Chú ý: trong bảng không có các số 8,16,24,32,40,48,56,64 là vị trí của những bít được thêm vào khi hình thành từ mới K

## 2.1. Mật mã DES (Data Encryption Standard)

- Ls<sub>i</sub>, i=1,2,...,16 là phép chuyển dịch vòng sang trái:  
VD: 00000100 dịch trái 2 bít thành 00010000
- Chuyển dịch một vị trí nếu i=1,2,9,16
- Chuyển dịch hai vị trí với giá trị i còn lại

Vòng lặp	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Số lần dịch trái	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

- Phép hoán vị PC2 biến mỗi từ 56 bít CiDi thành từ 48 bít Ki theo bảng dưới đây

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

## Kết luận DES:

- Thuật toán mã hóa E:
  - $y=E(K,x)$  với mỗi khóa  $K(K_1, K_2, \dots, K_{16})$  với bản rõ  $x$
- Thuật toán giải mã D:
  - $x=D(K,y)$  được thực hiện bằng cùng một quá trình tính toán như quá trình mã hóa, chỉ khác là thứ tự dùng khóa K sẽ là  $K_{16}, K_{15}, \dots, K_2, K_1$ .
- Độ an toàn DES: 30 năm đầu sau khi công bố → khá an toàn
  - Với tốc độ xử lý của siêu máy tính thì với độ dài khóa chỉ 56 bit → tính an toàn bị phá vỡ

Table 2.2 Average Time Required for Exhaustive Key Search

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at $10^9$ Decryptions/s	Time Required at $10^{13}$ Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	$2^{55} \text{ ns} = 1.125 \text{ years}$	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	$2^{127} \text{ ns} = 5.3 \times 10^{21} \text{ years}$	$5.3 \times 10^{17} \text{ years}$
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	$2^{167} \text{ ns} = 5.8 \times 10^{33} \text{ years}$	$5.8 \times 10^{29} \text{ years}$
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	$2^{191} \text{ ns} = 9.8 \times 10^{40} \text{ years}$	$9.8 \times 10^{36} \text{ years}$
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	$2^{255} \text{ ns} = 1.8 \times 10^{60} \text{ years}$	$1.8 \times 10^{56} \text{ years}$

## Kết luận về DES:

### *Không gian khóa*

- DES có  $2^{56} = 10^{17}$  khoá
- Nếu biết được một cặp “tin/mã” có thể thử tất cả  $10^{17}$  khả năng này để tìm ra khoá cho kết quả khớp nhất.
- Nếu một phép thử  $10^{-6}$ s thì sẽ mất  $10^{11}$ s tức là 7300 năm
- Vào năm 1976 và 1977, Diffie và Hellman đã ước lượng rằng có thể chế tạo được một máy tính chuyên dụng để vét cạn không gian khoá DES trong  $\frac{1}{2}$  ngày với cái giá 20 triệu đô la
- Đến năm 1990, hai nhà toán học người Do Thái - Biham và Shamir - đã phát minh ra phương pháp phá mã vi sai, đây là một kỹ thuật sử dụng những phỏng đoán khác nhau trong bản rõ để đưa ra những thông tin trong bản mã

## Kết luận về DES

**Tình bù:**

Nếu ta ký hiệu  $\bar{U}$  Là phần tử bù của U ( ví dụ 0100101 là phần bù của 1011010 ) thì DES có tính chất sau:

$$y = \text{DES}(x, k) \rightarrow \bar{y} = \text{DES}(\bar{x}, \bar{k})$$

=> Nếu biết bản mã y, bản rõ x và khóa k thì biết  $\bar{y}, \bar{x}, \bar{k}$

Do tính bù, ta có thể giảm độ phức tạp của tấn công duyệt toàn bộ xuống 2 lần (tương ứng với 1 bít) với điều kiện là ta có thể lựa chọn bản rõ.

## Khóa yếu

Khoá yếu là các khoá mà theo thuật toán sinh khoá con thì tất cả 16 khoá con đều như nhau:

$$K_1 = K_2 = \dots = K_{15} = K_{16}$$

=> Việc mã hóa và giải mã đối với khoá yếu là giống hệt nhau

Khoá yếu (Hex)				$C_0$	$D_0$
0101	0101	0101	0101	$\{0\}^{28}$	$\{0\}^{28}$
FEFE	FEFE	FEFE	FEFE	$\{1\}^{28}$	$\{1\}^{28}$
1F1F	1F1F	0E0E	0E0E	$\{0\}^{28}$	$\{1\}^{28}$
E0E0	E0E0	F1F1	F1F1	$\{1\}^{28}$	$\{0\}^{28}$

## Khóa yếu (tt)

Đồng thời còn có 6 cặp khoá nửa yếu (semi-weak key) khác với thuộc tính như sau:

$$y = \text{DES}(x, k_1) \text{ và } y = \text{DES}(x, k_2)$$

Nghĩa là với 2 khoá khác nhau nhưng mã hoá ra cùng một bản mã từ cùng một bản rõ:

C <sub>0</sub>	D <sub>0</sub>	Semi-weak key (Hex)								C <sub>0</sub>	D <sub>0</sub>
{01} <sup>14</sup>	{01} <sup>14</sup>	01FE	01FE	01FE	01FE	FE01	FE01	FE01	FE01	{10} <sup>14</sup>	{10} <sup>14</sup>
{01} <sup>14</sup>	{10} <sup>14</sup>	1FE0	1FE0	0EF1	0EF1	E01F	E01F	F10E	F10E	{10} <sup>14</sup>	{01} <sup>14</sup>
{01} <sup>14</sup>	{0} <sup>28</sup>	01E0	01E0	01F1	01F1	E001	E001	F101	F101	{10} <sup>14</sup>	{0} <sup>28</sup>
{01} <sup>14</sup>	{1} <sup>28</sup>	1FFE	1FFE	0EFE	0EFE	FE1F	FE1F	FE0E	FE0E	{10} <sup>14</sup>	{1} <sup>28</sup>
{0} <sup>28</sup>	{01} <sup>14</sup>	011F	011F	010E	010E	1F01	1F01	0E01	0E01	{0} <sup>28</sup>	{10} <sup>14</sup>
{1} <sup>28</sup>	{01} <sup>14</sup>	E0FE	E0FE	F1FE	F1FE	FEE0	FEE0	FEF1	FEF1	{1} <sup>28</sup>	{10} <sup>14</sup>

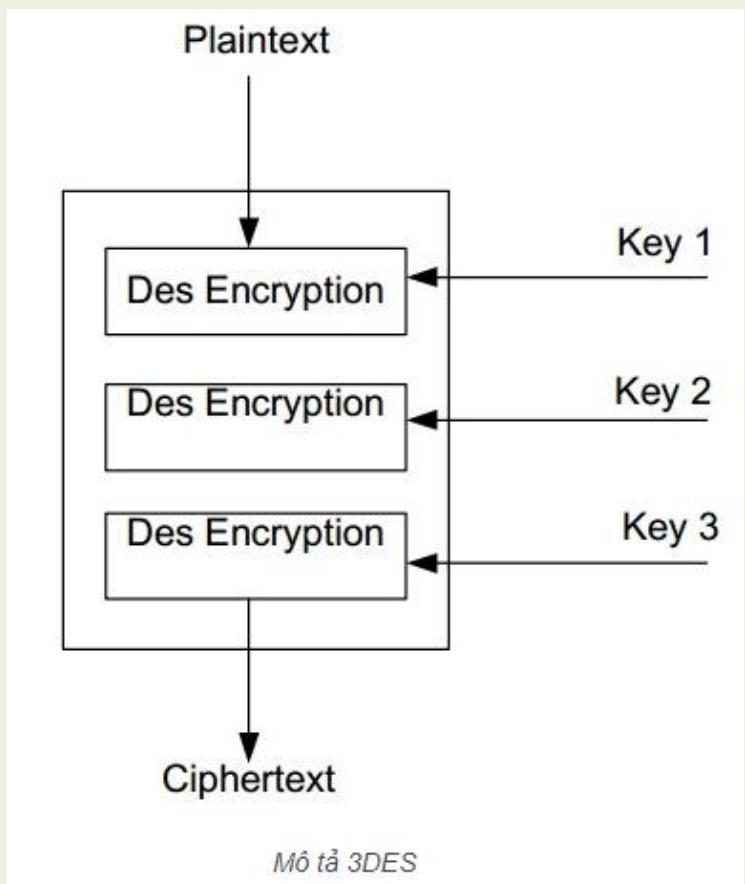
## 2.1 Mật mã DES (Data Encryption Standard)

- **Bài tập áp dụng:**

- Cho bản rõ mang nội dung:  
 $x = '0123456789ABCDEF'$ ; khóa  
 $K = 13345799BBCDDFF1$
- Trong hệ cơ số 16, thực hiện mã hóa văn bản rõ trên  
theo thuật toán DES

## 2.2. Mật mã 3-DES (Triple DES)

- Thuật toán mã hoá 3DES gồm 3 chìa khoá 64 bit, tức là toàn bộ chiều dài khoá là 192 bit: 03 khoá DES là  $K_1$ ,  $K_2$  và  $K_3$ .
- Thủ tục mã hoá cũng tương tự DES nhưng nó được lặp lại 3 lần tức là tăng lên 3 lần DES. Dữ liệu được mã hoá với chìa khoá đầu tiên, và được giải mã với chìa khoá 2, sau đó mã hoá lần nữa với chìa khoá thứ 3 để thu được dữ liệu mã hoá cuối cùng.
- Các mẫu hoạt động của 3DES:
  - Triple ECB (Triple Electronic Code Book): Sách mã hoá điện tử.
  - Triple CBC (Triple Cipher Chaining): Móc nối khối ký số.



## 2.2. Mật mã 3-DES (Triple DES)

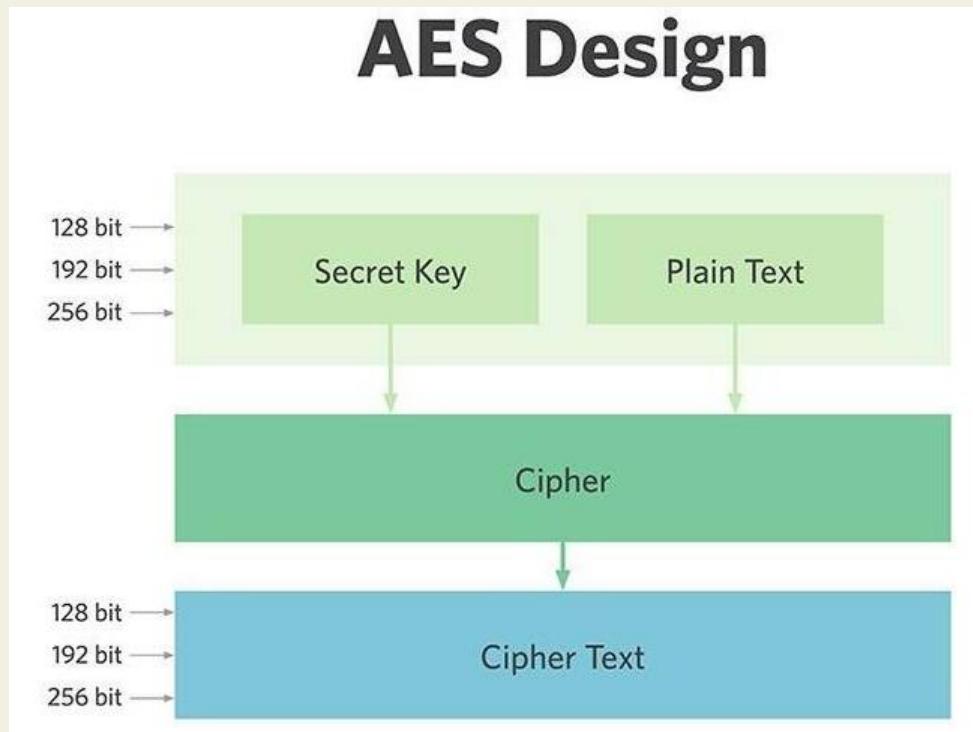
### • *Ưu và nhược điểm của 3DES*

- **Ưu điểm:** Khác với DES, thuật toán mã hoá 3DES được mã hoá 3 lần DES với kích cỡ không gian khoá 168 bit cho nên an toàn hơn rất nhiều so với DES.
- **Nhược điểm:** Vì 3DES sử dụng 3 lần mã hoá DES cho nên tốc độ mã hoá sẽ chậm hơn rất nhiều so với DES. Phần mềm ứng dụng tỏ ra rất chậm đối với hình ảnh số và một số ứng dụng dữ liệu tốc độ cao vì kích thước khối 64 bit vẫn còn là một nhược điểm đối với những hệ thống hiện nay.

## 2.3. Mật mã AES (Advanced Encryption Standard )

- Được công bố lần đầu năm 1997 bởi NIST
- AES được nghiên cứu và phát triển để thay thế cho DES
- NIST tuyên bố AES là giải pháp tốt nhất để bảo vệ thông tin nhạy cảm cho chính phủ (Mỹ) trong thế kỷ 21
- AES gồm ba mật mã khối AES-128, AES-192, AES-256 tương ứng với độ dài của key là 128 bit, 192 bit và 256 bit. Số vòng của key khác nhau, cụ thể 10 vòng cho 128 bit, 12 vòng cho 192 bit và 14 vòng cho 256 bit.
- Mỗi vòng đều thực hiện ba bước thay thế, biến đổi và hòa trộn khối plain text (văn bản thuần túy) đầu vào để biến nó thành Ciphertext (văn bản đã mã hóa).

## 2.3. Mật mã 3AES (Advanced Encryption Standard )



**AES có an toàn không?**

- ✓ AES nếu được triển khai đúng quy trình thì sẽ đảm bảo an toàn tuyệt đối.
- ✓ Thế nhưng một điều cần lưu ý đó là bất kỳ một hệ thống nào cũng có thể bị tấn công nếu hacker biết được key mã hóa.
- ✓ Do đó các key mã hóa AES phải được bảo vệ bằng nhiều cách khác nhau như dùng **mật khẩu** mạnh, xác thực, tường lửa hay phần mềm chống độc hại.

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Giới thiệu chung

- Vào năm 1999, cục tiêu chuẩn quốc gia Hoa Kỳ (NIST) đã ban hành một phiên bản mới của tiêu chuẩn DES chỉ ra rằng DES chỉ nên được sử dụng cho các hệ thống cũ và 3-DES được sử dụng.
- 3-DES có **2 ưu điểm** đảm bảo cho việc sử dụng rộng rãi trong vài năm tới:
  - Đầu tiên, với độ dài khóa 168-bit, nó khắc phục được lỗ hổng đối với cuộc tấn công vét cạn của DES.
  - Thứ hai, thuật toán mã hóa cơ bản trong 3-DES cũng giống như trong DES. → có khả năng chống thám mã tốt.
- 3-DES có **2 nhược điểm:**
  - Hạn chế chính của 3-DES là thuật toán tương đối chậm trong phần mềm. DES ban đầu được thiết kế để triển khai bằng phần cứng giữa những năm 1970 và không tạo ra mã phần mềm hiệu quả.
  - Một nhược điểm phụ là cả DES và 3-DES đều sử dụng kích thước khối 64-bit. Vì lý do cả hiệu quả và bảo mật, kích thước khối lớn hơn là cần thiết.

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

- Để thay thế, vào năm 1997 NIST đã đưa ra lời kêu gọi đề xuất **Tiêu chuẩn mã hóa nâng cao (AES)** mới, tiêu chuẩn này phải có sức mạnh bảo mật bằng hoặc tốt hơn 3-DES và cải thiện đáng kể hiệu quả.
- NIST quy định rằng AES phải là mật mã khối đối xứng với độ dài khối 128 bit và hỗ trợ độ dài khóa có thể là 128, 192 và 256 bit
- NIST đã chọn Rijndael làm thuật toán AES được đề xuất. Hai nhà nghiên cứu đã phát triển và gửi Rijndael cho AES đều là những nhà mật mã học đến từ Bỉ: Tiến sĩ Joan Daemen và Tiến sĩ Vincent Rijmen.
- Cuối cùng, AES được thiết kế để thay thế 3-DES, nhưng quá trình này sẽ mất một số năm. NIST dự đoán rằng DES ba lần vẫn sẽ là một thuật toán được sử dụng trong tương lai gần.

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

- Bảng dưới liệt kê tham số của AES tùy thuộc vào kích thước của khóa. Trong phần này ta lựa chọn khóa 128 bits là kích thước thông dụng thường được triển khai trong thực tế

Kích thước khóa (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Kích thước khối của bản rõ (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Số vòng	10	12	14
Kích thước khóa tại mỗi vòng (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Kích thước khóa mở rộng (words/bytes)	44/176	52/208	60/240

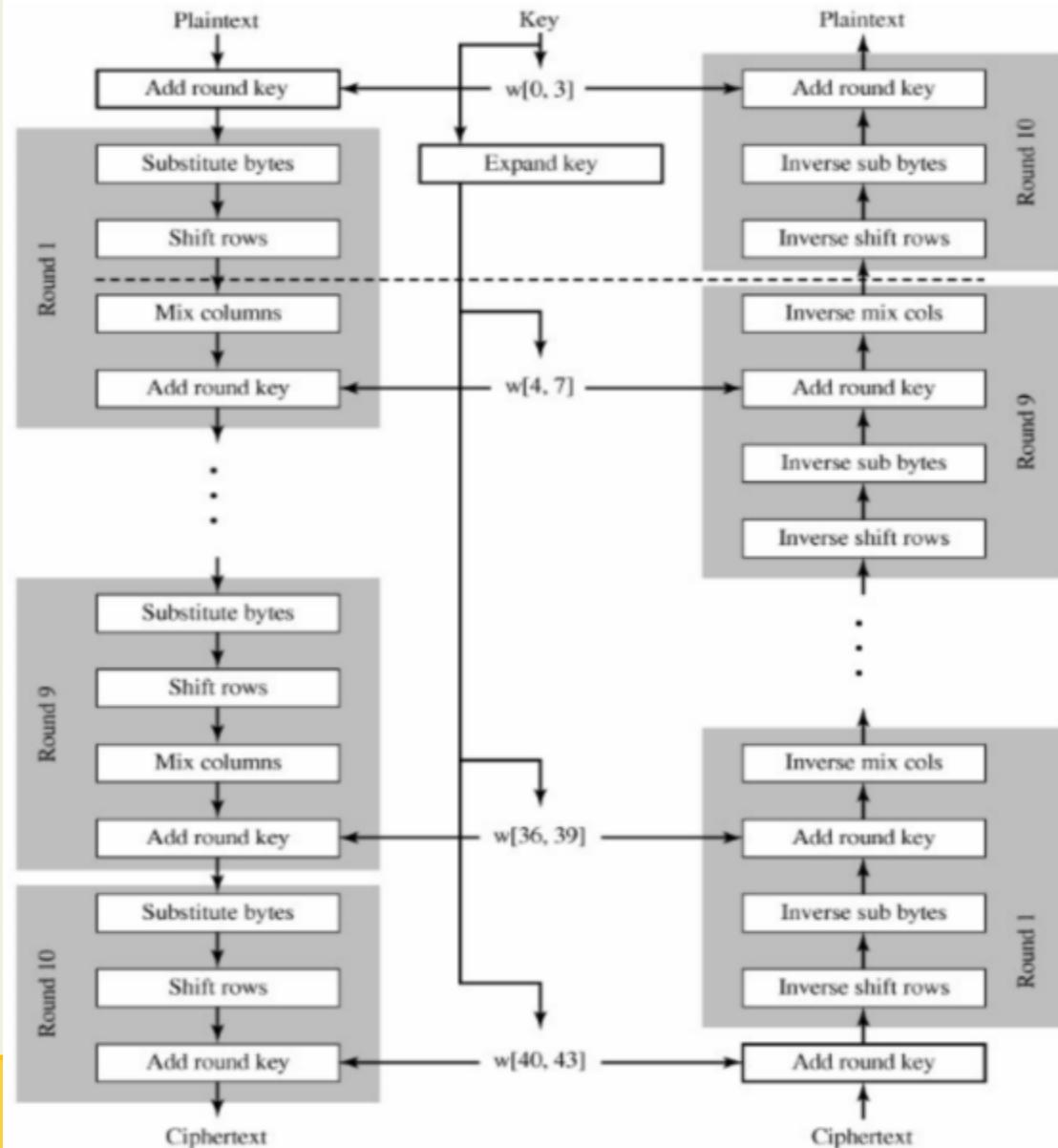
# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Thuật toán Mã hóa AES:

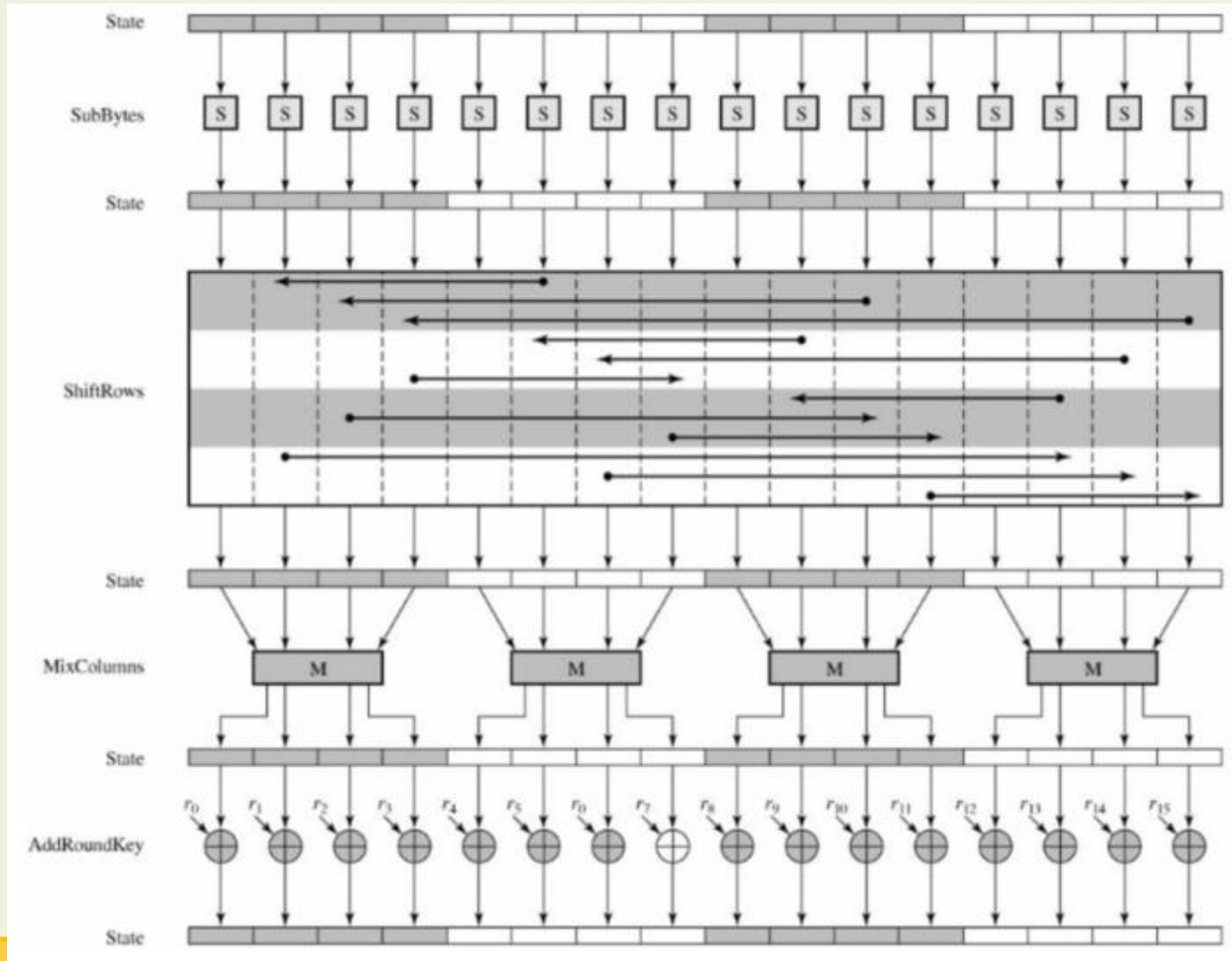
- Đầu vào cho thuật toán mã hóa và giải mã là một khối 128 bít, khối bít này được mô tả là một ma trận vuông, mỗi ô là 1 byte;
- Khối này được sao chép vào một mảng trạng thái, được sửa đổi ở mỗi giai đoạn mã hóa hoặc giải mã;
- Sau giai đoạn cuối cùng, mảng trạng thái này được sao chép vào một ma trận đầu ra.
- Tương tự, khóa 128 bit được mô tả như một ma trận vuông, mỗi phần tử là một byte;
- Khóa này sau đó được mở rộng thành một mảng các từ (word), mỗi từ là bốn byte và tổng chiều dài khóa là 44 từ cho khóa 128 bit
- Lưu ý rằng thứ tự của các byte trong ma trận là theo cột.
- Vì vậy, bốn byte đầu tiên của bản rõ 128 bit đầu vào chiếm cột đầu tiên của ma trận, bốn byte thứ hai chiếm cột thứ hai, v.v. Tương tự, bốn byte đầu tiên của khóa mở rộng, tạo thành một từ, chiếm cột đầu tiên của ma trận w.

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

*Cấu trúc mã hóa và giải mã AES*



# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)



*Một vòng  
mã hóa  
đầy đủ  
AES*

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Thuật toán Mã hóa AES:

Cả thuật toán mã hóa và giải mã đều bắt đầu giai đoạn AddRoundKey, tiếp theo là **9 vòng**, mỗi vòng đầy đủ **4 giai đoạn**:

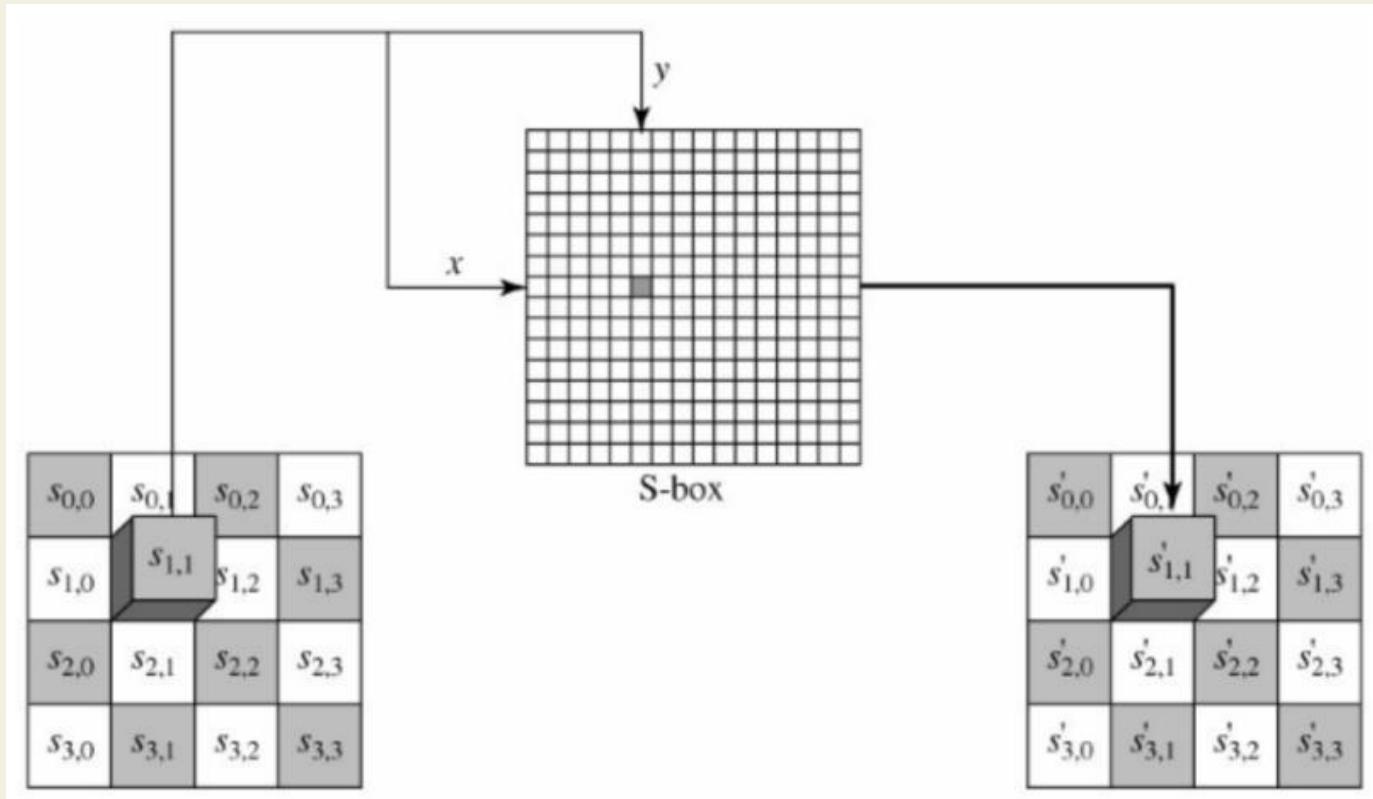
- ❖ Thay thế các bytes (*Substitute bytes*) sử dụng hộp S để thực hiện việc thay thế từng byte của khối;
- ❖ Dịch các dòng (*ShiftRows*) đơn giản là thực hiện hoán vị;
- ❖ Trộn cột (*MixColumns*) là phép thay thế sử dụng các phép toán số học trên  $Z_{256}$ ;
- ❖ *AddRoundKey* đơn giản chỉ là phép XOR của khối hiện tại với một phần của khóa được mở rộng.

Vòng cuối cùng chỉ có 3 giai đoạn.

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Thuật toán Mã hóa AES:

- **Hàm SubBytes:** Thay thế byte đơn giản chỉ là tra cứu trong bảng  $16 \times 16$ , mỗi ô là 1 byte và được gọi là hộp S-box và S-box đảo.



*Phép thay thế byte sử dụng hộp S*

TS. Lê Thị Anh

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Thuật toán Mã hóa AES:

- Hàm SubBytes: S-box

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Thuật toán Mã hóa AES:

- **Hàm SubBytes:** *Hộp S đảo (inverse S box)*

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Thuật toán Mã hóa AES:

- **Hàm SubBytes:** Ví dụ minh họa phép thay thế byte



The diagram illustrates the SubBytes function mapping. On the left, there is a 4x4 matrix of bytes:

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

An arrow points from this matrix to another 4x4 matrix on the right, representing the output of the SubBytes function:

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

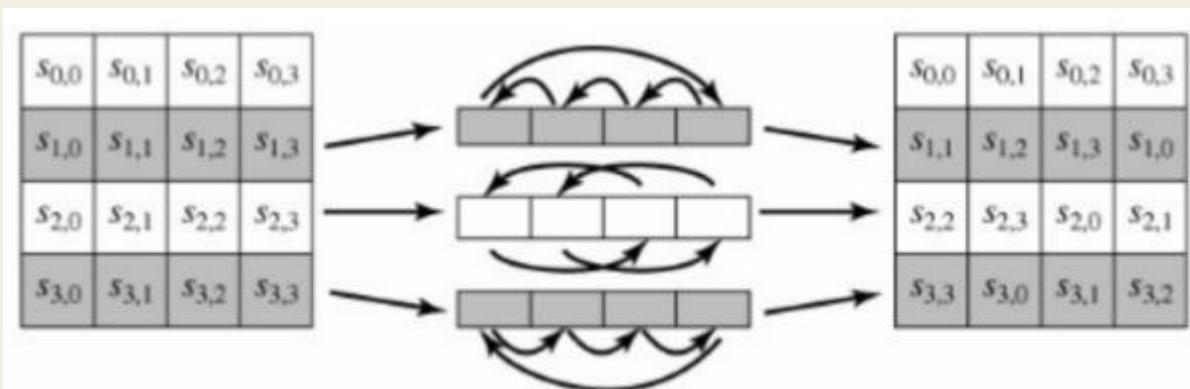
*Tìm byte thay thế của EA: Tra dòng E và cột A trong S-box được 87. Vậy thay thế EA bằng 87*

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Thuật toán Mã hóa AES:

**Dịch dòng (Shiftrows)** : Dòng đầu tiên của ma trận trạng thái được giữ nguyên, dòng thứ hai quay trái 1 byte, dòng thứ 3 quay trái 2 byte và dòng cuối cùng quay trái 3 byte.

*Minh họa  
phép dịch  
dòng*



*Ví dụ minh họa  
phép dịch dòng*

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6



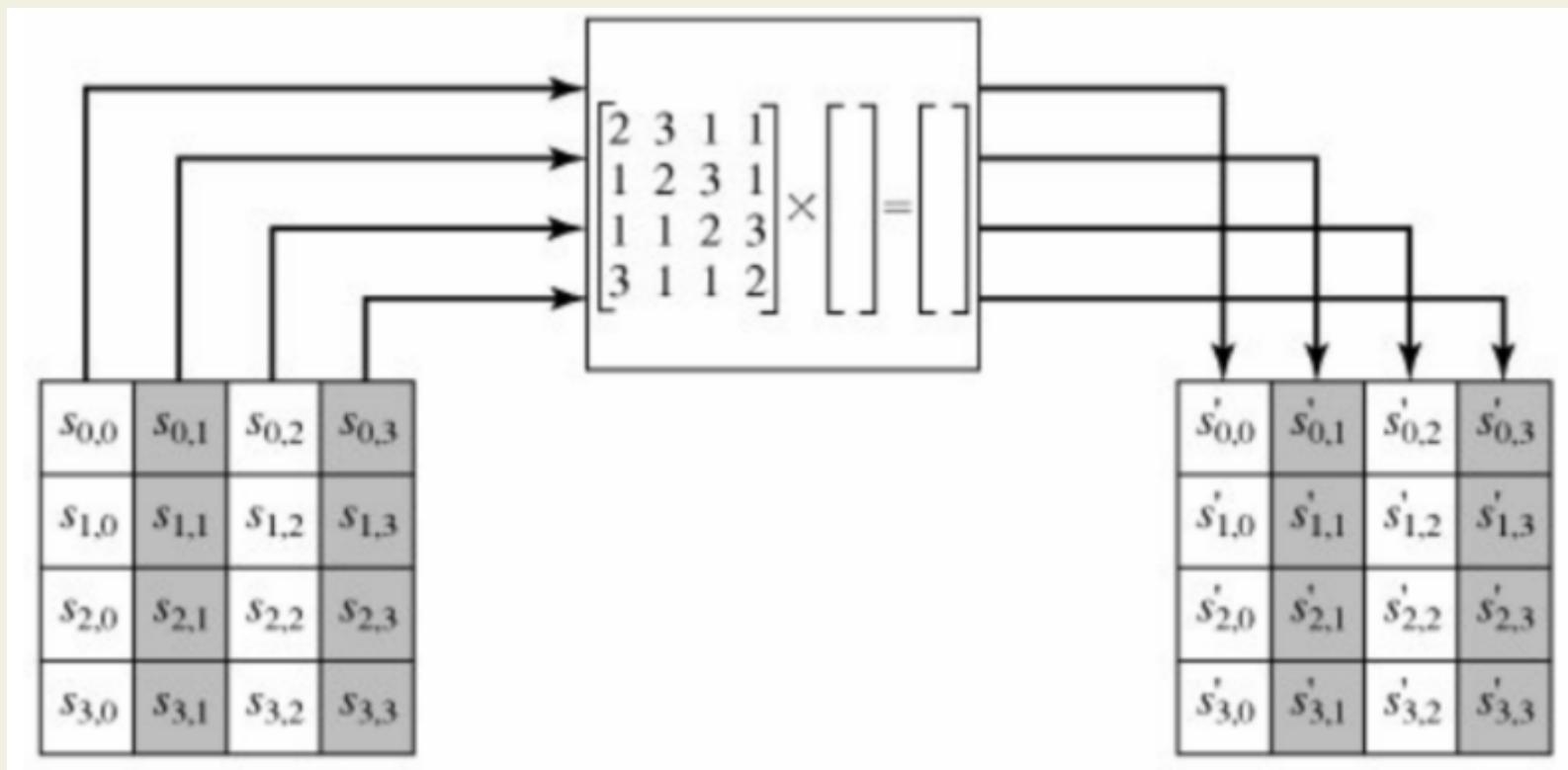
87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

**Đối với thuật toán giải mã ta sử dụng phép dịch dòng ngược.**

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

Thuật toán Mã hóa AES:

**Trộn cột:** Phép trộn cột được thực hiện như minh họa



# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Thuật toán Mã hóa AES:

**Trộn cột:** Kết quả phép trộn cột được tính như sau:

$$\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

Áp dụng phép nhân hai ma trận ta có?

$$s'_{0,j} = (2 \cdot s_{0,j}) + (3 \cdot s_{1,j}) + s_{2,j} + s_{3,j}$$

$$s'_{1,j} = s_{0,j} + (2 \cdot s_{1,j}) + (3 \cdot s_{2,j}) + s_{3,j}$$

$$s'_{2,j} = s_{0,j} + s_{1,j} + (2 \cdot s_{2,j}) + (3 \cdot s_{3,j})$$

$$s'_{3,j} = (3 \cdot s_{0,j}) + s_{1,j} + s_{2,j} + (2 \cdot s_{3,j})$$

Trong đó, phép nhân(.) được thực hiện theo luật sau: Giả sử  $s_{i,j}$  được biểu diễn dưới dạng 8 bit  $b_7b_6b_5b_4b_3b_2b_1b_0$  khi nhân với 2 sẽ được thực hiện theo công thức sau:

$$2 \cdot s_{i,j} = \begin{cases} b_7b_6b_5b_4b_3b_2b_1b_0 & \text{nếu } b_7 = 0 \\ b_7b_6b_5b_4b_3b_2b_1b_0 + 00011011 & \text{nếu } b_7 = 1 \end{cases}$$

$$3 \cdot s_{i,j} = s_{i,j} + 2 \cdot s_{i,j}$$

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

**Thuật toán Mã hóa AES:**

**Trộn cột:**

**Ví dụ minh họa**

87	F2	4D	97	→	47	40	A3	4C
6E	4C	90	EC		37	D4	70	9F
46	E7	4A	C3		94	E4	3A	42
A6	8C	D8	95		ED	A5	A6	BC

Ta diễn giải cách xác định phần tử đầu tiên trong ma trận sau khi thực hiện phép trộn cột.

$$s'_{0,0} = 2 \cdot (87) + 3 \cdot (6E) + 46 + A6$$

Chuyển các số từ hệ 16 sang hệ 2 thu được  $87h = 10000111$ .

Do bít  $b_7 = 1$  nên

$$2.(87) = 00001110 \text{ XOR } 00011011 = 00010101, 6Eh$$

$$= 01101110, 46h = 01000110,$$

$$A6h = 10100110 \text{ và } 3.(6E) = 6E + 2.(6E).$$

$$\text{Do bít } b_7 \text{ của } 6E \text{ là } 0 \text{ nên } 2.(6E) = 11011100.$$

$$\text{Do đó, } 3.(6E) = 01101110 \text{ XOR } 11011100 = 10110010.$$

2.(87)	=	0	0	0	1	0	1	0	1
3.(6E)	=	1	0	1	1	0	0	1	0
46	=	0	1	0	0	0	1	1	0
A6	=	1	0	1	0	0	1	1	0
XOR		0	1	0	0	0	1	1	1 = 47h

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Thuật toán Giải mã AES ??

**Trộn cột:** Phép chuyển đổi đảo trộn cột (inverse mix column transform) trong thuật toán giải mã được thực hiện như sau:

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \times \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

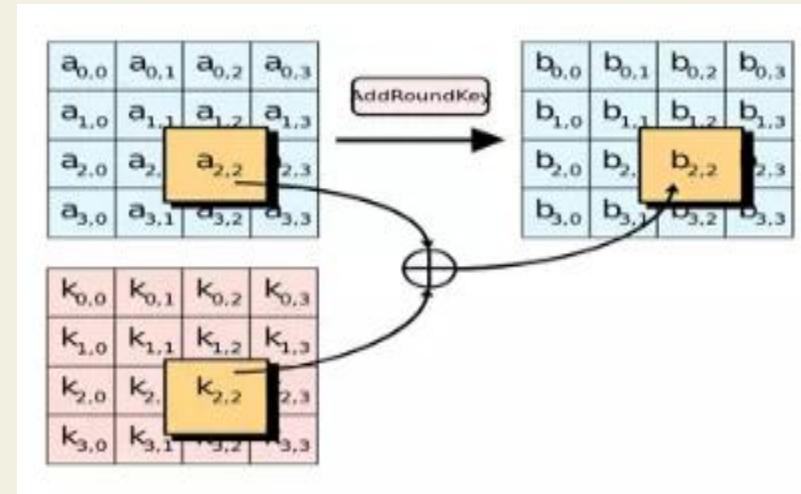
Thay thế công thức của phép trộn cột vào thì ta thu được công thức sau.

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \times \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Thuật toán Mã hóa AES:

- Hàm AddRoundKey:** phép cộng với khóa là thực hiện phép XOR bít của 128 bít của ma trận trạng thái và 128 bít của khóa tương ứng của vòng. Mỗi khóa vòng gồm có 4 từ (128 bit) được lấy từ lịch trình khóa. 4 từ đó được cộng vào mỗi cột của state, sao cho:



<table border="1"> <tr><td>47</td><td>40</td><td>A3</td><td>4C</td></tr> <tr><td>37</td><td>D4</td><td>70</td><td>9F</td></tr> <tr><td>94</td><td>E4</td><td>3A</td><td>42</td></tr> <tr><td>ED</td><td>A5</td><td>A6</td><td>BC</td></tr> </table>	47	40	A3	4C	37	D4	70	9F	94	E4	3A	42	ED	A5	A6	BC	$\oplus$	<table border="1"> <tr><td>AC</td><td>19</td><td>28</td><td>57</td></tr> <tr><td>77</td><td>FA</td><td>D1</td><td>5C</td></tr> <tr><td>66</td><td>DC</td><td>29</td><td>00</td></tr> <tr><td>F3</td><td>21</td><td>41</td><td>6A</td></tr> </table>	AC	19	28	57	77	FA	D1	5C	66	DC	29	00	F3	21	41	6A	=	<table border="1"> <tr><td>EB</td><td>59</td><td>8B</td><td>1B</td></tr> <tr><td>40</td><td>2E</td><td>A1</td><td>C3</td></tr> <tr><td>F2</td><td>38</td><td>13</td><td>42</td></tr> <tr><td>1E</td><td>84</td><td>E7</td><td>D2</td></tr> </table>	EB	59	8B	1B	40	2E	A1	C3	F2	38	13	42	1E	84	E7	D2
47	40	A3	4C																																																	
37	D4	70	9F																																																	
94	E4	3A	42																																																	
ED	A5	A6	BC																																																	
AC	19	28	57																																																	
77	FA	D1	5C																																																	
66	DC	29	00																																																	
F3	21	41	6A																																																	
EB	59	8B	1B																																																	
40	2E	A1	C3																																																	
F2	38	13	42																																																	
1E	84	E7	D2																																																	

Ví dụ minh họa: Phép cộng khóa

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

- *Mở rộng khóa*

Thuật toán mở rộng khóa có đầu vào là 4 từ (16 bytes) khóa và tạo ra một mảng đầu ra 44 từ (176 bytes). Mã giả của thuật toán được mô tả như sau:

```
KeyExpansion (byte key[16], word w[44])  
{  
    word temp  
    for(i=0; i<4; i++)  
        w[i] = (key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])  
    for(i=4, i<44; i++)  
    {  
        temp = w[i-1]  
        if(i mod 4 = 0)  
            temp = SubWord(RotWord(temp)) XOR Rcon[i/4]  
        w[i] = w[i-4] XOR temp  
    }  
}
```

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

- *Mở rộng khóa*

Trong đó, phép toán **RotWord** là thực hiện phép quay trái 1 byte, tức là đầu vào 1 từ có 4 byte [b0, b1, b2, b3] thì kết quả sau khi thực hiện phép quay trái 1 byte sẽ là [b1, b2, b3, b0].

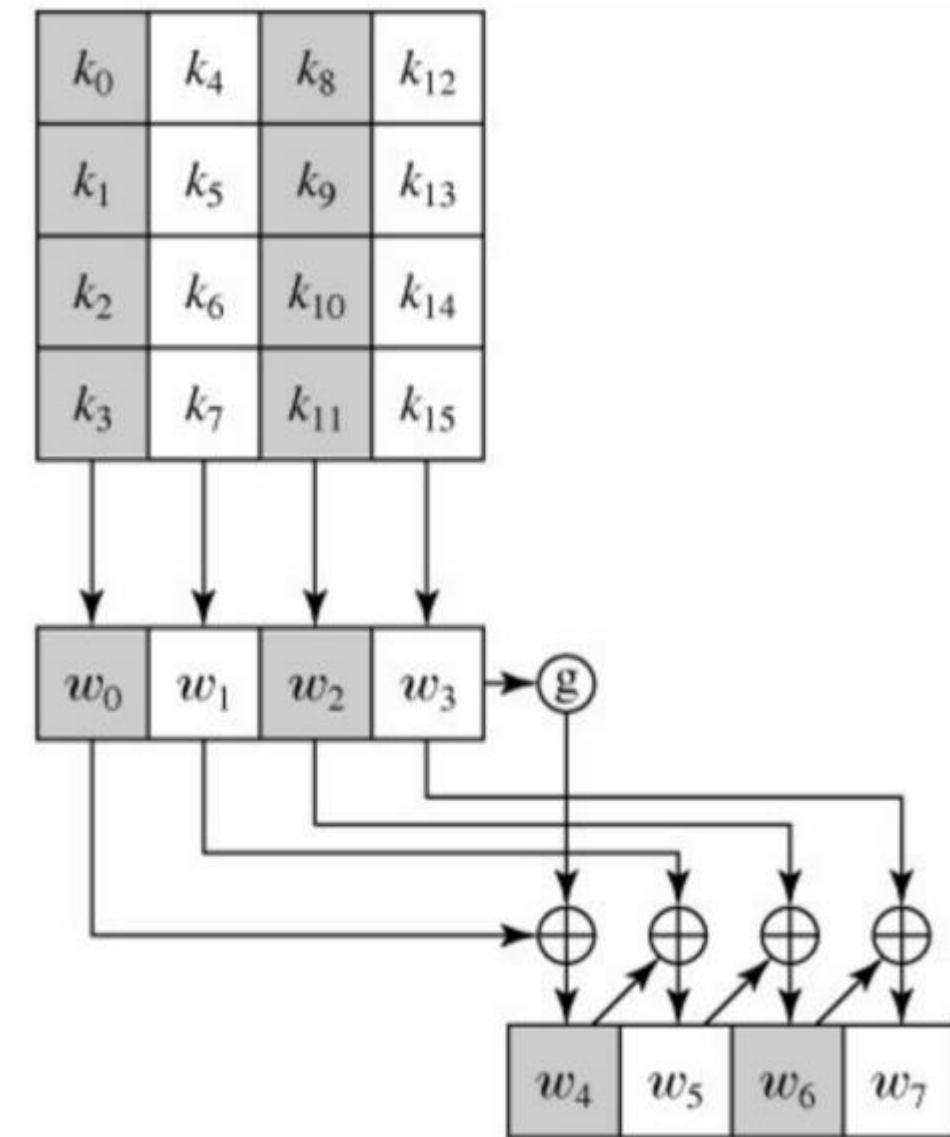
Phép toán **SubWord** là phép thay thế byte sử dụng bảng S. Hàng số cho mỗi vòng khóa  $Rcon[j] = (RC[j], 0, 0, 0)$ , với  $RC[1] = 1$ ,  $RC[j] = 2 \cdot RC[j-1]$  và phép nhân(.) được thực hiện theo luật như trong thuật toán trộn cột.

Giá trị của  $RC[j]$  được xác định như bảng dưới ở hệ thập lục phân (hexadecimal).

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

*Minh họa cách xác định  
khóa của vòng 1*



# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

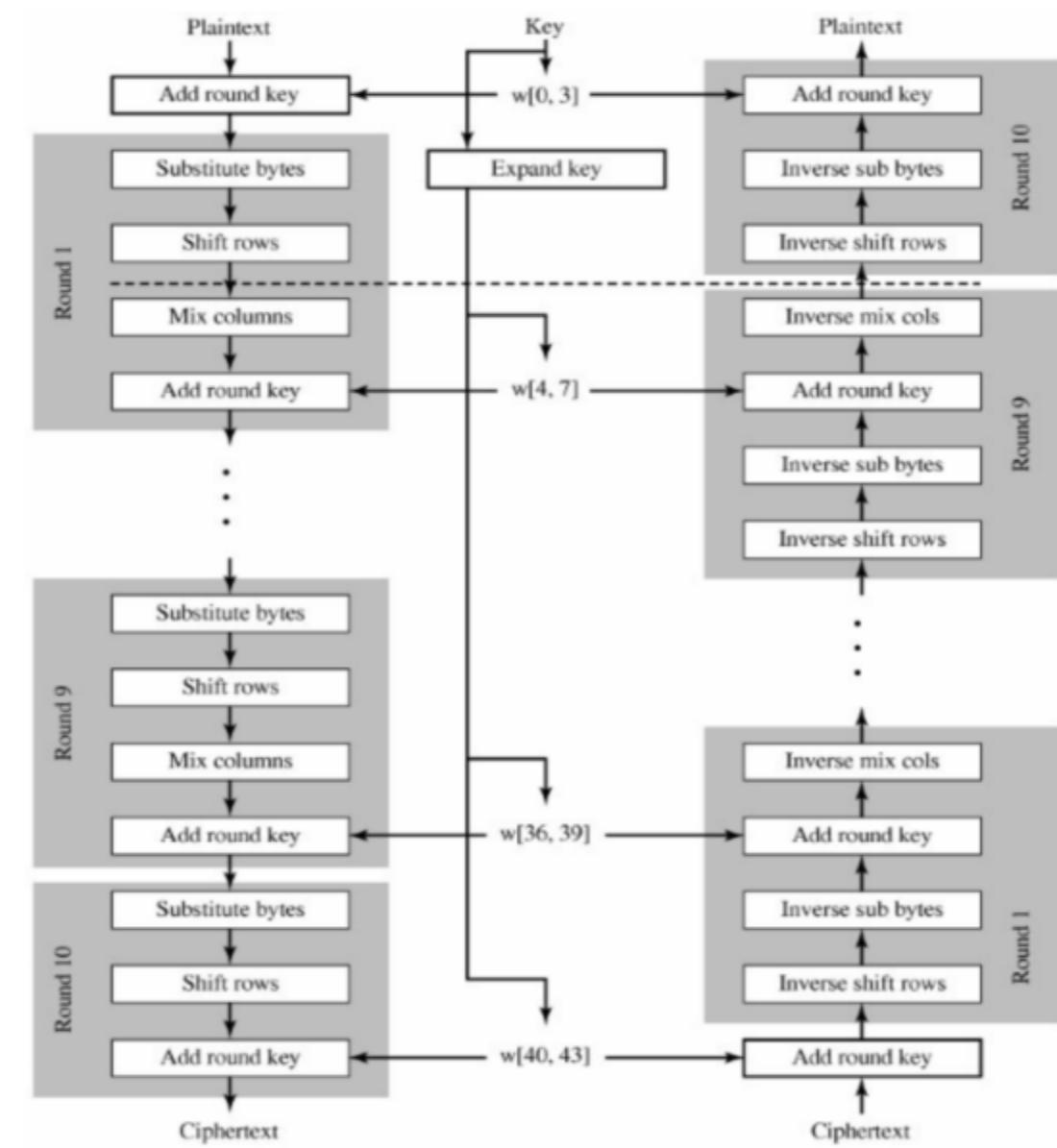
- Ví dụ minh họa cách xác định khóa cho vòng thứ 9 khi khóa tại vòng 8 là EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F tương ứng  $w[32] = [EA, D2, 73, 21]$ ,  $w[33] = [B5, 8D, BA, D2]$ ,  $w[34] = [31, 2B, F5, 60]$  và  $w[35] = [7F, 8D, 29, 2F]$ . Giá trị của khóa tại vòng 9 được xác định như bảng sau:

*Ví dụ xác định khóa tại vòng 8*

Giá trị i ở hệ thập phân	temp	Sau khi thực hiện phép RotWord	Sau khi thực hiện phép SubWord	Rcon(9)	Sau khi XOR với Rcon	w[i-4]	w[i] = temp XOR w[i-4]
36	7F8D292F	8D292F7F	5DA515D2	1B000000	46A515D2	EAD27321	AC7766F3
37	AC7766F3	AC7766F3	AC7766F3	1B000000	AC7766F3	B58DBAD2	19FABC21
38	19FABC21	19FABC21	19FABC21	1B000000	19FABC21	312BF560	28B14941
39	28B14941	28B14941	28B14941	1B000000	28B14941	7F8D292F	575C606E

Như vậy, khóa của vòng 9 sẽ là AC 77 66 F3 19 FA BC 21 28 B1 49 41 57 5C 60 6E.

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)



- Thuật toán giải mã: ngược lại với mã hóa

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Thuật toán giải mã:

- Thuật toán giải mã khá giống với thuật toán mã hóa về mặt cấu trúc nhưng 4 hàm sử dụng là 4 hàm ngược của quá trình mã hóa.

Mã Hóa	Giải Mã
<b>AddRoundKey()</b>	<b>InvAddRoundKey()</b>
<b>SubBytes()</b>	<b>InvSubBytes()</b>
<b>ShiftRows()</b>	<b>InvShiftRows()</b>
<b>MixColumns()</b>	<b>InvMixColumns()</b>

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## Tấn công AES và phương pháp phòng chống

✓ ***Side-channel attack***: Tấn công kênh phụ (định nghĩa là các kênh đầu ra không mong muốn từ một hệ thống)

Tấn công kênh bên hay còn gọi là Tấn công kênh phụ là loại tấn công dễ thực hiện trong các loại tấn công mạnh chống lại quá trình triển khai mã hóa, và mục tiêu của loại tấn công này là phân tích các nguyên tố, các giao thức, modul, và các thiết bị trong mỗi hệ thống.

✓ ***Known attacks***: Vào năm 2002, Nicolas Courtois và Josef Pieprzyk phát hiện một tấn công trên lý thuyết gọi là tấn công XSL và chỉ ra điểm yếu tiềm tàng của AES.

Tuy nhiên, một vài chuyên gia về mật mã học khác cũng chỉ ra một số vấn đề trong cơ sở toán học của tấn công này và cho rằng các tác giả đã có sai lầm trong tính toán. Việc tấn công dạng này có thực sự trở thành hiện thực hay không vẫn còn để ngỏ và cho tới nay thì tấn công XSL vẫn chỉ là suy đoán.

## Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

### Phương pháp phòng chống tấn công AES:

- ✓ Sử dụng mã hóa mạnh: Sử dụng các biện pháp để tăng tính bảo mật của thuật toán mã hóa
- ✓ Bảo vệ dữ liệu theo phương pháp vật lý: chống lại tấn công side-channel attack
- ✓ Kết hợp cả hai phương pháp trên

# Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

## KẾT LUẬN:

- **AES có an toàn không?** AES nếu được triển khai đúng quy trình thì sẽ đảm bảo an toàn tuyệt đối
- **Advanced Encryption Standard (AES)** là người bạn đồng hành không thể thiếu của chính phủ, cơ quan Nhà nước và tổ chức tư nhân.
- An toàn Thiết kế và độ dài khóa của thuật toán AES (128, 192 và 256 bit) là đủ an toàn để bảo vệ các thông tin TỐI MẬT. Các thông tin TUYỆT MẬT phải dùng khóa 192 hoặc 256 bit.

# BÀI TẬP ÔN TẬP

- Bài 1:** Tìm kết quả phép thay thế byte của thuật toán AES cho ma trận trạng thái đầu vào sau:

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

- Bài 2:** Tìm kết quả phép trộn cột của thuật toán AES cho ma trận trạng thái đầu vào sau:

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

# MÃ HÓA CÔNG KHAI

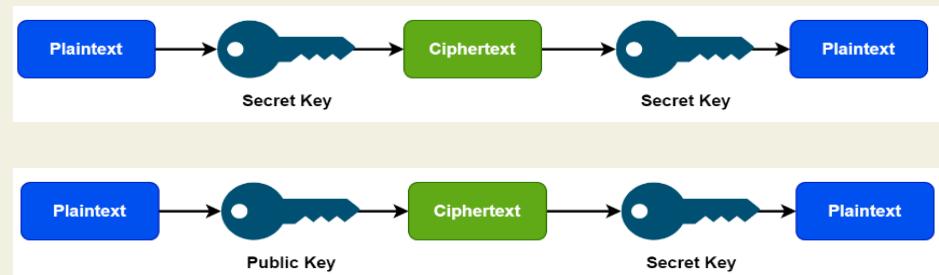
### III. Mã hóa công khai

#### - Vì sao lại ra đời mã hóa công khai?

- + Mã hóa bí mật – 1 khóa bí mật sử dụng cho cả mã hóa và giải mã
- + Khóa bí mật phải được chia sẻ trên kênh bí mật
- + Đảm bảo an toàn cho kênh bí mật có đơn giản?

→ Không cần trao đổi khóa

→ Mã hóa công khai ra đời



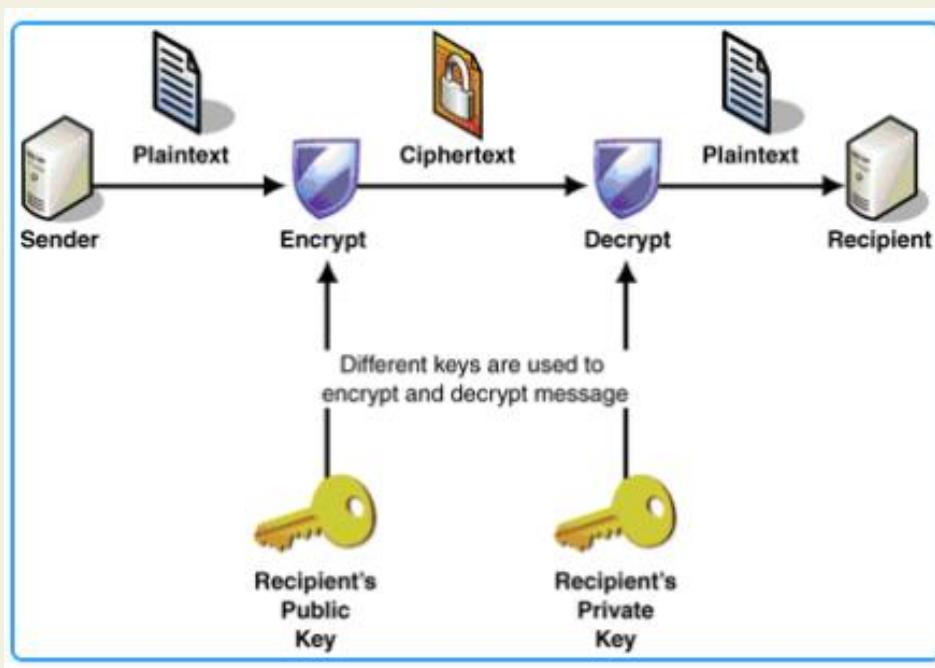
### III. Mã hóa công khai

- **Sự ra đời:**

- Ý tưởng về mã hóa công khai được Diffie và Hellman đưa ra năm 1976
- Tuy nhiên, việc thực hiện hệ mật công khai thì do Rivest, Shamir và Adleman đưa ra đầu tiên năm 1977 → Mã hóa công khai RSA
- Sau đó là hệ mật ElGamal, hay dựa trên đường cong Elliptics ra đời
- Đặc điểm chung của các hệ mật này là xuất phát từ **toán học**

### III. Mã hóa công khai

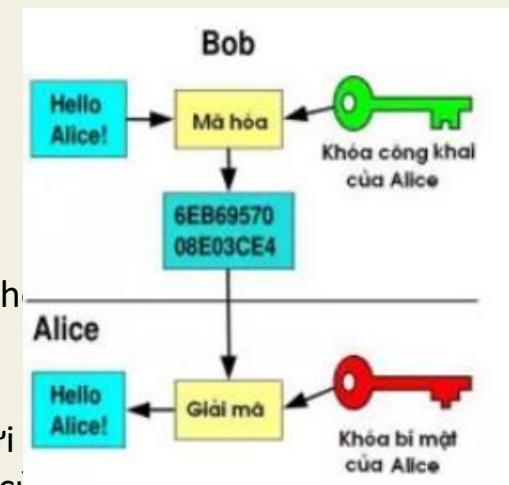
## Sơ đồ tổng quát



### III. Mã hóa công khai

#### Ý tưởng của mã hóa công khai

- ✓ Bob và Alice muốn gửi tin nhắn cho nhau
  - ✓ Alice sẽ tạo ra 2 khóa: 01 khóa công khai, 01 khóa bí mật
  - ✓ Trong đó:
    - Khóa công khai của Alice tất cả mọi người đều biết
    - Khóa bí mật chỉ 1 mình Alice biết
    - Khóa công khai và khóa bí mật liên hệ với nhau qua cơ chế toán học
    - Có khóa công khai cũng không suy ra được khóa bí mật
- Không cần chia sẻ khóa, Bob sẽ biết được khóa công khai của Alice
- ✓ Bob sẽ dùng **khóa công khai** của Alice để mã hóa bản tin mà Bob muốn gửi
  - ✓ Khi nhận được bản tin đã được mã hóa từ Bob, Alice sẽ dùng khóa bí mật của ..... để giải mã bản tin
  - ✓ Tương tự với chiều gửi tin nhắn từ Alice tới Bob, Bob cũng biết khóa công khai của Alice, còn khóa bí mật của Bob thì chỉ 1 mình Bob biết



Với thuật toán mã hóa công khai chúng ta sẽ đi vào cơ chế toán học sinh cặp khóa này.

### III. Mã hóa công khai

#### Hệ mật RSA:

- Ron Rivest, Adi Shamir và Len Adleman mô tả lần đầu tiên vào năm 1977 tại Học viện Công nghệ Massachusetts (MIT)
- Là hệ mật phù hợp nhất tạo ra chữ ký số điện tử đồng thời với việc mã hóa
- Đánh dấu sự tiến bộ vượt bậc của khoa học mật mã
- RSA được sử dụng phổ biến trong thương mại điện tử

### III. Mã hóa công khai

#### Hệ mật RSA

Thuật toán sinh khóa trong RSA

Vấn đề cốt lõi của sinh khóa trong RSA là tìm được bộ 3 số tự nhiên e, d, và n sao cho:

$$m^{ed} \equiv m \pmod{n}$$

Ở đây: m – là số tự nhiên được chuyển hóa từ bản rõ M

(d,n) – là khóa bí mật

(e,n) – là khóa công khai

Cần phải bảo mật d sao cho dù biết e và n hay thậm chí cả “m” cũng không thể tìm ra được “d”

### III. Mã hóa công khai

#### Hệ mật RSA

##### Bài toán RSA:

Cho một số nguyên dương:  $n=p*q$

Trong đó  $p,q$  là hai thừa số nguyên tố (khác 2)

Một số nguyên dương  $b$  sao cho:

$\text{USCLN}(b, (p-1)(q-1))=1$

Và một số nguyên  $c$

**Bài toán đặt ra:** Tìm số nguyên  $x$  sao cho

$$x^b \equiv c \pmod{n}$$

→ Giải được bài toán này là giải mã được

### III. Mã hóa công khai

## Hệ mật RSA

### Thuật toán: Sinh khóa cho hệ mật RSA

1. Sinh hai số nguyên tố lớn  $p$  và  $q$  có giá trị xấp xỉ nhau
2. Tính  $n=p \cdot q$ , và  $\phi(n) = (p-1)(q-1)$
3. Chọn một số ngẫu nhiên  $e$ ,  $1 < e < \phi(n)$   
sao cho  $\gcd(e, \phi(n)) = 1$
4. Sử dụng thuật toán Euclide mở rộng để tính số  $d$ ,  $1 < d < \phi(n)$   
Sao cho  $e \cdot d \equiv 1 \pmod{\phi(n)}$
5. Khóa công khai là  $(n, e)$ , khóa bí mật là  $(n, d)$ .  
Trong thực hành hay chọn  $e=65537$

### III. Mã hóa công khai

## Hệ mật RSA – Mã hóa và giải mã

- Với public key ( $n,e$ ) và private key ( $n,d$ )  $\rightarrow$  mã hóa phía người gửi và giải mã phía người nhận.
- Giả sử Bob gửi cho Alice bản rõ  $M$ .

Thực hiện mã hóa RSA như sau:

- Chuyển  $M$  về số tự nhiên  $m$  nằm trong khoảng  $(0,n)$  sao cho  $m, n$  là hai số nguyên tố cùng nhau

- Mã hóa  $m$  thành  $d$  như sau:

$$c \equiv m^e \pmod{n}$$

- Sau đó  $c$  sẽ được chuyển tới người nhận

Thực hiện giải mã RSA tại người nhận bằng private key ( $n,d$ ):

Kết quả:  $m = c^d \pmod{n}$

### III. Mã hóa công khai

#### Mã hóa RSA

Ví dụ 1:

$$p = 17, q = 11$$

$$\Rightarrow n = pq = 17 * 11 = 187$$

$$\Rightarrow \varphi(n) = 160$$

Chọn  $e=7$  vì  $\text{UCLN}(7, 160)=1$

Chọn  $d=?$  **Public key, Private key?**

### III. Mã hóa công khai

#### Hệ mật RSA

Ví dụ 1:

Giả sử  $m=32$

=> Mã hóa “ $m$ ” bằng RSA  
⇒ Bản mật?

$$c = 32 \wedge 5 \% 35 = 2$$

Giải mã  $c$  để thu được  $m$ ?

$$m = 2 \wedge 29 \% 35 = 32$$

### III. Mã hóa công khai

#### Hệ mật RSA

##### Ví dụ 2: mã hóa chuỗi nhị phân

Các tham số

Chọn  $p=11$  và  $q=13$

Khi đó  $n=11*13=143$

$$(p-1)(q-1)=120$$

Chọn  $e=37$  vì  $\gcd(e, 120)=1$

Sử dụng thuật toán gcd để tìm  $d$  sao cho  
 $e*d-1$  chia hết cho 120  $\rightarrow d=13$

### III. Mã hóa công khai

#### Hệ mật RSA

##### Ví dụ 2: mã hóa chuỗi nhị phân

Để mã hóa một chuỗi nhị phân gồm các bước:

“Bẻ” thành nhiều đoạn độ dài là u bít sao cho  $2^u < 143 \rightarrow u=7$   
Mỗi đoạn như vậy sẽ biểu diễn một số nằm trong khoảng 0-127

Tính bản mật Y theo công thức:  $Y = X^e \text{ mod } n$

Ví dụ X=(0000010)=2, ta có Y=?

Y=106 hay Y=(100 1010) → Bản mật gửi đi là Y

Giải mã? X=2?

# Hệ mã ElGamal

- Thuật toán ElGamal được giới thiệu năm 1984 bởi Taher Elgamal. Đây cũng là một thuật toán mã hóa bất đối xứng.
- Thuật toán mã hóa ElGamal cũng gồm 3 bước:
  - 1.Tạo khóa
  - 2.Mã hóa
  - 3.Giải mã

# 1. Tạo khóa

- Thực hiện các bước sau
  - 1. Chọn số nguyên tố lớn **p**, và cơ số **a**
  - 2. Chọn số **x**
  - 3. Tính số **y**:  $y = a^x \text{ mod } p$
- Ta được cặp khóa:
  - › **Khóa cá nhân**:  $\{p, a, x\}$
  - › **Khóa công khai**:  $\{p, a, y\}$

## 2. Mã hóa

- Thông điệp ban đầu: M
- Dùng **khóa công khai {p, a, y}** để mã hóa:
  1. Chọn số k, với  $1 \leq k \leq p-1$
  2. Tính  $K = y^k \text{ mod } p$
  3. Sau đó tính **cặp ciphertext**  $\{C_1, C_2\}$ :
    - $C_1 = a^k \text{ mod } p$
    - $C_2 = K \cdot M \text{ mod } p$
- Như vậy, M đã được mã hóa thành  $\{C_1, C_2\}$ :
$$M \rightarrow \{C_1, C_2\}$$
- k chỉ được dùng **một lần**, sau khi tính  $\{C_1, C_2\}$  sẽ bị hủy.

### 3. Giải mã

- Dùng **khóa cá nhân  $\{p, a, x\}$**  để giải mã  $\{C_1, C_2\}$ :
  1. Tính  $K = C_1^x \bmod p$ 
    - (vì  $C_1^x \bmod p = a^{k \cdot x} \bmod p = y^k \bmod p = K$ )
  2. Tính  $K^{-1} \bmod p$
  3. Tính  $M = C_2 \cdot K^{-1} \bmod p$

# Ví Dụ Mã Hóa ElGamal

## ▪ Tạo khóa:

1. Chọn  $p = 97$ ,  $a = 5$ ,
2. Chọn  $x = 58$ ,
3. Tính:  $y = 5^{58} = 44 \text{ mod } 97$

## ➔ Được cặp khóa:

- › Khóa bí mật:  $\{97, 5, 58\}$
- › Khóa công khai:  $\{97, 5, 44\}$

# Ví Dụ Mã Hóa ElGamal (tt)

- Mã hóa: Thông điệp  $M = 3$ :
  - Chọn  $k = 36$ ,
  - Tính  $K = 44^{36} = 75 \text{ mod } 97$
  - Tính cặp ciphertext  $\{C_1, C_2\}$ :
    - $C_1 = 5^{36} = 50 \text{ mod } 97$ ,
    - $C_2 = 75 \cdot 3 = 31 \text{ mod } 97$
- Giải mã:  $\{50, 31\}$ :
  - Tính  $K = 50^{58} = 75 \text{ mod } 97$ ,
  - Tính  $K^{-1} = 22 \text{ mod } 97$
  - Tính thông điệp ban đầu:  $M = C_2 \cdot K^{-1} \text{ mod } p = 31 \cdot 22 \text{ mod } 97 = 3 \text{ mod } 97$

### III. Mã hóa công khai

## Trao đổi khóa Diffie-Hellman

- ✓ Trao đổi khóa Diffie-Hellman là một trong những phát triển quan trọng nhất trong mật mã hóa công khai.
- ✓ Cho phép hai bên trước đây chưa gặp nhau thiết lập một cách an toàn một khóa mà họ có thể sử dụng để bảo mật thông tin liên lạc của họ
- ✓ Trao đổi khóa Diffie-Hellman là thường xuyên được thực hiện trong các giao thức bảo mật như TLS, IPsec, SSH, PGP và nhiều giao thức khác

### III. Mã hóa công khai

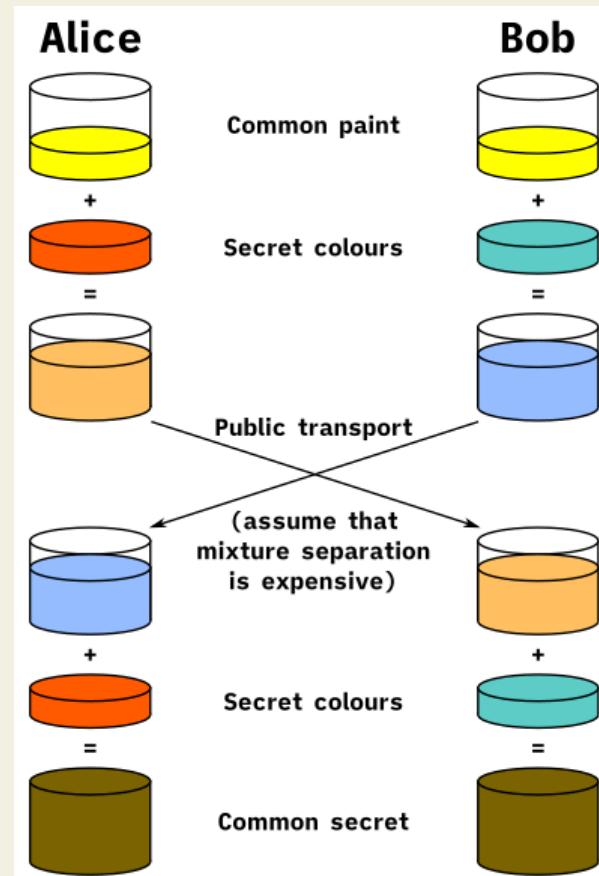
## Trao đổi khóa Diffie-Hellman

**Ý tưởng: Alice và Bob trao đổi màu sơn bí mật thông qua hỗn hợp sơn.**

• Đầu tiên Alice và Bob trộn màu đã biết chung (màu vàng) với màu bí mật riêng của mỗi người.

• Sau đó, mỗi người chuyển hỗn hợp của mình tới người kia thông qua một kênh vận chuyển công cộng.

• Khi nhận được hỗn hợp của người kia, mỗi người sẽ trộn thêm với màu bí mật của riêng mình và nhận được hỗn hợp cuối cùng.



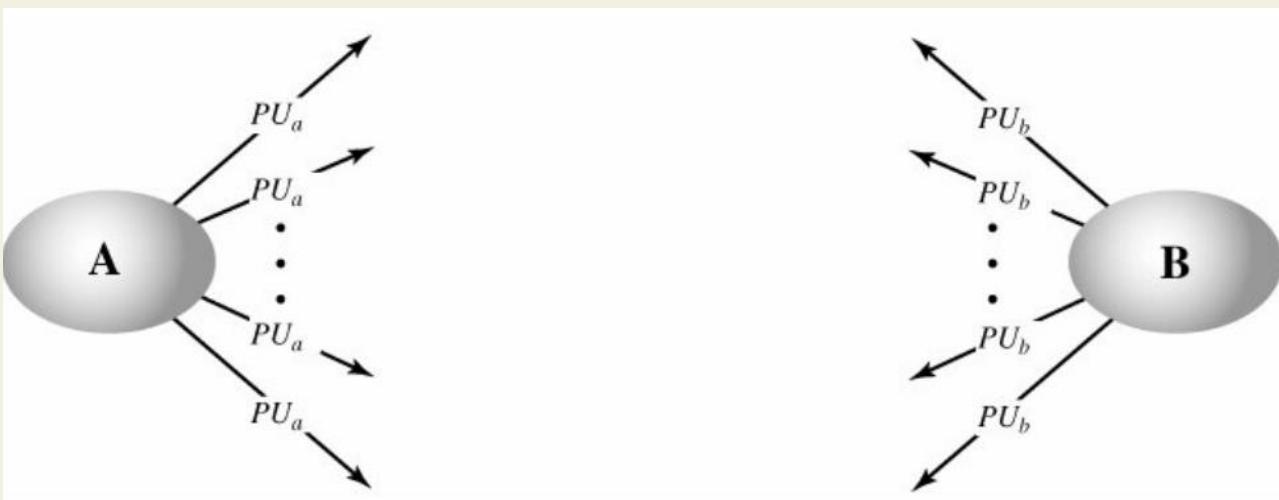
# Quản lý khóa

- **Có hai khía cạnh khác biệt đối với việc sử dụng mật mã khóa công khai về vấn đề này:**
  - Phân phối khóa công khai
  - Sử dụng hệ thống khóa mã hóa khóa công khai để phân phối khóa bí mật.
- **Phân phối khóa công khai:** một số kỹ thuật được sử dụng để phân phối khóa công khai
  - ✓ Thông báo công khai
  - ✓ Thẩm quyền khóa công khai
  - ✓ Chứng thực khóa công khai
- **Phân phối khóa bí mật sử dụng hệ mật mã khóa công khai:** Mã hóa khóa công khai được dùng để **thiết lập khóa bí mật** cho mỗi phiên trao đổi dữ liệu của **mã hóa đối xứng**. Lúc này khóa bí mật được gọi là khóa phiên (session key), các phiên trao đổi dữ liệu khác nhau sẽ dùng các khóa bí mật khác nhau.

# Quản lý khóa

- **Phân phối khóa công khai: Thông báo công khai**

Khi hai người sử dụng muốn truyền dữ liệu với nhau bằng phương pháp mã hóa khóa công khai, trước tiên họ phải trao đổi khóa công khai cho nhau.



*Phân phối khóa công khai một cách tự phát*

# Quản lý khóa

- ***Phân phối khóa công khai: Thông báo công khai***

Nhược điểm:

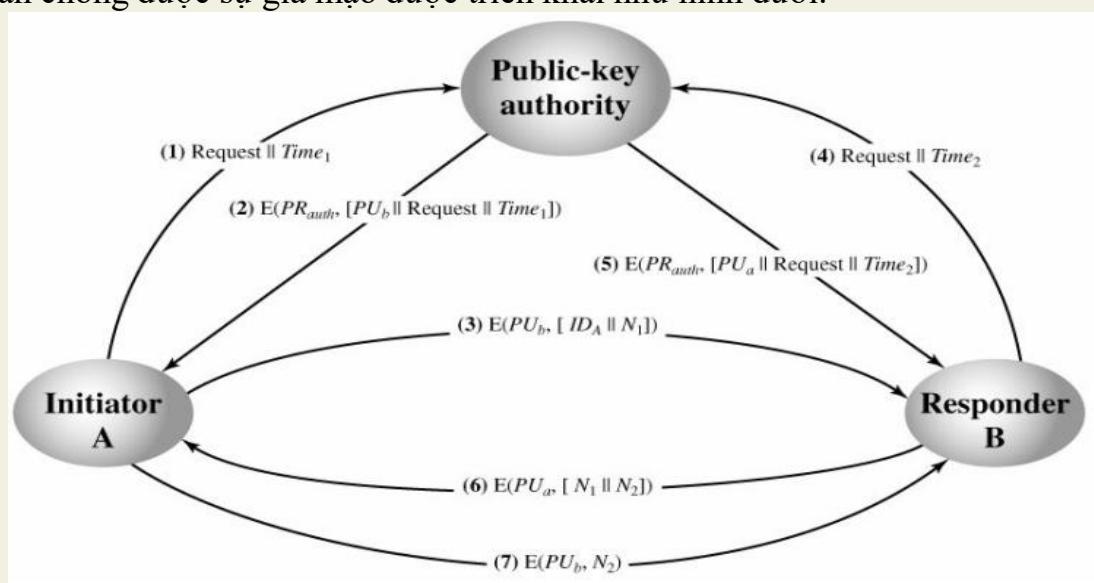
- ✓ Phương pháp trao đổi khóa này rất thuận lợi nhưng có một nhược điểm là bất kỳ ai cũng có thể giả mạo người khác để quảng bá khóa công khai của mình.
- ✓ Tức là, một người dùng bất kỳ có thể giả mạo người dùng A và gửi khóa công khai đến cho các người tham gia khác.
- ✓ Cho tới khi người dùng A phát hiện ra hành vi giả mạo và cảnh báo những người tham gia khác, kẻ giả mạo có thể đọc tất cả các tin nhắn được mã hóa dành cho A và có thể sử dụng các khóa giả mạo để xác thực.

# Quản lý khóa

## • Phân phối khóa công khai: Thẩm quyền khóa công khai

Phương pháp trao đổi khóa an toàn chống được sự giả mạo được triển khai như hình dưới.

Trung tâm thẩm quyền khóa công khai duy trì một thư mục động chứa khóa công khai của tất cả các người tham gia. Ngoài ra, mỗi người tham gia đều biết khóa công khai của trung tâm và chỉ trung tâm mới biết khóa bí mật (khóa riêng) tương ứng.



Trao đổi khóa công khai thông qua thẩm quyền khóa công khai

# Quản lý khóa

## • Phân phối khóa công khai: Thẩm quyền khóa công khai

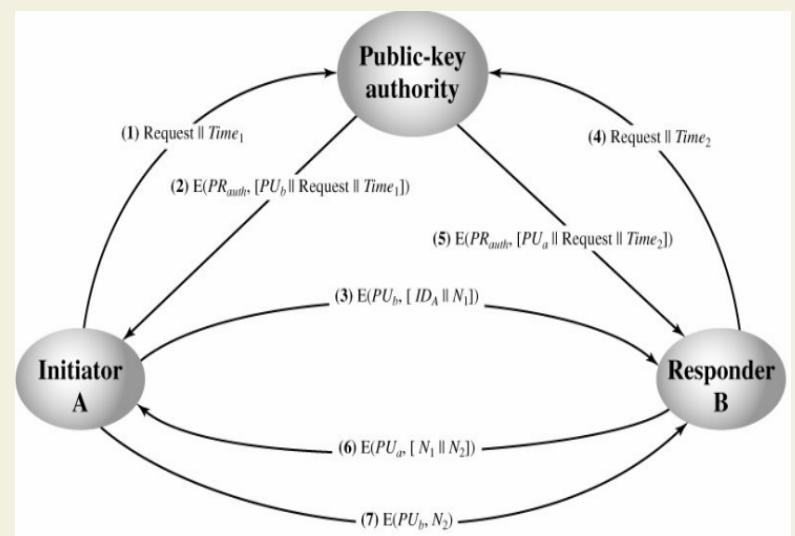
Quá trình trao đổi khóa giữa 2 người A và B thông qua trung tâm được diễn ra các bước như sau (7 bước):

**Bước 1:** Bên A gửi một thông điệp có gắn thời gian đến trung tâm thẩm quyền để yêu cầu khóa công khai hiện tại của bên B.

**Bước 2:** Trung tâm thẩm quyền trả lời lại cho A một thông điệp được mã hóa bằng khóa bí mật của mình  $PR_{auth}$ . Do đó, bên A có thể giải mã được thông điệp này sử dụng khóa công khai của trung tâm. Nội dung của thông điệp: Khóa công khai của B ( $PU_b$ ); Yêu cầu Request; Thời gian ban đầu.

**Bước 3:** A lưu trữ khóa công khai của B và sử dụng nó để mã hóa một thông điệp gửi tới B có chứa số định danh của A ( $ID_A$ ) và số ngẫu nhiên chỉ sử dụng một lần ( $N_1$ ) để xác định duy nhất giao dịch này

**Bước 4:** Bên B nhận khóa công khai của A từ trung tâm có thẩm quyền tương tự như cách mà bên A nhận khóa công khai của B.



# Quản lý khóa

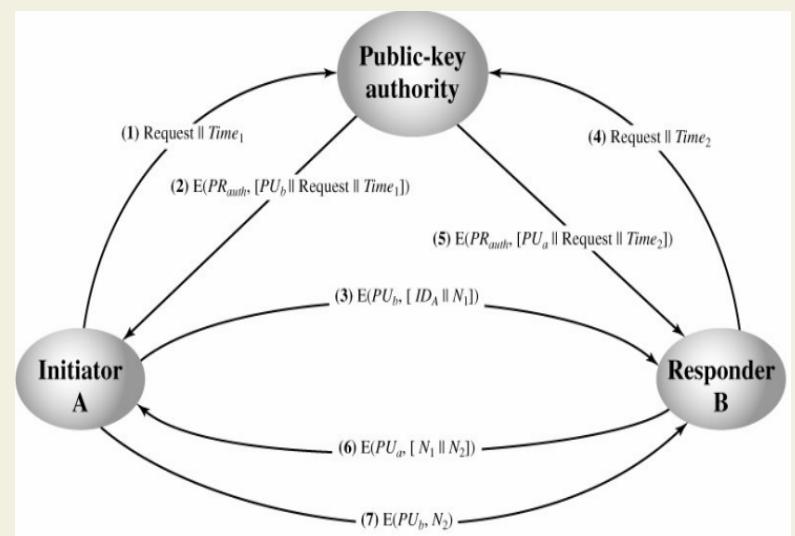
- **Phân phối khóa công khai: Thẩm quyền khóa công khai**

Quá trình trao đổi khóa giữa 2 người A và B thông qua trung tâm được diễn ra các bước như sau:

**Bước 5:** Tại thời điểm này việc phân phối khóa công khai của A và B đã được thực hiện một cách bảo mật, A, B có thể trao đổi thông tin an toàn cho nhau.

**Bước 6:** Bên B gửi một thông điệp chứa số ngẫu nhiên N1 nhận được từ A và số ngẫu nhiên N2 do B tạo đến bên A được mã hóa bằng khóa công khai của A (PUa).

**Bước 7:** A trả lại B thông điệp chứa N2 được mã hóa bằng mã công khai của B (PUB) để đảm bảo cho B rằng chính là A



# Quản lý khóa

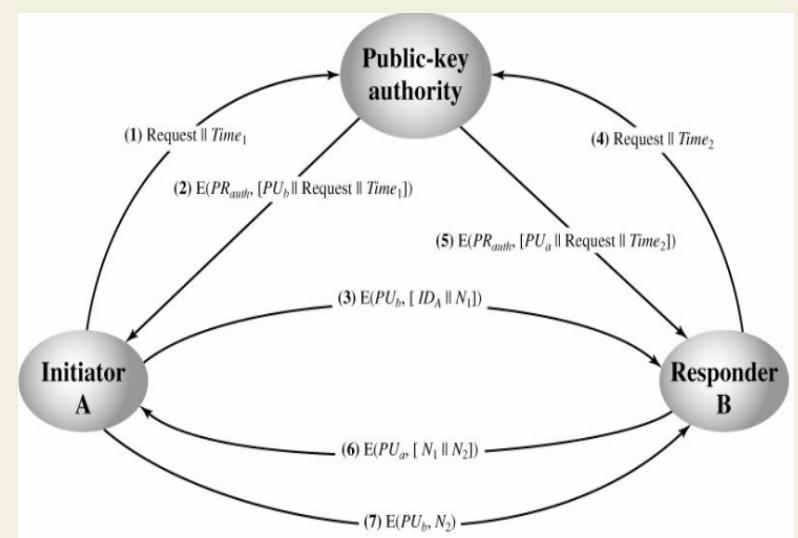
## • Phân phối khóa công khai: Chứng thực khóa công khai (public key certificates)

Chứng thực khóa công khai ghép với số nhận dạng của chủ sở hữu khóa, toàn bộ thông tin này được ký bởi bên thứ ba đáng tin cậy.

Thông thường, bên thứ ba là cơ quan cấp chứng thực, chẳng hạn như cơ quan chính phủ hoặc tổ chức tài chính, được cộng đồng người dùng tin cậy.

Người dùng có thể gửi khóa công khai của mình cho cơ quan quản lý khóa một cách an toàn và nhận lại chứng thực. Sau đó, người dùng có thể phân phối chứng thực của mình.

Bất kỳ ai cần khóa công khai của người dùng này đều có thể nhận chứng thực và xác minh rằng nó hợp lệ bằng chữ ký đáng tin cậy đính kèm.

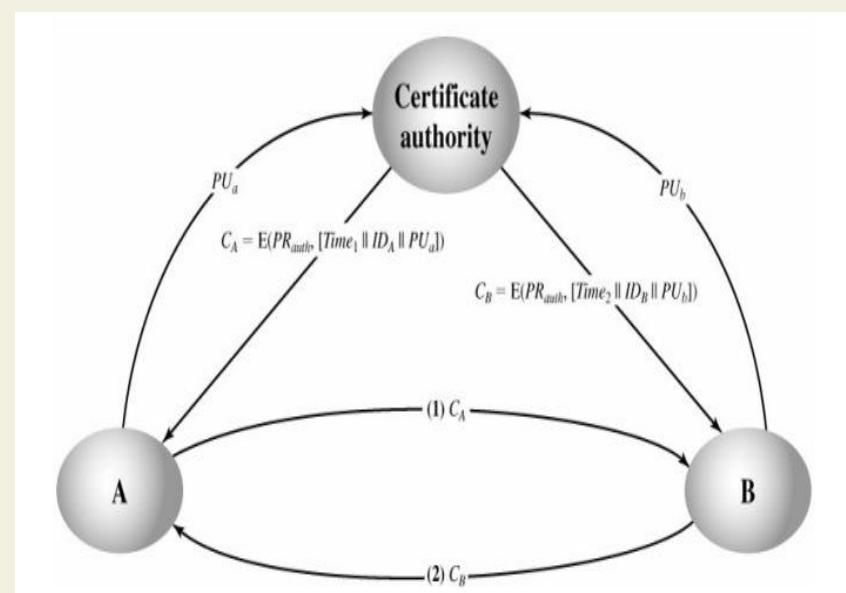


# Quản lý khóa

- **Phân phối khóa công khai: Chứng thực khóa công khai (public key certificates)**

Các bước nhận và phân phối chứng thực được minh họa trên hình bên và các yêu cầu đối với phương pháp trao đổi khóa này như sau:

- **Yêu cầu 1:** Bất kỳ người tham gia nào cũng có khả năng đọc chứng thực để xác định tên và khóa công khai của chủ sở hữu chứng thực.
- **Yêu cầu 2:** Bất kỳ người tham gia nào cũng có thể xác minh rằng chứng thực có nguồn gốc từ cơ quan cấp chứng thực và không phải là giả mạo.
- **Yêu cầu 3:** Chỉ tổ chức phát hành chứng thực mới có thể tạo và cập nhật chứng thực.
- **Yêu cầu 4:** Bất kỳ người tham gia nào cũng có thể xác minh tính hiện thời của chứng thực.



# Quản lý khóa

- **Phân phối khóa công khai: Chứng thực khóa công khai (public key certificates)**

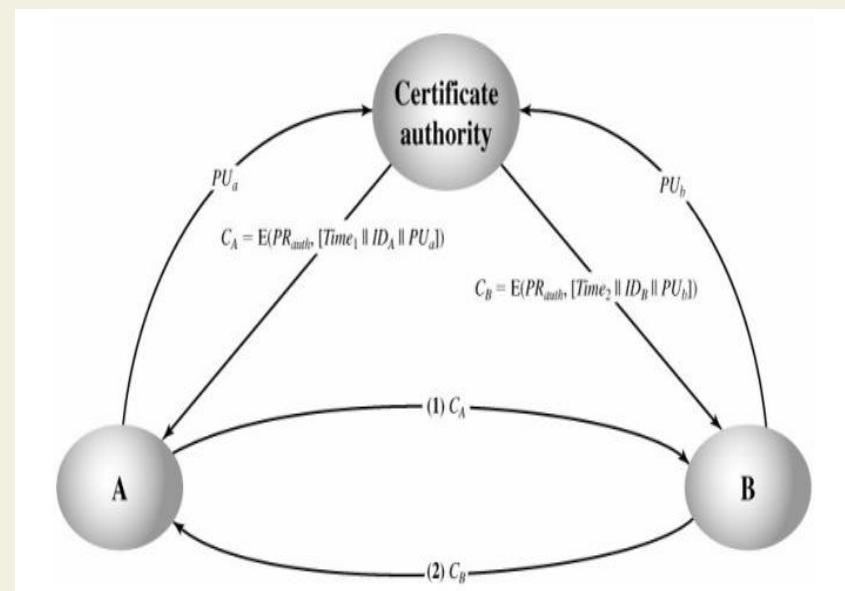
Đối với bên A, tổ chức quản lý chứng thực cung cấp cho chứng thực dưới dạng  $C_A = E(PR_{auth}, [T||ID_A||PU_a])$ .

Trong đó,  $PR_{auth}$  là khóa bí mật của tổ chức cấp chứng thực,  $T$  là nhãn thời gian để phản ánh tính hiện thời của chứng thực,  $ID_A$  là định danh của A và  $\parallel$  là phép ghép.

Sau đó, A có thể chuyển chứng thực này cho bất kỳ người tham gia nào khác, những người này đọc và xác minh chứng thực như sau:  $D(PU_{auth}, CA) = D(PU_{auth}, E(PR_{auth}, [T||ID_A||PU_a])) = (T||ID_A||PU_a)$ .

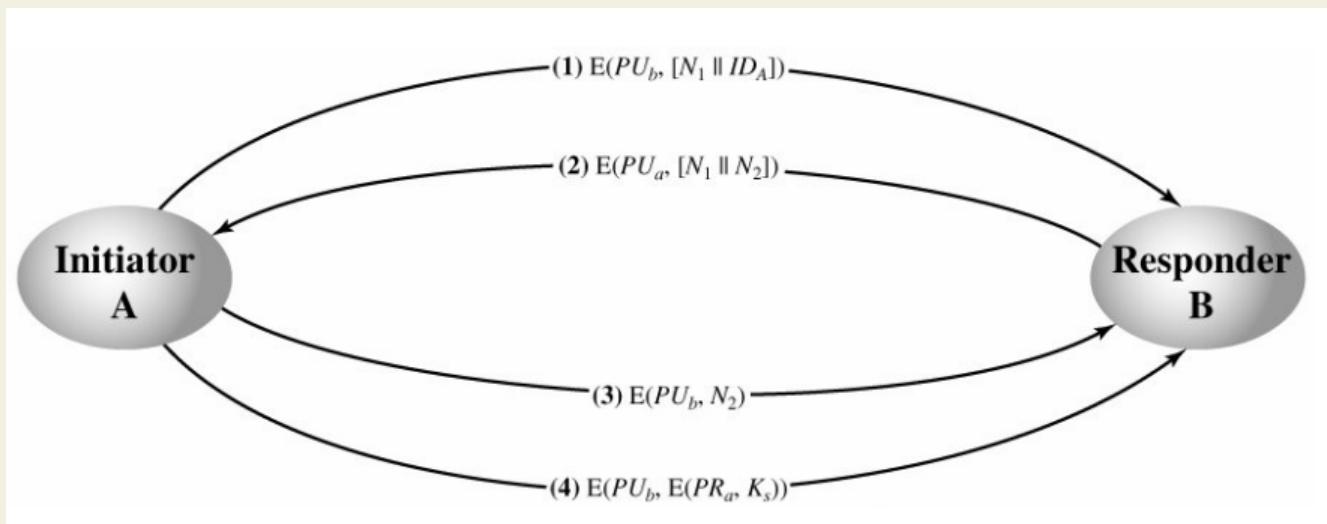
Người nhận sử dụng khóa công khai của tổ chức cấp chứng thực,  $PU_{auth}$  để giải mã chứng thực.

Các phần tử  $IDA$  và  $PUA$  cung cấp cho người nhận tên và khóa công khai của chủ sở hữu chứng chỉ. Nhãn thời gian  $T$  xác định tính hiện thời của chứng thực.



# Quản lý khóa

- **Phân phối khóa bí mật sử dụng hệ thống mã hóa công khai**
- Mã hóa khóa công khai được dùng để thiết lập khóa bí mật cho mỗi phiên trao đổi dữ liệu. Lúc này khóa bí mật được gọi là khóa phiên (session key), các phiên trao đổi dữ liệu khác nhau sẽ dùng các khóa bí mật khác nhau



Trao đổi khóa bí mật sử dụng hệ thống mã hóa công khai

# Quản lý khóa

- Phân phối khóa bí mật sử dụng hệ thống mã hóa công khai**

Quá trình trao đổi khóa bí mật được thực hiện qua các bước

sau:

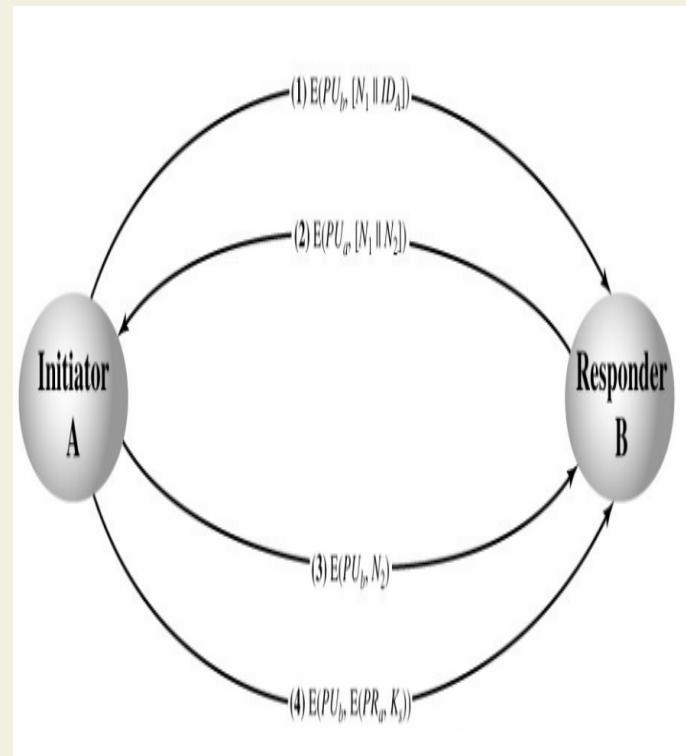
**Bước 1:** Bên A sử dụng khóa công khai của bên B ( $PU_b$ ) để mã hóa thông điệp bao gồm một số ngẫu nhiên chỉ sử dụng 1 lần (nonce)  $N_1$  và định danh của A ( $ID_A$ ).

**Bước 2:** Bên B gửi một thông điệp đến bên A bao gồm số ngẫu nhiên sử dụng 1 lần của A ( $N_1$ ), cùng một số ngẫu nhiên mới được tạo bởi B ( $N_2$ ) được mã hóa bằng khóa công khai của A ( $PU_a$ ). Bởi vì chỉ có B mới có thể giải mã được thông điệp (1) do A gửi, sự hiện diện của  $N_1$  trong thông điệp (2) đảm bảo cho A rằng chính là B.

**Bước 3:** Bên A trả về cho bên B thông điệp chứa  $N_2$  được mã hóa bằng mã công khai của B ( $PU_b$ ) để đảm bảo cho B rằng thông điệp này do A gửi.

**Bước 4:** Bên A lựa chọn khóa bí mật  $K_s$  và gửi thông điệp được mã hóa  $M = E(PU_b, E(PR_a, K_s))$  cho B. Mã hóa thông điệp bằng mã công khai của B để đảm bảo rằng chỉ B mới có thể đọc được, mã hóa bằng khóa riêng của A ( $PR_a$ ) để đảm bảo rằng chỉ có A mới có thể gửi thông điệp này.

**Bước 5:** B giải mã để khôi phục lại khóa bí mật  $K_s$



# Quản lý khóa

- **Trao đổi khóa Diffie Hellman**

Mục đích của thuật toán là để hai người dùng có thể trao đổi khóa một cách an toàn và khóa này được sử dụng để mã hóa cho các thông điệp trao đổi sau đó. Thuật toán được thực hiện như sau:

- Đầu tiên 2 bên sử dụng công khai số nguyên tố  $q$  và  $\alpha$  là primary root của  $q$ . Tiếp theo, bên A chọn một số nguyên ngẫu nhiên  $X_A < q$  và tính  $Y_A = \alpha^{X_A} \text{ mode } q$ .
- Tương tự, bên B chọn một số ngẫu nhiên  $X_B < q$  và tính  $Y_B = \alpha^{X_B} \text{ mode } q$ .
- Cả 2 bên giữ X bí mật và gửi Y công khai cho nhau.
- Cuối cùng, bên A tính được khóa  $K_A = (Y_B)^{X_A} \text{ mode } q$  và B tính được  $K_B = (Y_A)^{X_B} \text{ mode } q$ . Hai giá trị  $K_A$  và  $K_B$  tính được là trùng nhau

$$K_A = (Y_B)^{X_A} \text{ mod } q = (\alpha^{X_B} \text{ mod } q)^{X_A} \text{ mod } q = (\alpha^{X_B})^{X_A} \text{ mod } q = \alpha^{X_A \cdot X_B} \text{ mod } q$$

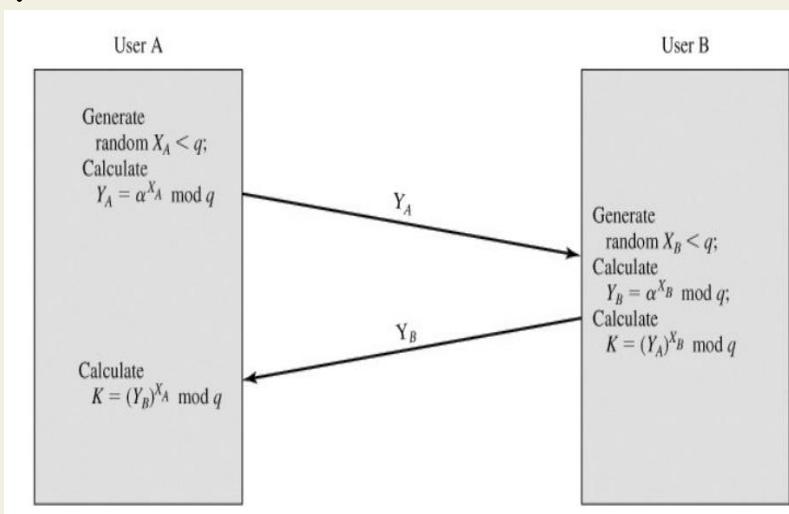
$$K_B = (Y_A)^{X_B} \text{ mod } q = (\alpha^{X_A} \text{ mod } q)^{X_B} \text{ mod } q = (\alpha^{X_A})^{X_B} \text{ mod } q = \alpha^{X_A \cdot X_B} \text{ mod } q$$

*Khóa K này có thể sử dụng làm khóa bị mất cho thuật toán mã hóa đối xứng*

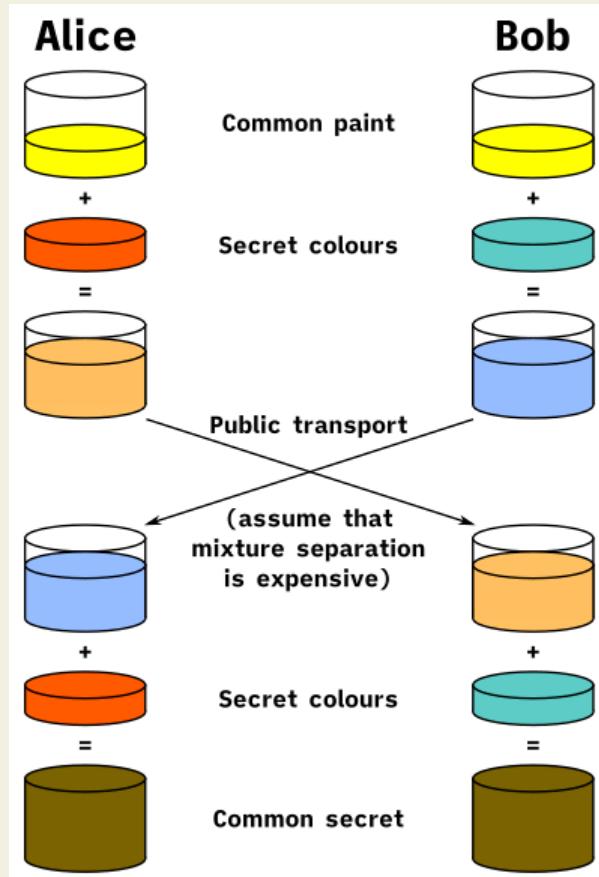
# Quản lý khóa

- Trao đổi khóa Diffie Hellman

*Mình họa*



Tuy nhiên, phương pháp trao đổi khóa theo thuật toán Diffie Hellman không chống được hình thức tấn công kẽ ở giữa (man in the middle attack)



### III. Mã hóa công khai

*Một câu hỏi khá thú vị là có thể đảo vai trò của public key và private key hay không?*

**Chữ ký số - Digital signature**

### III. Mã hóa công khai

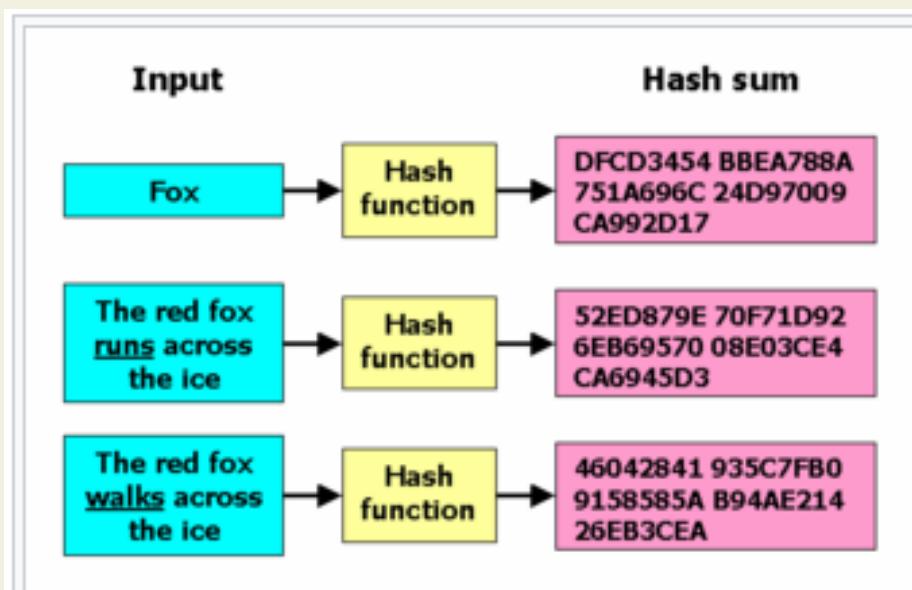
## Cryptographic Hash – Hàm băm mật mã

- *Cryptographic hash function* là một hàm băm với một số tính chất bảo mật nhất định để phù hợp việc sử dụng trong nhiều ứng dụng bảo mật thông tin đa dạng, chẳng hạn như chứng thực (authentication) và kiểm tra tính nguyên vẹn của thông điệp (*message integrity*)
- Một hàm băm nhận đầu vào là một xâu ký tự dài (hay *message*) có độ dài tùy ý và tạo ra kết quả là một xâu ký tự có độ dài cố định
- Một số hàm băm thông dụng: **MD5, SHA-1**

### III. Mã hóa công khai

## Cryptographic Hash – Hàm băm mật mã

- Ví dụ:



### III. Mã hóa công khai

## Chữ ký số sử dụng RSA

- Việc ký tên và xác thực chữ ký số sử dụng hệ mã hóa RSA tương tự như quá trình mã hóa mà giải mã ở trên
- Tuy nhiên vai trò của public key và private thì có thay đổi
- Để tạo chữ ký, người gửi sẽ dùng private key và người nhận sẽ dùng public key để xác thực chữ ký đó.
- Tuy nhiên, vì bản tin rất dài nên việc mã hóa toàn bộ bản tin sẽ rất mất thời gian
- Chữ ký số thường sử dụng phương pháp mã hóa giá trị **hash** của bản tin.

### III. Mã hóa công khai

## Chữ ký số sử dụng RSA

- Các hàm hash là hàm 1 chiều, vì vậy dù có được hash cũng không thể biết được bản tin gốc
- Độ dài hash là cố định và thường rất nhỏ, vì vậy chữ số sẽ không chiếm quá nhiều dung lượng
- Giá trị hash còn có thể dùng để kiểm tra lại bản tin nhận được có nguyên vẹn hay không?
- Chữ ký số đem lại nhiều giá trị hơn chữ ký tay rất nhiều
- Việc xử lý chữ ký số phức tạp hơn hẳn chữ ký tay truyền thống.

### III. Mã hóa công khai

## Chữ ký số sử dụng RSA

### Xác định nguồn gốc

- Hệ mã hóa bất đối xứng cho phép tạo chữ ký với private key mà chỉ người sở hữu chữ ký mới biết.
- Khi nhận gói tin:
  - Người nhận xác thực chữ ký bằng cách dùng public key giải mã,
  - Sau đó tính giá trị hash của bản tin gốc và so sánh với hash trong gói tin nhận được,
  - Hai chuỗi này phải trùng khớp với nhau

### III. Mã hóa công khai

## Chữ ký số sử dụng RSA

### Dữ liệu được giữ một cách toàn vẹn

- Tin nhắn gửi từ chủ private key rất khó có thể bị giả mạo
- Không thể thay đổi tin nhắn được vì không có private key để sửa đổi chữ ký số cho phù hợp.

### Chữ ký số không thể phủ nhận

- Trong giao dịch, một gói tin kèm chữ ký số rất dễ dàng tìm ra được nguồn gốc của chữ ký đó.
- Bởi vì private key là bí mật và chỉ người chủ của nó mới có thể biết, họ không thể chối cãi rằng chữ ký này không phải do họ phát hành

## Câu hỏi ôn tập

1. Nêu các đặc điểm của DES, và so sánh với 3-DES, AES?
2. So sánh ưu nhược điểm của mã hóa bí mật và mã hóa công khai?
3. Ứng dụng của AES và RSA?
4. Hàm hash một chiều là gì? Ứng dụng của hash-function?
5. Thuật toán trao đổi khóa Diffie-Hellman không an toàn trước các cuộc tấn công Man-in-the-Middle như thế nào?

# AN TOÀN IP VÀ WEB

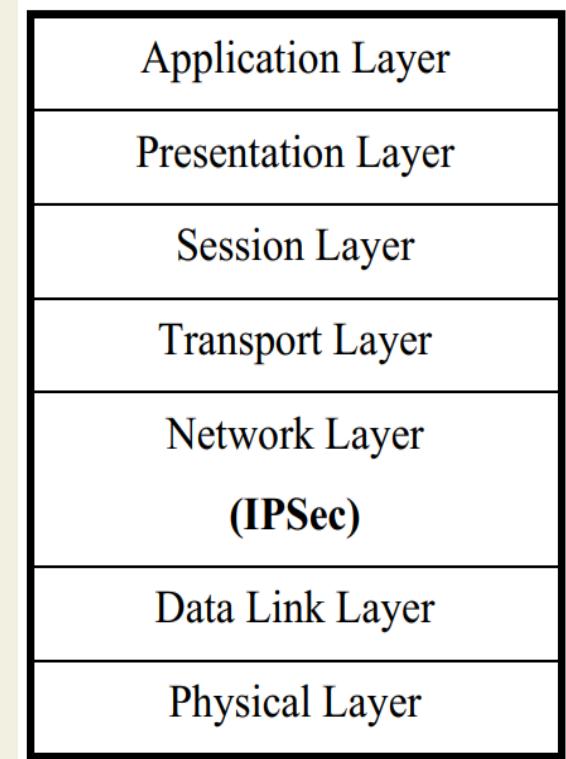
## An toàn IP

- Bảo mật cấp độ IP bao gồm ba chức năng: xác thực, bảo mật và quản lý khóa.
  - Chức năng xác thực đảm bảo rằng gói tin nhận được là đúng từ phía người gửi. Ngoài ra, cơ chế này đảm bảo rằng gói tin không bị thay đổi trong quá trình truyền tải.
  - Chức năng bảo mật cho phép các nút giao tiếp mã hóa thông báo để ngăn chặn việc nghe trộm bởi các bên thứ ba.
  - Chức năng quản lý khóa liên quan đến việc trao đổi khóa an toàn.
- Internet là một hệ thống thông tin toàn cầu gồm các mạng máy tính được liên kết với nhau. Hệ thống này truyền thông tin theo kiểu nối chuyển gói dữ liệu (packet switching) dựa trên một giao thức liên mạng đã được chuẩn hóa (giao thức IP)
- Để bảo mật các dữ liệu qua mạng Internet thì việc sử dụng giao thức IPSec là một trong những giải pháp hiện nay

# An toàn IP

## • Giới thiệu giao thức IPSec (IP security)

- ✓ IP Security (IPSec – Internet Protocol Security) là một giao thức được chuẩn hoá bởi IETF (Internet Engineering Task Force) từ năm 1998.
- ✓ Giao thức IPSec được xây dựng nhằm mục đích: nâng cấp các cơ chế mã hoá và xác thực thông tin cho chuỗi thông tin truyền đi trên mạng bằng giao thức IP.
- ✓ Hay nói cách khác, IPSec là sự tập hợp của các chuẩn mở được thiết lập để đảm bảo vấn đề bảo mật dữ liệu, tính toàn vẹn dữ liệu và chứng thực dữ liệu giữa các thiết bị mạng



# An toàn IP

## *Đánh giá giao thức IPSec* **Ưu điểm:**

- ❖ IPSec có tính năng an toàn bảo mật cao. Khi IPSec được triển khai trên tường lửa hoặc bộ định tuyến của một mạng riêng thì tính năng an toàn của IPSec có thể áp dụng cho toàn bộ các truy cập vào ra của mạng riêng đó, các thành phần khác không cần phải xử lý thêm các công việc liên quan đến bảo mật.
- ❖ IPSec được thực hiện bên dưới lớp TCP và UDP, đồng thời nó hoạt động trong suốt đối với các lớp này. Do vậy, không cần phải thay đổi phần mềm hay cấu hình lại các dịch vụ khi IPSec được triển khai.
- ❖ IPSec có thể được cấu hình để hoạt động một cách trong suốt đối với các ứng dụng đầu cuối, điều này giúp che giấu những chi tiết cấu hình phức tạp mà người dùng phải thực hiện khi kết nối đến mạng nội bộ từ xa thông qua mạng Internet

## Đánh giá giao thức IPSec

### Nhược điểm:

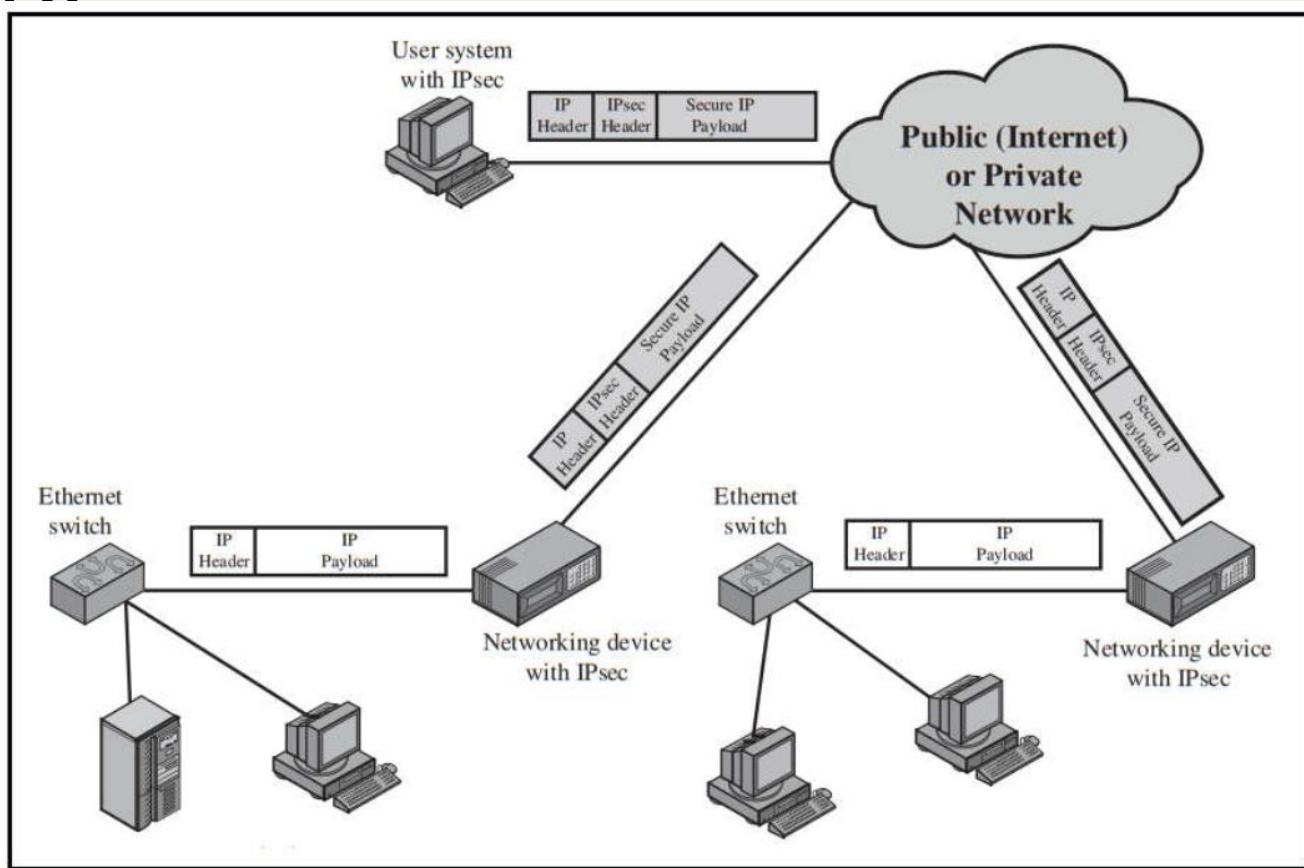
- ✓ Tất cả các gói được xử lý theo IPSec sẽ bị tăng kích thước do phải thêm vào các tiêu đề khác nhau, điều này làm cho thông lượng hiệu dụng của mạng giảm xuống. Vấn đề này có thể được khắc phục bằng cách nén dữ liệu trước khi mã hóa, song các kĩ thuật như vậy vẫn còn đang nghiên cứu và chưa được chuẩn hóa.
- ✓ IPSec được thiết kế chỉ để hỗ trợ bảo mật cho lưu lượng IP, không hỗ trợ các dạng lưu lượng khác.
- ✓ Việc tính toán nhiều giải thuật phức tạp trong IPSec vẫn còn là một vấn đề khó đối với các trạm làm việc và máy tính có cấu hình thấp.
- ✓ Việc phân phối các phần cứng và phần mềm mã hóa vẫn còn bị hạn chế đối với chính phủ tại một số quốc gia

## An toàn IP

**Ứng dụng của giao thức IPSec :** Giao thức IPsec cung cấp khả năng bảo mật thông tin liên lạc qua mạng LAN, mạng riêng ảo, mạng WAN và mạng Internet. Cụ thể như sau:

- **Kết nối văn phòng chi nhánh an toàn qua Internet:** mỗi công ty có thể xây dựng một mạng riêng ảo an toàn qua Internet hoặc qua mạng WAN công cộng;
- **Truy cập từ xa an toàn qua Internet:** người dùng cuối có hệ thống được trang bị với các giao thức bảo mật IP có thể thực hiện cuộc gọi nội mạng đến nhà cung cấp dịch vụ Internet (ISP) và truy cập an toàn vào mạng của công ty;
- **Thiết lập các kết nối mạng extranet và intranet:** IPsec có thể được sử dụng để bảo mật thông tin liên lạc với các tổ chức khác, đảm bảo tính xác thực và tính bảo mật cũng như cung cấp cơ chế trao đổi khóa;
- **Tăng cường an ninh thương mại điện tử:** mặc dù một số trang Web và các ứng dụng thương mại điện tử có các giao thức bảo mật được tích hợp sẵn, nhưng khi sử dụng IPsec sẽ tăng cường khả năng bảo mật đó. IPsec đảm bảo rằng tất cả lưu lượng do quản trị viên mạng chỉ định đều được mã hóa và xác thực, đồng thời thêm một lớp bảo mật bổ sung cho bất kỳ lưu lượng nào được cung cấp tại lớp ứng dụng.

## An toàn IP



Ứng dụng giao thức IPSec

## Lợi ích của sử dụng IPSec:

- Tại tường lửa hoặc bộ định tuyến, IPSec đảm bảo an ninh cho mọi luồng thông tin vượt biên
- Tại tường lửa, IPSec ngăn chặn thâm nhập trái phép từ Internet vào
- IPSec nằm dưới tầng giao vận (TCP, UDP), do vậy trong suốt với các ứng dụng
- IPSec có thể trong suốt với người dùng cuối
- IPSec có thể áp dụng cho người dùng đơn lẻ
- IPSec bảo vệ an ninh kiến trúc định tuyến

## Kiến trúc an ninh IP

- Đặc tả IPSec khá phức tạp
- Định nghĩa trong nhiều tài liệu:
  - Kiến trúc (RFC 4301), Authentication Header (RFC 4302), Encapsulating Security Payload (RFC 4303), Internet Key Exchange (RFC 4306)
    - AH không còn được sử dụng trong các ứng dụng mới
  - Các tài liệu mô tả các giải thuật mật mã:
    - Mã hóa, xác thực thông báo, hàm giả ngẫu nhiên, trao đổi khóa
  - Các tài liệu khác
    - Chính sách an ninh và cơ sở thông tin quản lý (MIB)
- Việc hỗ trợ IPSec là bắt buộc đối với IPv6, tùy chọn đối với IPv4

## Các dịch vụ IPsec

- RFC 4301 liệt kê các dịch vụ sau:

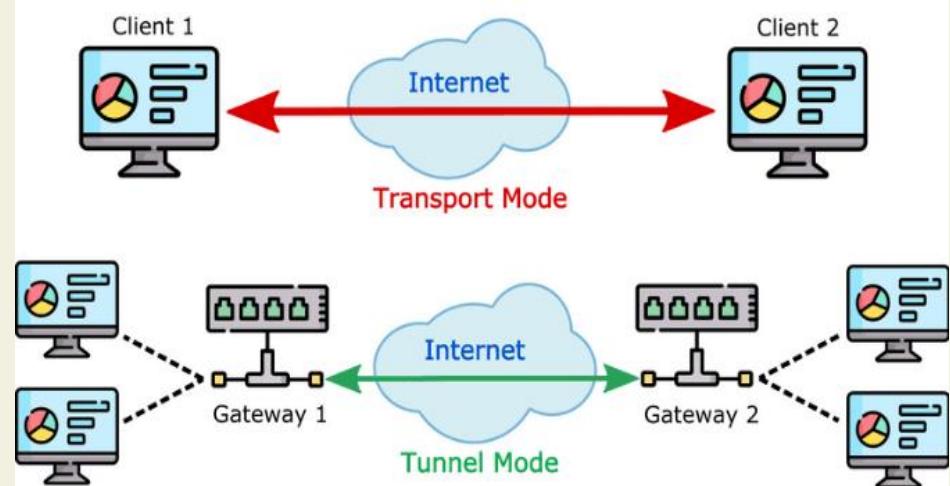
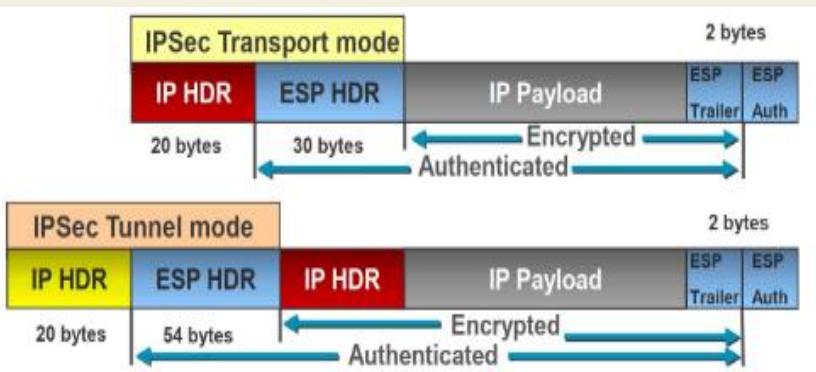
- Kiểm soát truy cập
- Tính toàn vẹn phi kết nối
- Xác thực nguồn gốc dữ liệu
- Từ chối các gói phát lại (một dạng toàn vẹn chuỗi một phần)
- Bảo mật (mã hóa)
- Bảo mật lưu lượng hạn chế

## Các giao thức chính của IPSec:

- Giao thức AH – Authentication Header hay Tiêu đề xác thực: là một giao thức xác thực được chỉ định bởi tiêu đề của giao thức. Mới bảo đảm được tính **xác thực**.
- Giao thức đóng gói tải trọng bảo mật ESP - Encapsulation Secure Payload : một giao thức mã hóa/xác thực kết hợp có nghĩa là bảo đảm cả tính **xác thực** và **mã hóa**
- Như vậy: AH đảm bảo tính xác thực và toàn vẹn gói tin Ip, gói tin không được mã hóa. Nếu cần thêm tính bí mật thì sử dụng ESP: ESP vừa bảo đảm tính toàn vẹn và bí mật

## Transport and Tunnel Modes

- Cả AH và ESP đều hỗ trợ hai chế độ sử dụng: chế độ vận chuyển và đường hầm (transport mode và tunnel mode). Tuy nhiên, hoạt động của hai chế độ này được hiểu rõ nhất trong bối cảnh mô tả ESP.
- Transport mode cung cấp cơ chế bảo vệ cho dữ liệu của các lớp cao hơn (TCP, UDP hoặc ICMP) trong khi đó Tunnel Mode sẽ bảo vệ toàn bộ gói dữ liệu.



## Transport and Tunnel Modes

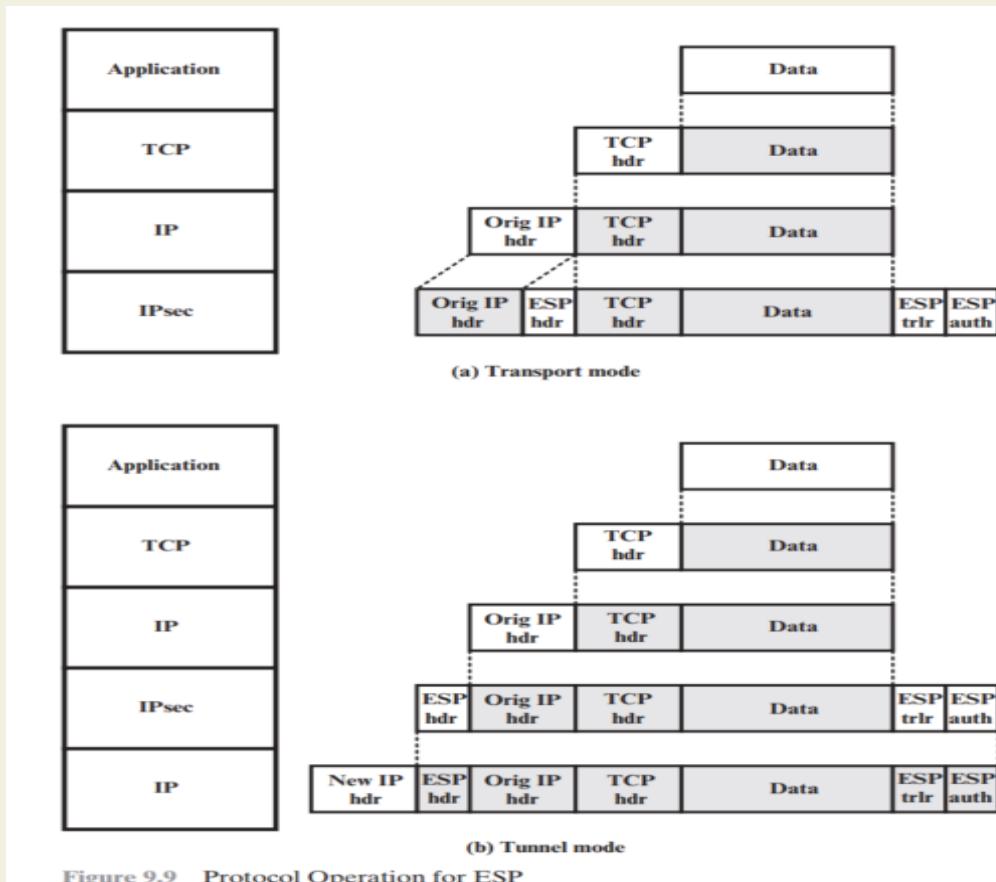
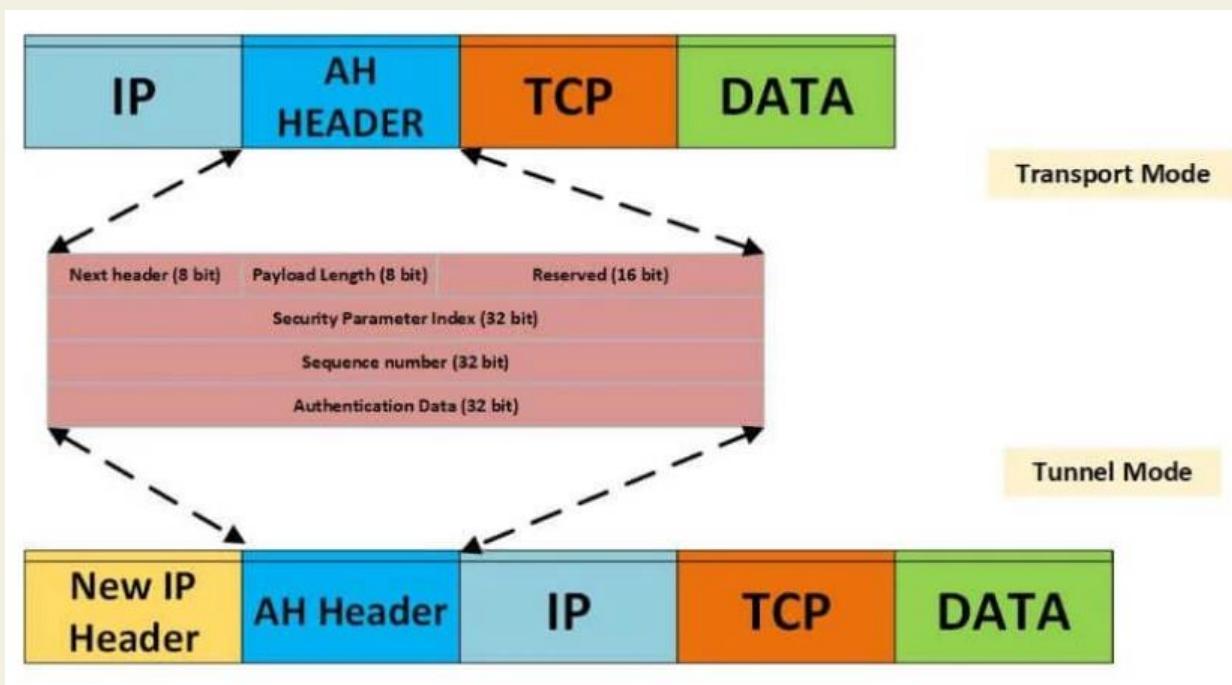


Figure 9.9 Protocol Operation for ESP

**Giao thức xác thực AH:** AH cho phép xác thực người dùng, xác thực ứng dụng và thực hiện các cơ chế lọc gói tương ứng. Ngoài ra AH còn có khả năng hạn chế các tấn công giả danh (spoofing) và tấn công phát lại (replay).

Cấu trúc gói:



## Giao thức xác thực AH

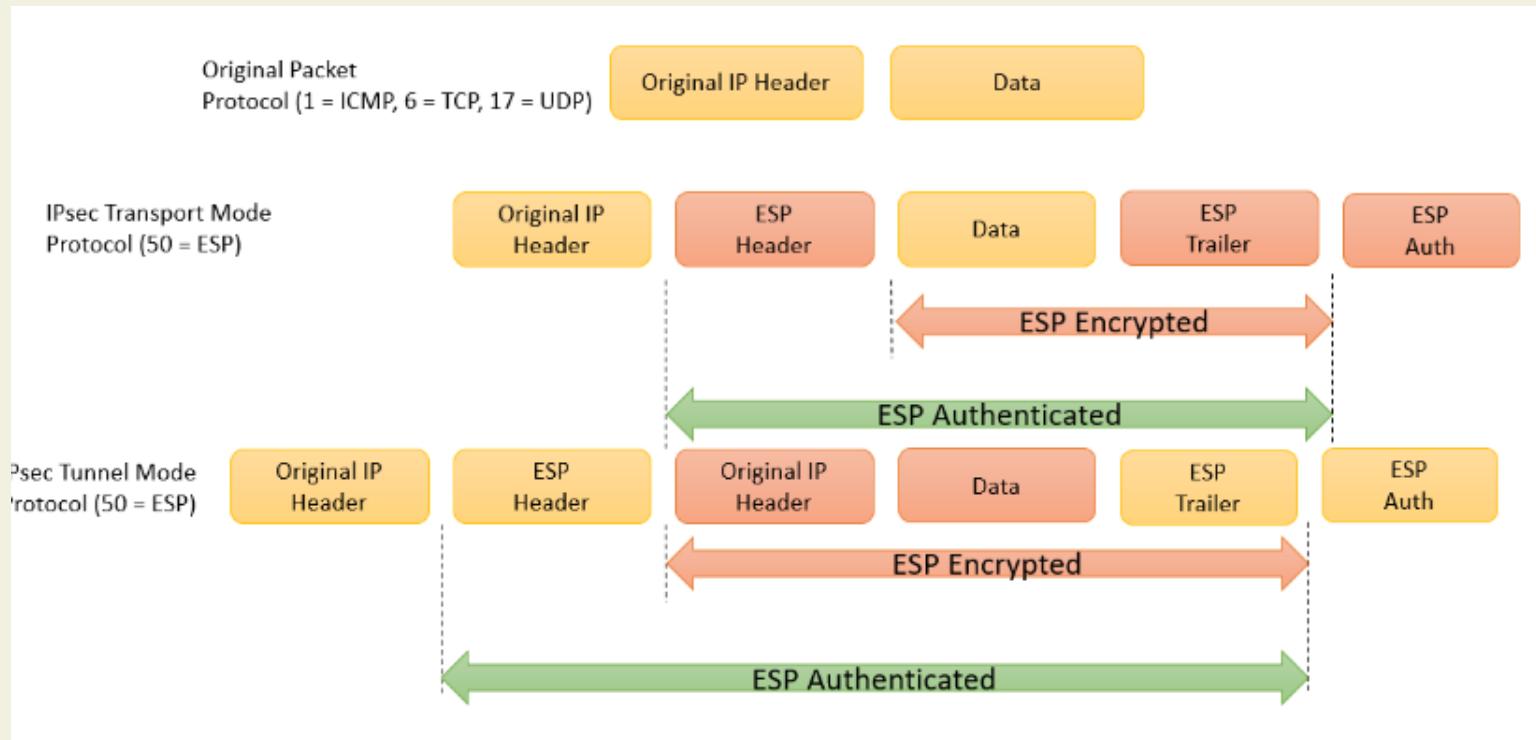
### Cơ chế xác thực:

- Xác thực từ đầu cuối đến đầu cuối (End-to-End Authentication): là trường hợp xác thực trực tiếp giữa hai hệ thống đầu cuối (giữa máy chủ với trạm làm việc hoặc giữa hai trạm làm việc), việc xác thực này có thể diễn ra trên cùng mạng nội bộ hoặc giữa hai mạng khác nhau, chỉ cần hai đầu cuối biết được khoá bí mật của nhau. Trường hợp này sử dụng chế độ vận chuyển (Transport Mode) của AH.
- Xác thực từ đầu cuối đến trung gian (End-to-Intermediate Authentication): là trường hợp xác thực giữa hệ thống đầu cuối với một thiết bị trung gian (router hoặc firewall). Trường hợp này sử dụng chế độ đường hầm (Tunnel Mode) của AH.

**Giao thức đóng gói ESP - Encapsulating Security Payload** là một lựa chọn khác để thực thi IPSec bên cạnh giao thức xác thực thông tin AH.

- Chức năng chính của ESP là cung cấp tính bảo mật cho dữ liệu truyền trên mạng IP bằng các kỹ thuật mã hóa.
- Tuy nhiên ESP cũng còn một tuỳ chọn khác là cung cấp cả dịch vụ bảo đảm tính toàn vẹn của dữ liệu thông qua cơ chế xác thực.
- Như vậy khi sử dụng ESP, người dùng có thể chọn hoặc không chọn chức năng xác thực, còn chức năng mã hóa là chức năng mặc định của ESP.

## Giao thức đóng gói ESP - Encapsulating Security Payload



## Giao thức đóng gói ESP

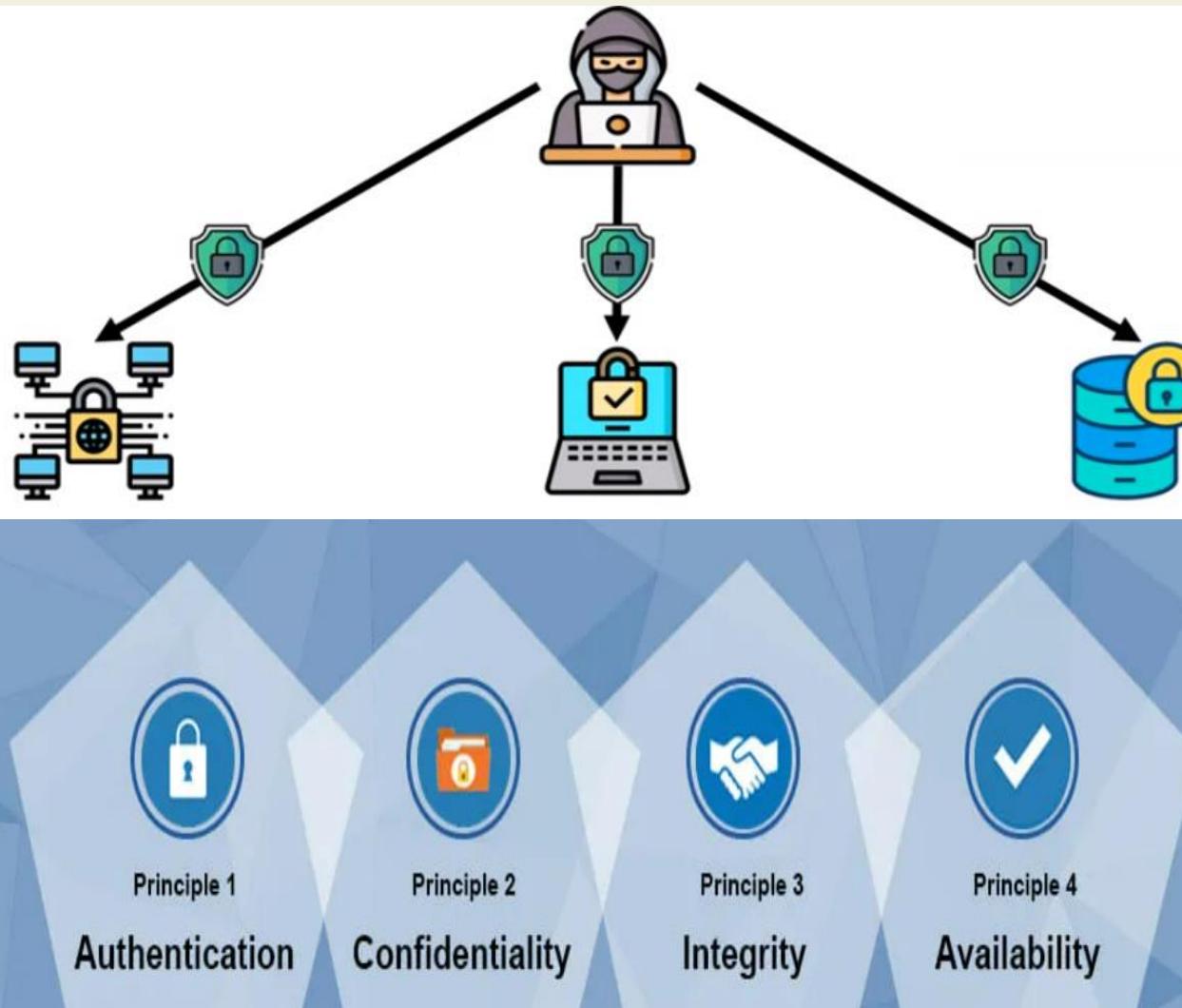
Nguyên tắc hoạt động:

- Về nguyên tắc hoạt động thì ESP sử dụng mật mã đối xứng để cung cấp sự mật hoá dữ liệu cho các gói tin IPSec.
- Cho nên, để kết nối của cả hai đầu cuối đều được bảo vệ bởi mã hoá ESP thì hai bên phải sử dụng key giống nhau mới mã hoá và giải mã được gói tin.
- Khi một đầu cuối mã hoá dữ liệu, nó sẽ chia dữ liệu thành các khối (block) nhỏ, và sau đó thực hiện thao tác mã hoá nhiều lần sử dụng các block dữ liệu và khóa (key).
- Khi một đầu cuối khác nhận được dữ liệu mã hoá, nó thực hiện giải mã sử dụng key giống nhau và quá trình thực hiện tương tự, nhưng trong bước này ngược với thao tác mã hoá.

# An toàn WEB

- ***Web security*** refers to networks, computer system and data are protected from unauthorized person or group.
- ***Purpose of Web Security*** is to prevent security attack like Passive attack and Active attack.

→ How can achieve ***Web Security?***



# An toàn WEB

## Giới thiệu về SSL

- SSL (Secure Socket Layer) là dịch vụ an toàn tầng vận chuyển (transport layer) được phát triển bởi Netscape
- SSL cung cấp khả năng mã hóa để bảo mật các kết nối giữa máy khách và máy chủ
- SSL có thể sử dụng để hỗ trợ các giao dịch an toàn cho rất nhiều ứng dụng khác nhau trên Internet.

### Nhiệm vụ:

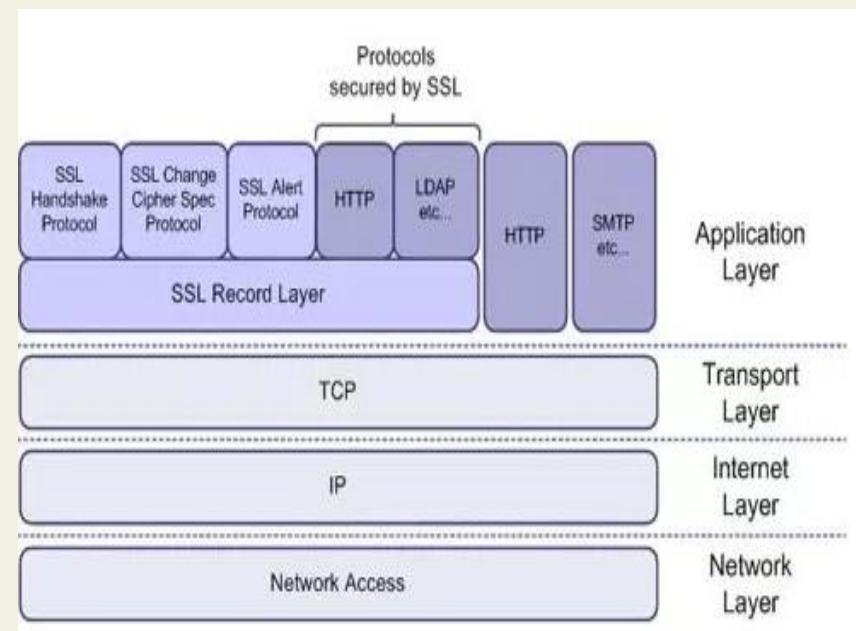
Xác thực máy chủ: cho phép người sử dụng xác thực được máy chủ muốn kết nối. Lúc này, phía browser sử dụng các kỹ thuật mã hoá công khai để chắc chắn rằng chứng chỉ và khoá công khai của máy chủ là có giá trị và được cấp phát bởi một CA trong danh sách các CA đáng tin cậy của máy trạm

Xác thực máy trạm: cho phép phía máy chủ xác thực được người sử dụng muốn kết nối. Phía máy chủ cũng sử dụng các kỹ thuật mã hoá công khai để kiểm tra xem chứng chỉ và khoá công khai của máy chủ có giá trị hay không và được cấp phát bởi một CA trong danh sách các CA đáng tin cậy không.

Mã hoá kết nối: tất cả các thông tin trao đổi giữa máy trạm và máy chủ được mã hoá trên đường truyền nhằm nâng cao khả năng bảo mật.

## • Cấu trúc của giao thức SSL:

- ✓ Các bên giao tiếp (nghĩa là client và server) có thể xác thực nhau bằng cách sử dụng mật mã khóa chung
- ✓ Sự bí mật của lưu lượng dữ liệu được bảo vệ vì nối kết được mã hóa trong suốt sau khi một sự thiết lập quan hệ ban đầu và sự thương lượng khóa session đã xảy ra.
- ✓ Tính xác thực và tính toàn vẹn của lưu lượng dữ liệu cũng được bảo vệ vì các thông báo được xác thực và được kiểm tra tính toàn vẹn một cách trong suốt bằng cách sử dụng MAC.



# An toàn WEB

## Giới thiệu về SSL

- Hoạt động của SSL dựa trên hai nhóm con giao thức là giao thức “bắt tay” và giao thức “bản ghi”. Các bước thực hiện trong quá trình bắt tay như sau:

1. **Máy trạm sẽ gửi cho máy chủ số phiên bản SSL** đang dùng, các tham số của thuật toán mã hoá, dữ liệu được tạo ra ngẫu nhiên (chữ ký số) và một số thông tin khác mà máy chủ cần để thiết lập kết nối với máy trạm.
2. **Máy chủ gửi cho máy trạm số phiên bản SSL** đang dùng, các tham số của thuật toán mã hoá, dữ liệu được tạo ra ngẫu nhiên và một số thông tin khác mà máy trạm cần để thiết lập kết nối với máy chủ
3. **Máy trạm sử dụng một số thông tin mà máy chủ** gửi đến để xác thực máy chủ
4. **SD tất cả các thông tin được tạo ra trong giai đoạn bắt tay ở trên, máy trạm (cùng với sự cộng tác của máy chủ và phụ thuộc vào thuật toán được sử dụng)** sẽ tạo ra premaster secret cho phiên làm việc, mã hoá bằng khoá công khai mà máy chủ gửi đến trong chứng chỉ ở bước 2, và gửi đến máy chủ.
5. **Nếu máy chủ có yêu cầu xác thực máy trạm**, thì phía máy trạm sẽ đánh dấu vào phần thông tin riêng chỉ liên quan đến quá trình “bắt tay” này mà hai bên đều biết.
6. **Máy chủ sẽ xác thực máy trạm.** Trường hợp máy trạm không được xác thực, phiến làm việc sẽ bị ngắt. Còn nếu máy trạm được xác thực thành công, máy chủ sẽ sử dụng khoá bí mật để giải mã premaster secret, sau đó thực hiện một số bước để tạo ra master secret

# An toàn WEB

## Giới thiệu về SSL

**7. Máy trạm và máy chủ sẽ sử dụng master secret** để tạo ra các khóa phiên, đó chính là các khoá đối xứng được sử dụng để mã hoá và giải mã các thông tin trong phiên làm việc và kiểm tra tính toàn vẹn dữ liệu

**8. Máy trạm sẽ gửi thông báo đến máy chủ thông báo** rằng các thông điệp tiếp theo sẽ được mã hoá bằng khoá phiên. Sau đó nó gửi một lời nhắn đã được mã hoá để thông báo rằng phía máy trạm đã kết thúc giai đoạn “bắt tay”

**9. Máy chủ gửi lại thông báo đến máy trạm thông báo rằng** các thông điệp tiếp theo sẽ được mã hoá bằng khoá phiên. Sau đó nó gửi một lời nhắn đã được mã hoá để thông báo rằng máy chủ đã kết thúc giai đoạn “bắt tay”.

**10. Lúc này giai đoạn “bắt tay” đã hoàn thành, và phiên làm việc SSL bắt đầu.** Cả hai phía máy trạm và máy chủ sẽ sử dụng các khoá phiên để mã hoá và giải mã thông tin trao đổi giữa hai bên, kiểm tra tính toàn vẹn dữ liệu

## Lợi ích về An toàn bảo mật khi sử dụng chứng chỉ SSL

**Chứng chỉ bảo mật SSL đã đem lại rất nhiều lợi ích về bảo mật cho website và trình duyệt web của người dùng**

- ✓ Xác thực website, giao dịch.
- ✓ Nâng cao hình ảnh, thương hiệu và uy tín doanh nghiệp
- ✓ Bảo mật các giao dịch giữa khách hàng và doanh nghiệp, các dịch vụ truy nhập hệ thống
- ✓ Bảo mật webmail và các ứng dụng như Outlook Web Access, Exchange và Office Communication Server
- ✓ Bảo mật các ứng dụng ảo hóa như Citrix Delivery Platform hoặc các ứng dụng điện toán đám mây
- ✓ Bảo mật dịch vụ FTP
- ✓ Bảo mật truy cập control panel
- ✓ Bảo mật các dịch vụ truyền dữ liệu trong mạng nội bộ, file sharing, extranet
- ✓ Bảo mật VPN Access Servers, Citrix Access Gateway

# An toàn WEB

- **Giao thức TLS:** là sự kế thừa và thay thế SSL
  - TLS protocol là một giao thức quan trọng để bảo mật các kết nối mạng trực tuyến.
  - Các kết nối này được bảo mật bằng cách sử dụng mật mã đối xứng để mã hóa dữ liệu truyền đi
  - Các keys được tạo ra duy nhất cho mỗi kết nối và dựa trên một chia sẻ bí mật ở đầu phiên kết nối gọi là TLS handshake
  - TLS protocol được sử dụng rộng rãi trên Internet để bảo vệ các giao tiếp dữ liệu giữa client and server, bao gồm cả trang web, email và các dịch vụ mạng khác.
  - TLS là sự kế thừa cho SSL (Secure Sockets Layer )
  - Có thể nói rằng giao thức TLS v1.0 được phát triển dựa trên giao thức SSL v3.0 nhưng giữa chúng có những điểm khác biệt

- **Chức năng của giao thức của TLS:**
- Chức năng chính của giao thức TLS là cung cấp sự riêng tư bảo đảm sự nguyên vẹn cho dữ liệu giữa hai ứng dụng trong môi trường mạng.
- Vì TLS là giao thức được phát triển từ giao thức SSL nên giao thức TLS cũng theo mô hình client-server.
- Trong mô hình TCP/IP thì giao thức TLS gồm có hai lớp: Lớp Record Layer và lớp Handshake Layer.
- Record layer là lớp thấp nhất gồm TLS record protocol (trên tầng giao vận như giao thức điều khiển truyền tải TCP, giao thức truyền vận không tin cậy [UDP](#)).

- **Chức năng của giao thức của TLS:**

- Tính năng kết nối riêng tư: ứng dụng mã hoá đối xứng được sử dụng để mã hoá dữ liệu (mã hoá AES...).
- Các khoá để mã hoá đối xứng được sinh ra trong mỗi lần thực hiện kết nối, được thỏa thuận bí mật của giao thức khác (ví dụ TLS).
- Nhờ vậy mà giao thức TLS có thể được sử dụng mà không cần mã hoá.
- Tính năng kết nối đáng tin cậy: Một thông điệp vận chuyển thông báo sẽ bao gồm kiểm tra tính toàn vẹn (sử dụng hàm Băm ví dụ SHA-1).
- Không chỉ có vậy, giao thức TLS còn có thể sử dụng để đóng gói, mã hóa dữ liệu, phân mảnh, hỗ trợ các máy chủ nhận ra nhau để từ đó tiến hành thỏa thuận mã hóa.

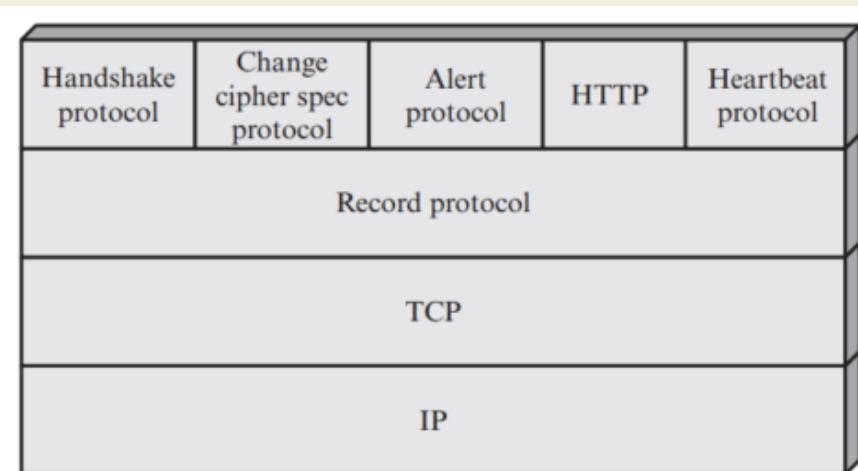
- **Cấu trúc TLS:**

TLS được thiết kế để sử dụng TCP nhằm cung cấp dịch vụ bảo mật đầu cuối đáng tin cậy

Hai khái niệm TLS quan trọng là phiên TLS (session) và kết nối TLS (connection)

- Kết nối: Đối với TLS, các kết nối là mối quan hệ ngang hàng. Các kết nối là tạm thời. Mỗi kết nối được liên kết với một phiên.
- Phiên: Phiên TLS là sự kết hợp giữa máy khách và máy chủ. Các phiên được tạo bởi Giao thức bắt tay. Các phiên xác định một tập hợp các tham số bảo mật bằng mật mã, có thể được chia sẻ giữa nhiều kết nối. Các phiên được sử dụng để tránh thương lượng tốn kém các tham số bảo mật mới cho mỗi kết nối.

*Về cơ bản, kiến trúc của TLS cũng tương tự như SSL. Tuy nhiên, có bổ sung thêm Heartbeat Protocol*



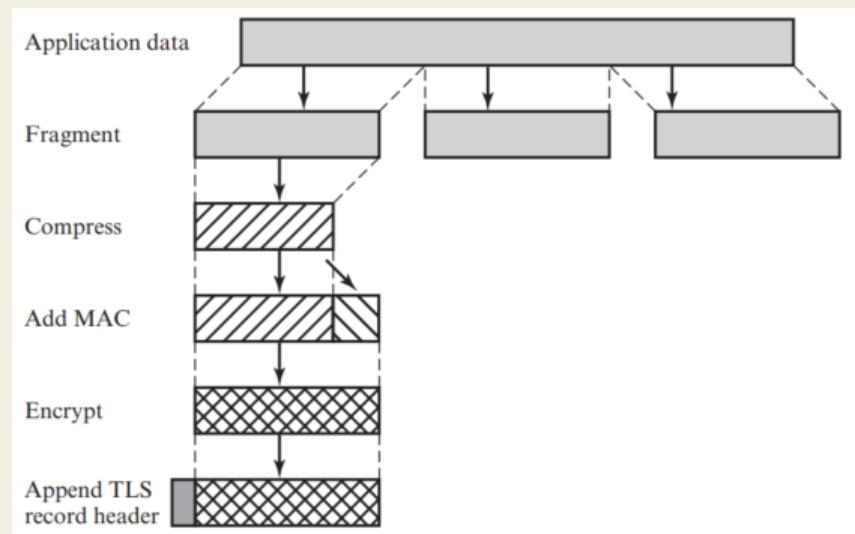
# An toàn WEB

- **Cấu trúc TLS:**

**Giao thức TLS Record:** cung cấp hai dịch vụ cho các kết nối TLS

Tính bí mật: Giao thức bắt tay xác định một khóa bí mật dùng chung được sử dụng để mã hóa thông thường các tải trọng TLS

Tính toàn vẹn của thông báo: Giao thức bắt tay cũng xác định khóa bí mật dùng chung được sử dụng để tạo mã xác thực tin nhắn (MAC).



# An toàn WEB

- **Cấu trúc TLS:**

**Alert Record:** cung cấp hai dịch vụ cho các kết nối TLS

- ✓ Giao thức cảnh báo được sử dụng để truyền các cảnh báo liên quan đến TLS đến thực thể ngang hàng
- ✓ Các thông báo cảnh báo được nén và mã hóa, như được chỉ định bởi trạng thái hiện tại
- ✓ Mỗi thông báo trong giao thức này bao gồm hai byte (Hình dưới). Byte đầu tiên nhận giá trị cảnh báo (1) hoặc nghiêm trọng (2) để truyền tải mức độ nghiêm trọng của thông báo.
- ✓ Nếu mức độ nghiêm trọng, TLS sẽ ngay lập tức chấm dứt kết nối
- ✓ Các kết nối khác trong cùng một phiên có thể tiếp tục, nhưng không có kết nối mới nào trong phiên này có thể được thiết lập.

# An ninh tầng giao vận (Transport –Level security)

- **Cấu trúc TLS:**

## **Alert Record:**

Byte thứ hai chứa mã cho biết cảnh báo cụ thể.

- unexpected\_message: Đã nhận được một tin nhắn không phù hợp.
- bad\_record\_mac: Đã nhận được MAC không chính xác.
- ecompression\_failure: Chức năng giải nén nhận đầu vào không đúng (ví dụ: không thể giải nén hoặc giải nén lớn hơn mức tối đa cho phép chiều dài).
- handshake\_failure: Người gửi không thể thương lượng một bộ tham số bảo mật có thể chấp nhận được với các tùy chọn có sẵn.
- invalid\_parameter: Một trường trong thông báo bắt tay nằm ngoài phạm vi hoặc không phù hợp với các trường khác.

- **Cấu trúc TLS:**

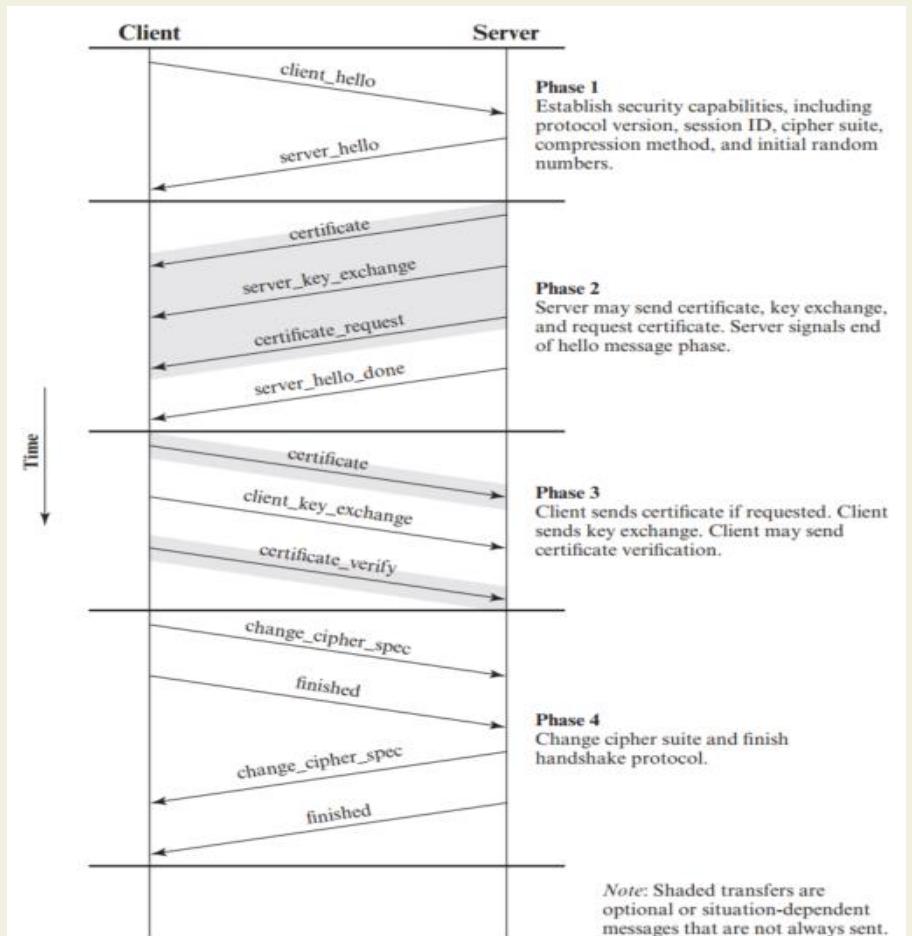
## **Handshake Protocol: Đây là phần phức tạp nhất của TLS**

- ✓ Giao thức này cho phép máy chủ và máy khách xác thực lẫn nhau và thương lượng một thuật toán mã hóa và MAC cũng như các khóa mật mã được sử dụng để bảo vệ dữ liệu được gửi trong bản ghi TLS.
- ✓ Giao thức bắt tay được sử dụng trước khi bất kỳ dữ liệu ứng dụng nào được truyền đi.
- ✓ Giao thức bắt tay bao gồm một loạt các thông báo được trao đổi bởi máy khách và máy chủ

- Cấu trúc TLS:

## Handshake Protocol: 4 pha

**Pha 1.** Thiết lập khả năng bảo mật  
Giai đoạn 1 bắt đầu một kết nối logic và thiết lập các khả năng bảo mật sẽ được liên kết với nó. Quá trình trao đổi được bắt đầu bởi ứng dụng khách, ứng dụng này sẽ gửi message `client_hello` với các tham số: Version (TLS version), Random, Session ID



## • Cấu trúc TLS:

### Handshake Protocol: 4 pha

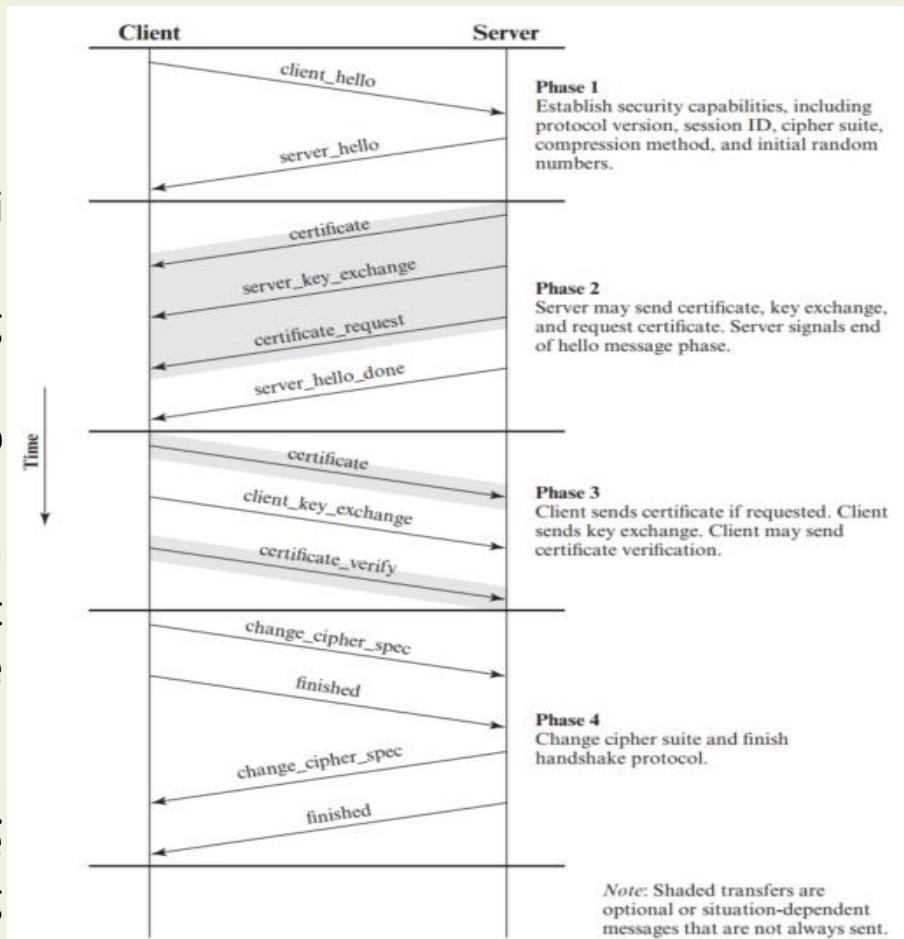
**Pha 2.** Xác thực máy chủ server và trao đổi khóa

Máy chủ bắt đầu pha này bằng cách gửi chứng chỉ của nó nếu nó cần được xác thực

Sau đó, một message server\_key\_exchange có thể được gửi nếu cần thiết.

Sau đó một máy chủ không ẩn danh có thể yêu cầu chứng chỉ từ client là certificate\_request message – chứng chỉ này gồm certificate\_type và certificateAuthorities

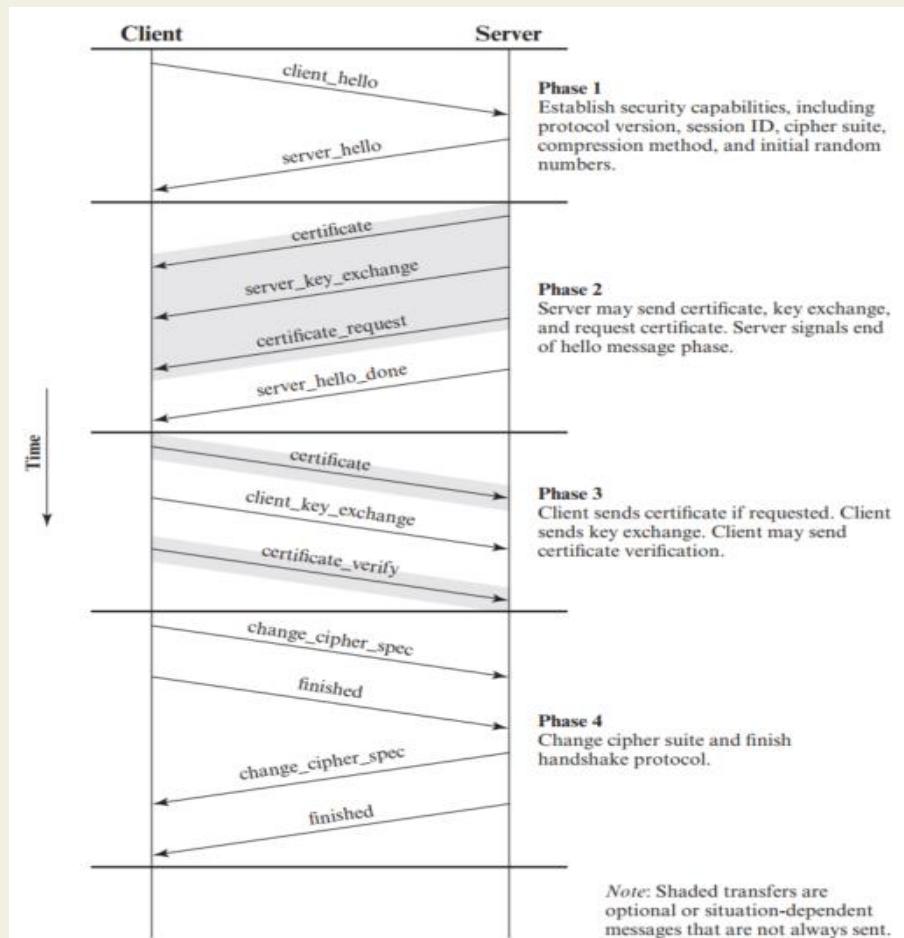
Message cuối cùng trong phase 2 là server\_done message – được gửi từ server để cho biết đã kết thúc “server hello” và các thông báo liên quan.



- Cấu trúc TLS:

## Handshake Protocol: 4 pha

Pha 3. Xác thực Client và trao đổi  
khóa



- Cấu trúc TLS:

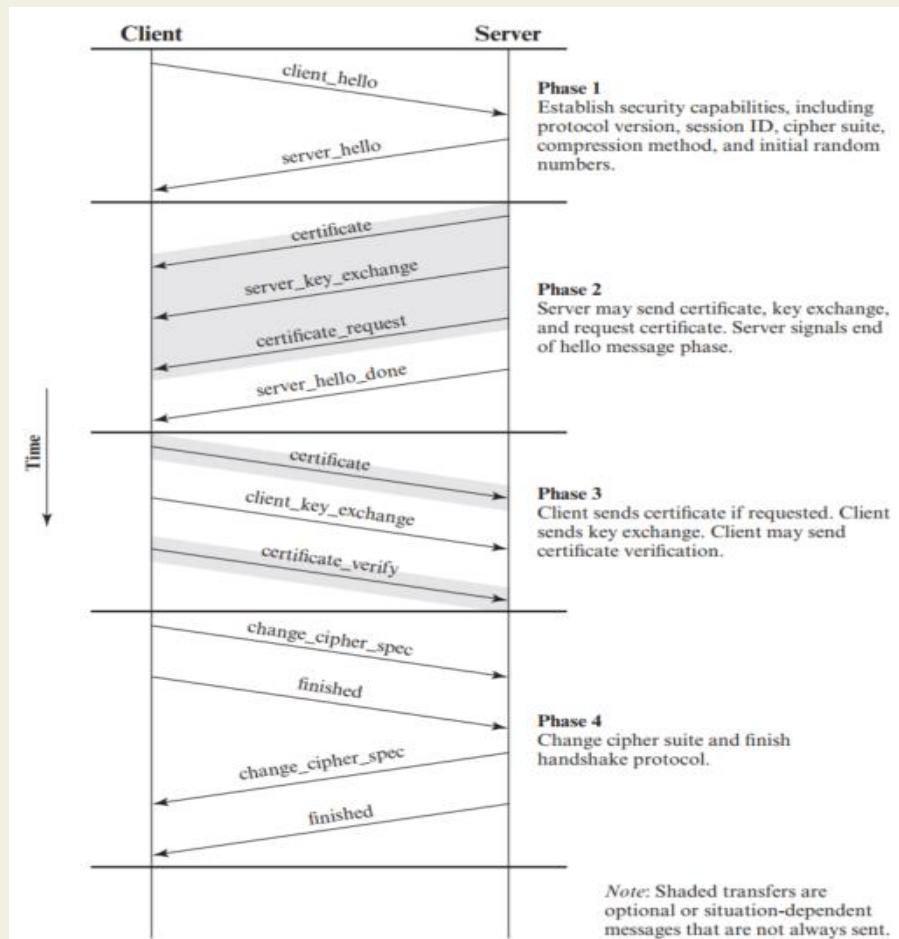
## Handshake Protocol: 4 pha

Pha 4. Kết thúc – hoàn tất thiết lập kết nối an toàn.

Client gửi change\_cipher\_spec message

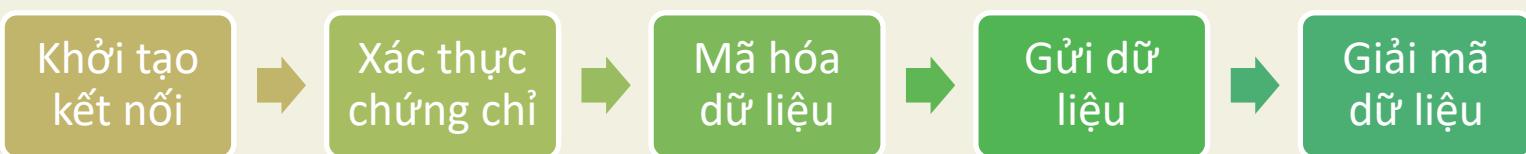
Client gửi finished message theo các thuật toán, khóa và bí mật mới.

Finished message xác minh rằng quá trình trao đổi khóa và xác thực thành công

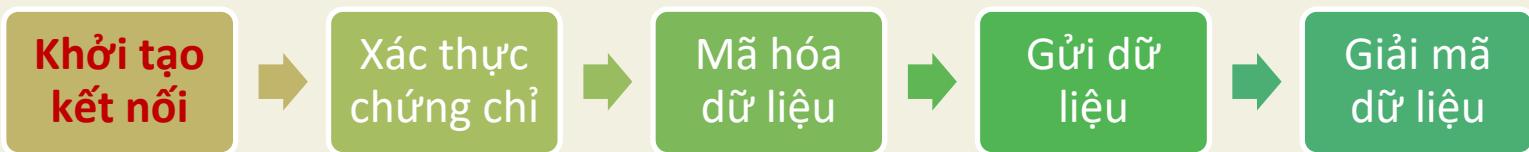


- **Cơ chế hoạt động của TLS:**

- TLS protocol hoạt động bằng cách sử dụng private key và cơ chế xác thực để bảo vệ dữ liệu truyền đi trên mạng
- Để hiểu rõ hơn cơ chế hoạt động của TLS, ta có thể phân tích quá trình bảo mật kết nối mạng theo các bước sau:



- Cơ chế hoạt động của TLS:



Trong quá trình **khởi tạo kết nối**, hai bên kết nối (ví dụ máy tính người dùng và *Web server*) sẽ trao đổi các thông tin cơ bản như phiên bản của TLS, các thuật toán *mã hóa* và các *chứng chỉ bảo mật*.

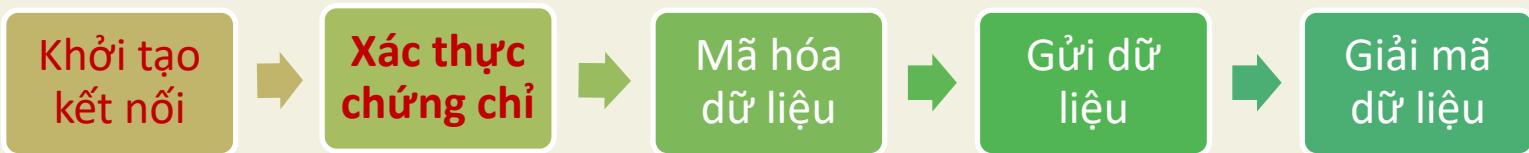
Web server: chịu trách nhiệm xử lý các yêu cầu truy cập từ các thiết bị khác (người dùng) và trả lại tài nguyên tương ứng. Có thể hiểu nó đóng vai trò quan trọng trong cung cấp nội dung cho người dùng trên Internet.

Thuật toán mã hóa: đối xứng và công khai

Chứng chỉ bảo mật: là các lối chứng chỉ bảo mật SSL (Secure Sockets layer) – là tiêu chuẩn an ninh công nghệ toàn cầu tạo ra một liên kết được mã hóa giữa web server và trình duyệt.

# An toàn WEB

- Cơ chế hoạt động của TLS:



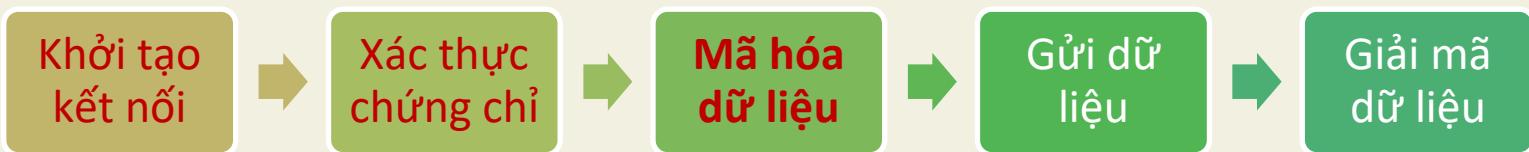
**Xác thực chứng chỉ:** Sau khi 2 bên đã trao đổi các thông tin cơ bản, server sẽ gửi một chứng chỉ bảo mật đến máy tính của người dùng.

Máy tính người dùng sẽ kiểm tra chứng chỉ này bằng cách sử dụng các chứng chỉ của tổ chức xác thực đã được lưu trữ trên hệ thống

Nếu chứng chỉ được xác thực, kết nối sẽ tiếp tục, ngược lại sẽ ngắt kết nối

# An toàn WEB

- Cơ chế hoạt động của TLS:



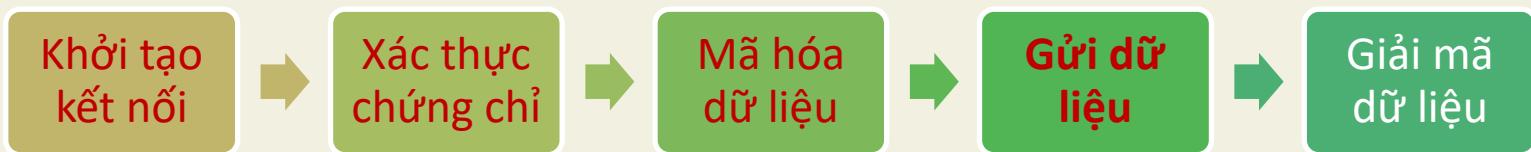
**Mã hóa dữ liệu:** Sau khi chứng chỉ đã được xác thực, hai bên sẽ sử dụng các private key để mã hóa dữ liệu trước khi gửi đi.

Những khóa này được tạo ra từ quá trình trao đổi thông tin cơ bản ban đầu và sẽ khác nhau giữa hai bên.

Khi dữ liệu được gửi đi, nó sẽ được mã hóa bằng các khóa này và chỉ người nhận dữ liệu mới có thể giải mã nó bằng các khóa bí mật của họ.

Nhờ có mã hóa dữ liệu, người dùng có thể yên tâm rằng dữ liệu của họ không bị truy cập bởi bất kì ai khác ngoài người nhận dữ liệu

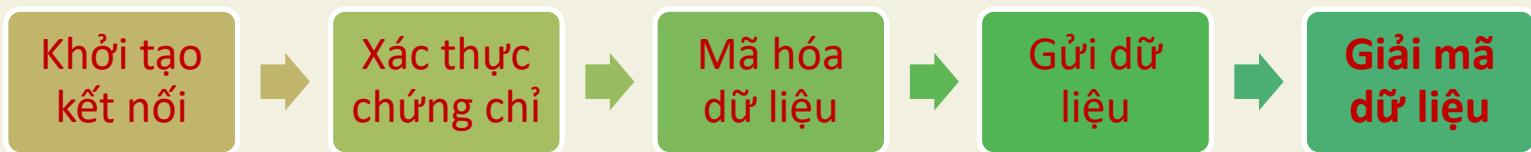
- Cơ chế hoạt động của TLS:



**Gửi dữ liệu:** Sau khi dữ liệu đã được mã hóa, nó có thể được gửi đi an toàn qua mạng.

Nếu có bất kỳ ai khác cố gắng truy cập dữ liệu này, họ sẽ không thể đọc được nội dung của nó vì nó đã được mã hóa bằng các khóa bí mật khác.

- Cơ chế hoạt động của TLS:



**Giải mã dữ liệu:** Khi dữ liệu đến tại người nhận, nó sẽ được giải mã bằng các khóa bí mật của người nhận để trở thành dữ liệu đọc được.

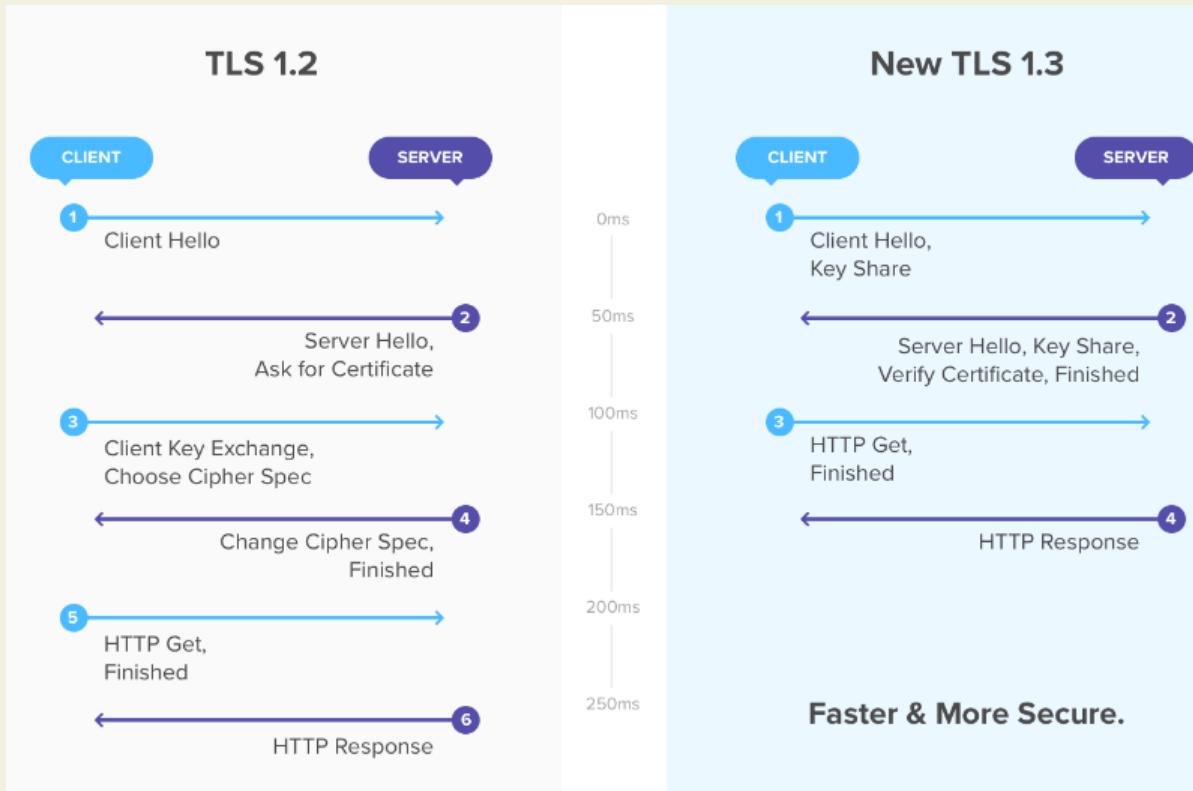
Sau đó, người nhận có thể sử dụng dữ liệu này để thực hiện các thao tác cần thiết.

Như vậy, giao thức TLS hoạt động bằng cách sử dụng private key và cơ chế xác thực để bảo vệ dữ liệu trong quá trình truyền đi trên mạng. Nó giúp ngăn chặn các tấn công mạng và uy tín của doanh nghiệp trong mắt khách hàng và đối tác.

- Các phiên bản của TLS:

- 1.TLS 1.0:** Phiên bản đầu tiên của TLS, ra mắt vào năm 1999. TLS 1.0 đã được nâng cấp nhiều lần để khắc phục các lỗ hổng bảo mật, nhưng vẫn còn được sử dụng trên một số hệ thống cũ hơn.
- 2.TLS 1.1:** Phiên bản thứ hai của TLS, ra mắt vào năm 2006. TLS 1.1 đã khắc phục một số lỗ hổng bảo mật của TLS 1.0 và có sự cải tiến về hiệu năng.
- 3.TLS 1.2:** Phiên bản thứ ba của TLS, ra mắt vào năm 2008. TLS 1.2 là phiên bản được sử dụng rộng rãi hiện nay và đã khắc phục nhiều lỗ hổng bảo mật của TLS 1.0 và 1.1.
- 4.TLS 1.3:** Phiên bản mới nhất của TLS, ra mắt vào năm 2018. TLS 1.3 được coi là phiên bản bảo mật nhất và có sự cải tiến về hiệu năng so với các phiên bản trước.

- Giao thức TLS:



- **Ứng dụng TLS:**

- Truyền dữ liệu trên mạng:** Giao thức TLS được sử dụng để bảo vệ các kết nối mạng trong quá trình truyền thông dữ liệu giữa hai máy tính hoặc hệ thống mạng khác nhau.
- Truy cập các trang web an toàn:** Giao thức TLS được sử dụng để bảo vệ các kết nối truy cập trang web qua giao thức HTTPS (Hypertext Transfer Protocol Secure). Khi người dùng truy cập vào một trang web qua HTTPS, dữ liệu của họ sẽ được mã hóa.
- Gửi và nhận email:** Giao thức TLS cũng được sử dụng để bảo vệ các kết nối gửi và nhận email qua giao thức SMTP (Simple Mail Transfer Protocol) và IMAP (Internet Mail Access Protocol). Khi người dùng gửi hoặc nhận email qua một máy chủ email an toàn, dữ liệu được mã hóa bằng TLS để bảo vệ khỏi các tấn công mạng.
- Truy cập các dịch vụ trực tuyến:** Giao thức TLS cũng được sử dụng để bảo vệ các kết nối truy cập các dịch vụ trực tuyến, như ngân hàng trực tuyến, bảo hiểm trực tuyến và y tế trực tuyến.

## Giới thiệu chung về HTTPS

- Hạn chế của HTTP:

- Không có cơ chế để người dùng kiểm tra tính tin cậy của Web server → lỗ hổng để kẻ tấn công giả mạo dịch vụ hoặc chèn mã độc vào trang web HTML
- Không có cơ chế mã mật → lỗ hổng để kẻ tấn công nghe lén đánh cắp thông tin nhạy cảm

- Secure HTTP: Kết hợp HTTP và SSL/TLS:

- Xác thực
- Bảo mật

## Tấn công vào HTTPS

- Tấn công sslstrip: lợi dụng lỗ hổng chuyển từ truy cập qua HTTP sang truy cập qua HTTPS (như thế nào?)



# Phần mềm mã độc và tường lửa

Report 2022,  
Bkav

## TOP 5 MÃ ĐỘC PHỔ BIẾN TẤN CÔNG HÀNG TRIỆU MÁY TÍNH TẠI VIỆT NAM



# Giới thiệu Malware

- ❑ **Malware (Malicious software)** hay còn gọi là mã độc (Malicious code) là tên gọi chung cho các phần mềm được thiết kế, lập trình đặc biệt để gây hại cho máy tính hoặc làm gián đoạn môi trường hoạt động mạng. Mã độc có thể được phát triển với các mục tiêu độc hại như đánh cắp thông tin cá nhân, gây hỏng hóc hệ thống, hoặc theo dõi hoạt động người dùng mà không được sự cho phép của họ.
- ❑ Mã độc thâm nhập vào một hệ thống máy tính mà không có sự đồng ý của nạn nhân.
- ❑ Mã độc hại còn được định nghĩa là “một chương trình (program) được chèn một cách bí mật vào hệ thống với mục đích làm tổn hại đến tính bí mật, tính toàn vẹn hoặc tính sẵn sàng của hệ thống”
- ❑ Mã độc mang ý nghĩa rộng hơn virus

# Lịch sử phát triển mã độc

Năm	Tên mã độc	Hành vi và thiệt hại
1971	Creeper Virus	Khi máy tính bị lây nhiễm sẽ hiển thị trên màn hình dòng chữ "The creeper". Mặc dù không gây ra thiệt hại gì nhưng Creeper đã dự báo về một tương lai mã độc có thể lây lan diện rộng. Reaper là phần mềm AV đầu tiên tạo ra để loại bỏ Creeper
1978	Animal	Đây là Trojan đầu tiên, không phá hủy hệ thống nhưng có thể tự lan truyền trên mạng.
1981	Elk Cloner	Lây truyền nhanh chóng trên các máy Apple II thông qua đĩa mềm và hiển thị một bài thơ ngắn để chế nhạo người dùng. Được viết bởi một cậu bé 15 tuổi.
1983	Virus	Thuật ngữ virus lần đầu tiên được sử dụng để mô tả một chương trình máy tính trong một cuốn tiểu thuyết của Frederick Cohen
1986	Brain	Được phát minh bởi 2 anh em người Pakistan (17 tuổi và 24 tuổi). Mục đích để trừ phạt và theo dõi những máy tính nào đã ăn cắp bản quyền phần mềm y tế viết cho máy tính IBM của họ.
1987	Jerusalem	Được thiết kế để phá hủy các tệp tin vào lõi lần xuất hiện của Thứ Sáu, ngày 13. Là một trong những virus phá hoại nhiều máy tính nhất (hoạt động 8 năm)

# Lịch sử phát triển mã độc

Năm	Tên mã độc	Hành vi và thiệt hại
1988	Morris Worm	Là sâu máy tính đầu tiên, gây ra tấn công DDOS đầu tiên. Lợi dụng các thiết bị kết nối Internet, liên tục gửi các tập tin và lưu lượng đến cùng một địa chỉ IP, gây quá tải, và sập và hoàn toàn mất kết nối (hàng nghìn máy tính bị lây nhiễm)
1992	Michelangelo	Hàng ngàn máy tính chạy MS-DOS ngừng hoạt động vì một loại virus tự động kích hoạt vào đúng ngày sinh nhật của nhà điêu khắc lừng danh Michelangelo - 6/3
1999	Happy99, Melissa, Kak	Những mã độc này lây lan rất nhanh thông qua môi trường Microsoft bởi người dùng Internet
2000	ILOVEYOU	Là một loại sâu đã lây nhiễm thông qua 1 email, sau 10 ngày phát hành thiệt hại 50tr máy tính Windows. Thiệt hại lên tới 10 tỷ \$ → Đại dịch virus toàn cầu.
2000	Yahoo.com	Thông qua tấn công DDOS, một cậu bé 15 tuổi người Canada đã đánh sập Yahoo.com
2001	Nimda	Lây lan qua email, bằng cách tìm kiếm các địa chỉ email trong những file .html, mặt khác chúng lấy địa chỉ email thông qua giả mạo bằng dịch vụ MAPI. Gây thiệt hại lên tới 530tr trong tuần đầu tiên.

# Lịch sử phát triển của mã độc

Năm	Tên mã độc	Hành vi và thiệt hại
2004	Santy	Là “webworm” đầu tiên, có khả năng lây lan mạnh, lây nhiễm vào các máy chủ Web đặt các diễn đàn trực tuyến viết bằng ngôn ngữ PHP, đồng thời SD goolge để tìm ra những máy chủ có lỗi hỏng để lây nhiễm.
2007		Tấn công DDoS có chủ ý, làm sập trang web của thủ tướng cũng như một số tổ chức do chính phủ điều hành như trường học và ngân hàng
2008	Conficker	Đã lây nhiễm khoảng 10 triệu hệ thống máy chủ của Microsoft, bao gồm cả máy chính phủ và quân đội → An ninh mạng trong ý thức cộng đồng tăng lên
2010	Stuxnet	Nhắm vào các cơ sở hạt nhân của Iran. Được coi là dạng phần mềm độc hại tiên tiến nhất từng tạo ra. Khoảng 60 nghìn máy tính bị nhiễm trong đó 60% là ở Iran. Khoảng 5000 máy ly tâm của Iran đã “hóa điên” và kéo lùi dự án hạt nhân của Iran trong 2 năm.
2017	WannaCry	Tội phạm mạng đã sử dụng chính những tool của NSA để phát tán và lây lan mã độc. Nó mã hóa tập tin và người dùng phải trả 1 khoản tiền (BTC ~ 300usd) cho hacker. Gây ảnh hưởng hàng triệu users, 150 quốc gia, hơn 200 nhignf hệ thống mạng bị ảnh hưởng trong đó có VN.

## Malware

# Xu hướng phát triển

- **1950s - 1970s: Ngày đầu của máy tính và virus đầu tiên**

Những năm đầu của máy tính chứng kiến sự xuất hiện của virus đầu tiên, như Creeper và Reaper. Chúng không gây hại nghiêm trọng và thường chỉ được sử dụng để thử nghiệm.

- **1980s: Sự gia tăng của virus máy tính**

Trong thập kỷ 1980, virus máy tính bắt đầu trở nên phổ biến hơn và gây ra các vấn đề bảo mật nghiêm trọng. Trong giai đoạn này, Michelangelo là một trong những virus nổi tiếng đầu tiên.

# Malware

## Xu hướng phát triển

- **1990s: Thời kỳ của Trojan và Worms**
  - Các thập kỷ này chứng kiến sự gia tăng đáng kể về Trojan và Worms. Worm nổi tiếng nhất trong giai đoạn này là "ILOVEYOU", gây thiệt hại 50tr máy tính và thiệt hại 10 tỷ \$
- **2000s: Sự gia tăng của Ransomware và Botnets**
  - Ransomware và botnets trở nên phổ biến vào thập kỷ này, với các ví dụ như CryptoLocker và Conficker.



# Malware

## Xu hướng phát triển

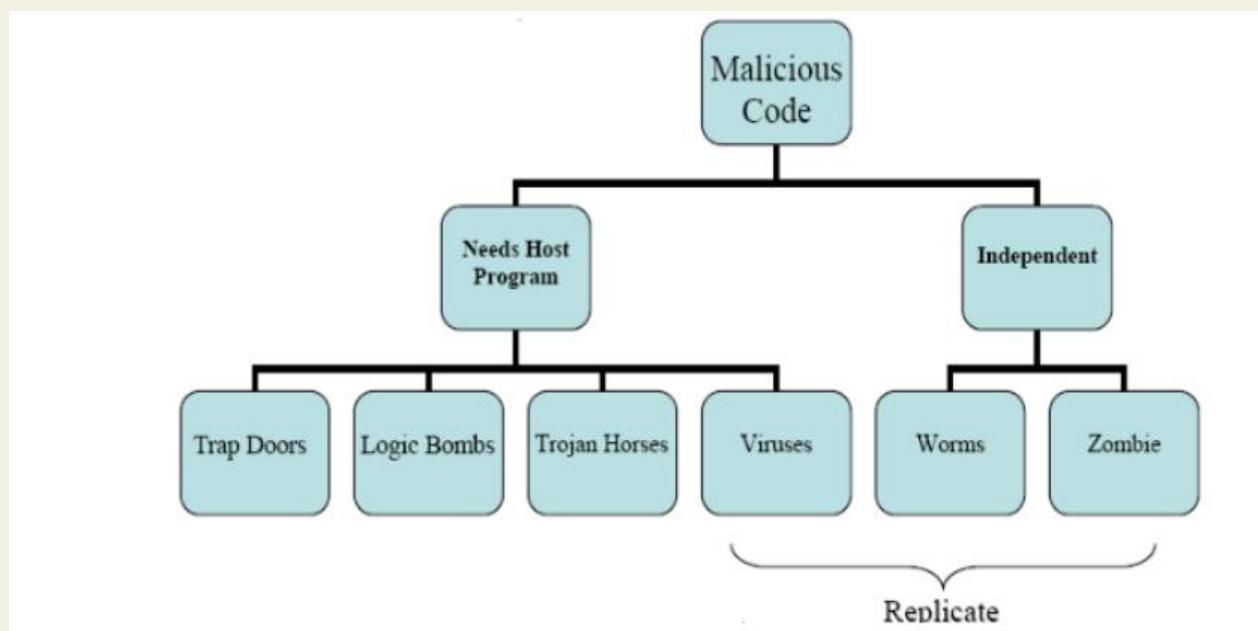
- **2010s: Phát triển của APT và tấn công nâng cao**
  - Advanced Persistent Threats (APT) và các hình thức tấn công cao cấp đánh dấu giai đoạn này, với các vụ tấn công nổi tiếng như Stuxnet và các cuộc tấn công liên quan đến tiềm năng quốc gia.
- **Hiện tại (2020s): Sự tăng cường của AI và IoT Malware**
  - Các công nghệ mới như trí tuệ nhân tạo (AI) và Internet of Things (IoT) đã mở ra những cơ hội mới cho malware và các hình thức tấn công phức tạp.



# Malware

## Phân loại mã độc

- Theo hình thức lây nhiễm



## Malware

# Phân loại mã độc

- **Phân loại của NIST**
- **Virus** : là một loại mã độc hại (Malicious code) có khả năng tự nhân bản và lây nhiễm chính nó vào các file, chương trình hoặc máy tính
- **Worm**: là một chương trình có khả năng tự nhân bản và tự lây nhiễm trong hệ thống tuy nhiên nó có khả năng “tự đóng gói”, điều đó có nghĩa là Worm không cần phải có “file chủ” để mang nó khi nhiễm vào hệ thống
- **Trojan Horse**: Là loại mã độc hại được đặt theo sự tích “Ngựa thành Troy”. Trojan horse không tự nhân bản tuy nhiên nó lây vào hệ thống với biểu hiện rất ôn hoà nhưng thực chất bên trong có ẩn chứa các đoạn mã với mục đích gây hại

•

# Malware

## Phân loại mã độc

- **Malicious Mobile Code:** Là một dạng mã phần mềm có thể được gửi từ xa vào để chạy trên một hệ thống mà không cần đến lời gọi thực hiện của người dùng hệ thống đó.

Malicious Mobile Code được coi là khác với virus, worm ở đặc tính là nó không nhiễm vào file và không tìm cách tự phát tán

- **Tracking Cookie:** Là một dạng lạm dụng cookie để theo dõi một số hành động duyệt web của người sử dụng một cách bất hợp pháp
- **Phần mềm gián điệp (Spyware):** Là loại phần mềm chuyên thu thập các thông tin từ các máy chủ (thông thường vì mục đích thương mại) qua mạng Internet mà không có sự nhận biết và cho phép của chủ máy



# Malware

## Phân loại mã độc

- **Phần mềm quảng cáo (Adware)**: rất hay có ở trong các chương trình cài đặt tải từ trên mạng. Một số phần mềm vô hại, nhưng một số có khả năng hiển thị thông tin lên màn hình, cưỡng chế người sử dụng.
- **Attacker Tool**: Là những bộ công cụ tấn công có thể sử dụng để đẩy các phần mềm độc hại vào trong hệ thống. Các bộ công cụ này có khả năng giúp cho kẻ tấn công có thể truy nhập bất hợp pháp vào hệ thống hoặc làm cho hệ thống bị lây nhiễm mã độc hại
- **Phishing**: Là một hình thức tấn công thường có thể xem là kết hợp với mã độc hại. Phishing là phương thức dụ người dùng kết nối và sử dụng một hệ thống máy tính giả mạo nhằm làm cho người dùng tiết lộ các thông tin bí mật về danh tính (ví dụ như mật khẩu, số tài khoản, thông tin cá nhân...)

•

# Malware

## Phân loại mã độc

- **Virus Hoax:** Là các cảnh báo giả về virus. Các cảnh báo giả này thường nấp dưới dạng một yêu cầu khẩn cấp để bảo vệ hệ thống. Mục tiêu của cảnh báo virus giả là cố gắng lôi kéo mọi người gửi cảnh báo càng nhiều càng tốt qua email

# Một số loại mã độc tiêu biểu

## Mustang panda

- Mustang panda là nhóm tin tặc hoạt động từ năm 2014, có trụ sở hoạt động tại Trung Quốc, thường sử dụng các dòng mã độc PlugX, Cobalt Strike. Là nhóm tấn công APT có độ động cơ cao, được hậu thuẫn, kĩ thuật tinh vi để lây nhiễm và cài cắm mã độc với mục tiêu có được quyền truy cập vào máy nạn nhân để từ đó thực hiện hoạt động gián điệp và đánh cắp thông tin.
- Quốc gia mục tiêu: Úc, Bangladesh, Bỉ, Trung Quốc, Ethiopia, Đức, Hồng Kông, Ấn Độ, Mông Cổ, Myanmar, Nepal, Pakistan, Singapore, Hàn Quốc, Đài Loan, Anh, Mỹ, Việt Nam và LHQ.
- Các lĩnh vực mục tiêu: Hàng không, Chính phủ, Tổ chức phi chính phủ, Viễn thông

# Một số loại mã độc tiêu biểu

## Mustang panda

### • Các cuộc tấn công

TT	Thời gian	Chiến dịch
1	8/2019	Anomanil Threat Research Team đã phát hiện ra các file .lnk đáng ngờ trong quá trình thu thập thông tin tình báo thường xuyên. Với phương pháp spear phishing có liên quan đến nhóm Mustang Panda.
2	01/2020	Các nhà bảo mật của hãng Avira phát hiện một phiên bản PlugX mới từ Mustang Panda APT, được sử dụng để do thám mục tiêu tại Hồng Kông và Việt Nam.
3	03/2020	Công ty an ninh mạng Việt Nam VinCSS phát hiện nhóm Mustang Panda phát tán email có đính kèm file .rar với mục đích giả mạo chỉ thị của Thủ tướng Việt Nam về đợt bùng phát Covid
4	03/2020	ATR xác định nhóm Mustang Panda đang sử dụng các lừa đảo theo chủ đề liên quan đến Corona virus.
5	03/2020	Anomali – Mustang tiến hành chiến dịch lợi dụng Covid-19 để tấn công vào Đài Loan

# Một số loại mã độc tiêu biểu

## Mustang panda

6	05/2020	Lab52 - Chiến dịch Dll-Sideload trojan với máy chủ C&C tạm thời
7	09/2020	Proofpoint – Chiến dịch tấn công Vaticant và các nhóm ngoại giao mục tiêu sử dụng biến thể viết bằng Golang mới của loader PlugX
8	03/2021	McAfee – Chiến dịch tấn công vào các hãng viễn thông nhằm đánh cắp bí mật 5G.
9	09/2021	Insikt Group – phát hiện Mustang tấn công vào các cơ quan chính phủ Indonesia.
10	02/2022	TalosIntelligence – Phát hiện chiến dịch tấn công vào các nước Châu Âu và Nga.

# Một số loại mã độc tiêu biểu

## Mã độc WannaCry

- Còn được biết đến với các tên khác như: WannaCrypt, WannaCrypt0r, WCRY là loại mã độc đặc biệt nguy hiểm, Ngoài việc mã hóa dữ liệu người dùng, còn lợi dụng lỗ hổng trên hệ điều hành windows để lây nhiễm cho các thiết bị khác đồng thời cài đặt cửa hậu trên máy tính bị nhiễm
- WannaCry đã tấn công hơn 512000 thiết bị trên toàn thế giới. Nạn nhân sẽ phải trả một khoản tiền chuộc ít nhất 300 \$ (qua bitcoin) để lấy lại dữ liệu của mình, số tiền này sẽ tăng lên gấp đôi sau thời hạn 3 ngày, và sẽ bị mất sau 7 ngày nếu người dùng không thanh toán
- Chỉ sau 2 ngày được phát hiện, gây ảnh hưởng 10.000 tổ chức, 200.000 cá nhân trên 150 quốc gia.
- Được coi là mã độc nguy hiểm nhất thế giới,



Xuất hiện vào tháng 05/2017

# Malware

## Các kĩ thuật lây nhiễm và phá hoại trong Malware

- **Lây lan qua USB:** tạo ra một tệp autorun.inf trong thư mục gốc của USB. Khi phát hiện có thiết bị lưu trữ mới được cắm vào (USB, CD, Floppy Disk... ), Window mặc nhiên sẽ kiểm tra tệp autorun.inf nằm trong đó, nếu có nó sẽ tự động thực hiện các dòng lệnh theo cấu trúc được sắp xếp trước
- **Lây lan qua Yahoo!Messenger:** tin nhắn rất hấp dẫn của bạn bè gửi cho và sau đó là đường link đến một trang web lạ, Virus đã được tự động down về máy và kích hoạt
- **Lây lan qua trình duyệt truy cập web:** thông qua đường link (một trang web bị nhiễm mã độc (dạng VBScript
- **Lây lan qua Email, Outlook Express:** "giả dạng" một e mail với một địa chỉ bất kì nào có nội dung là một tấm thiệp, một file attach hay đường link có chứa file malware gây nguy hiểm cho máy tính

# Malware

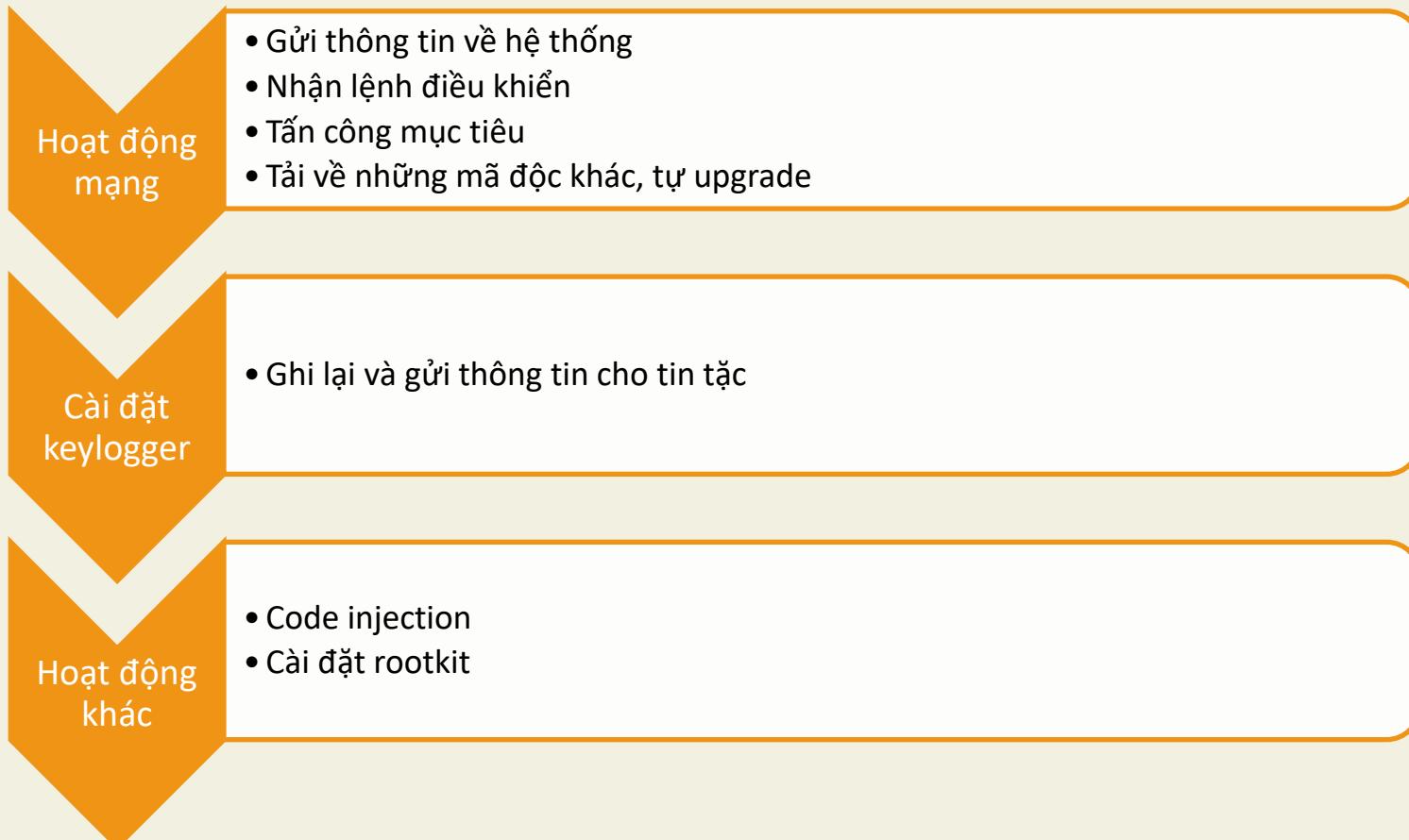
## Các kĩ thuật lây nhiễm và phá hoại trong Malware

- **Lây lan vào các tệp tin thực thi:** các file thực thi EXE, phổ biến cho các hệ thống Windows
- **Chia sẻ file**
- **Lỗ hổng của hệ điều hành hoặc ứng dụng**

# Ảnh hưởng và tác hại của mã độc

- Hệ thống file**
  - Nhân bản mã độc, lây nhiễm vào file
  - Giấu trong những thư mục hệ thống
  - Thiết lập thuộc tính ẩn, hệ thống để gây khó khăn trong việc tìm diệt
  - Ẩn trong hệ thống file NTFS
- Registry**
  - Tạo những khóa mới để lưu giữ thông tin
  - Thay đổi những giá trị của hệ thống
  - Đọc những thuộc tính thông tin quan trọng
  - Thường dùng để thiết lập chạy khi khởi động
- Tiến trình**
  - Khởi tạo một tiến trình mới
  - Khởi tạo một dịch vụ mới hoặc thay thế một dịch vụ đang hoạt động Malware DLL
  - Kiểm tra những tiến trình đang chạy để tránh bị phát hiện
  - Tự gắn vào những tiến trình hệ thống để tránh bị gỡ bỏ

# Ảnh hưởng và tác hại của mã độc



# Các phương pháp phát hiện mã độc

- Có 3 kỹ thuật nhận dạng đã được áp dụng:
  - dựa vào chuỗi nhận dạng (signaturebased approach),
  - dựa vào hành vi nghi ngờ (suspicious behavior-based approach)
  - dựa vào ý định (intention-based approach)

# Malware

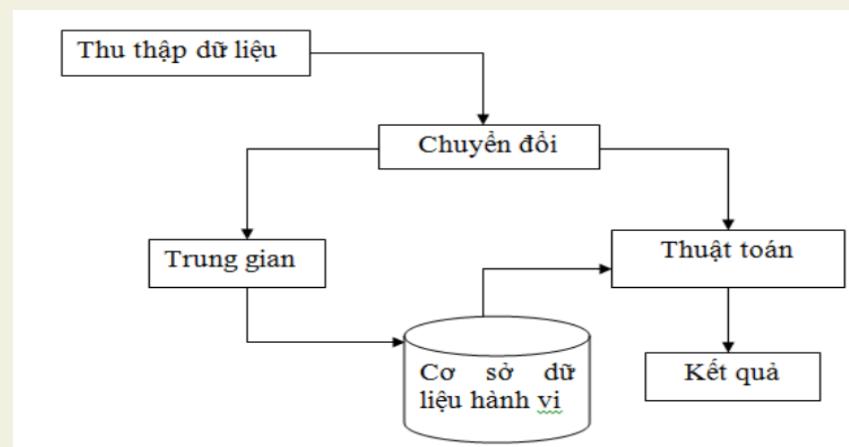
## Các phương pháp phát hiện mã độc

- *Phương pháp phát hiện dựa vào chuỗi nhận dạng:*
- Hoạt động theo nguyên lý nhận dạng mẫu, các AV sử dụng một CSDL chứa mẫu virus (ID-virus library). Mỗi khi có virus mới, các chuyên gia anti-virus sẽ giải mã, trích chọn và cập nhật chuỗi nhận dạng virus vào thư viện. Thông tin về đối tượng chẩn đoán (ghi nhận từ hệ thống đích) cùng với thông tin của virus (trong thư viện mẫu) sẽ cho kết luận về tình trạng của đối tượng
- Nhận dạng mẫu giúp AV phát hiện các virus đã biết trên tập dữ liệu chẩn đoán với độ chính xác cao
- Nhược điểm: Công kênh, Bị động, Nhầm lẫn

# Malware

## Các phương pháp phát hiện mã độc

- *Phương pháp phát hiện dựa trên hành vi*
- xác định các hành động thực hiện của mã độc hơn là việc xác định cấu trúc nhị phân của chương trình. Các chương trình không giống với cú pháp hay cấu trúc nhưng có hành vi giống với những hành vi đã xác định trước là đã xác định được nó là mã độc hay không



# Các phương pháp phát hiện mã độc Malware

- *Phát hiện virus dựa vào ý định :*
- Những thay đổi quan trọng trong tập tin, cấu hình hệ thống hay HĐH đều được cảnh báo như một mối hiểm họa tiềm tàng
- Mặc dù đơn giản nhưng tiếp cận này tỏ ra khá hiệu quả vì nó có thể bảo vệ máy tính khỏi các mối đe dọa chưa được biết đến, kể cả virus máy tính

# Phân tích mã độc

## • Phân tích mã độc để làm gì?

Phân tích mã độc là thực hiện các biện pháp nghiệp vụ để thu thập, tìm hiểu, nghiên cứu mọi thông tin về mã độc

- ✓ Hành vi của mã độc
- ✓ Ảnh hưởng, tác hại



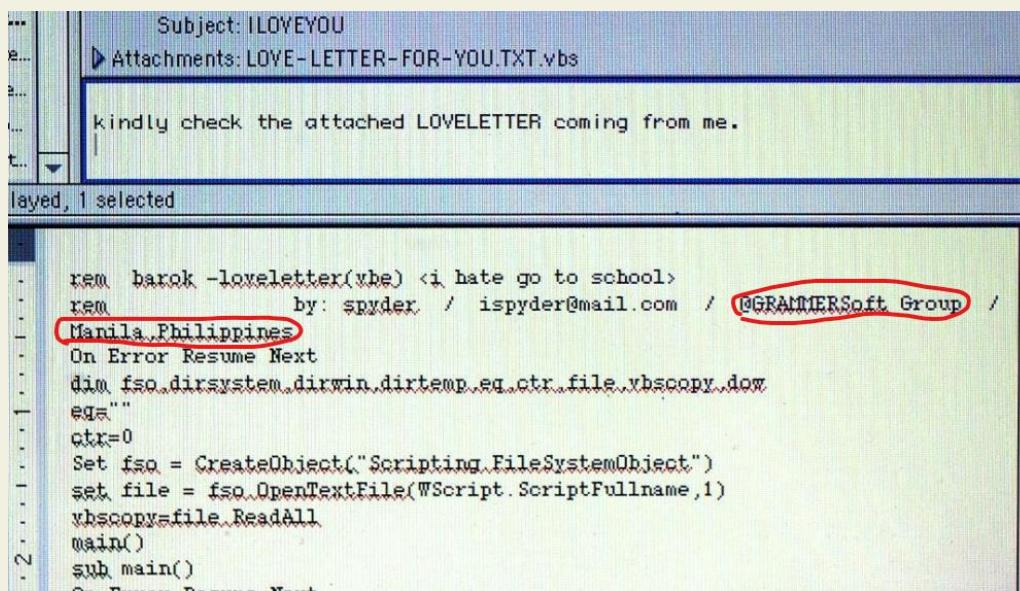
- Cảnh báo
- Biện pháp phòng chống, ngăn chặn

- ✓ Lây lan của mã độc
- ✓ Lỗ hổng bị khai thác

## Phân tích mã độc

- Phân tích mã độc để làm gì?

# Ví dụ: ILOVEYOU



## • Phương pháp phân tích mã độc

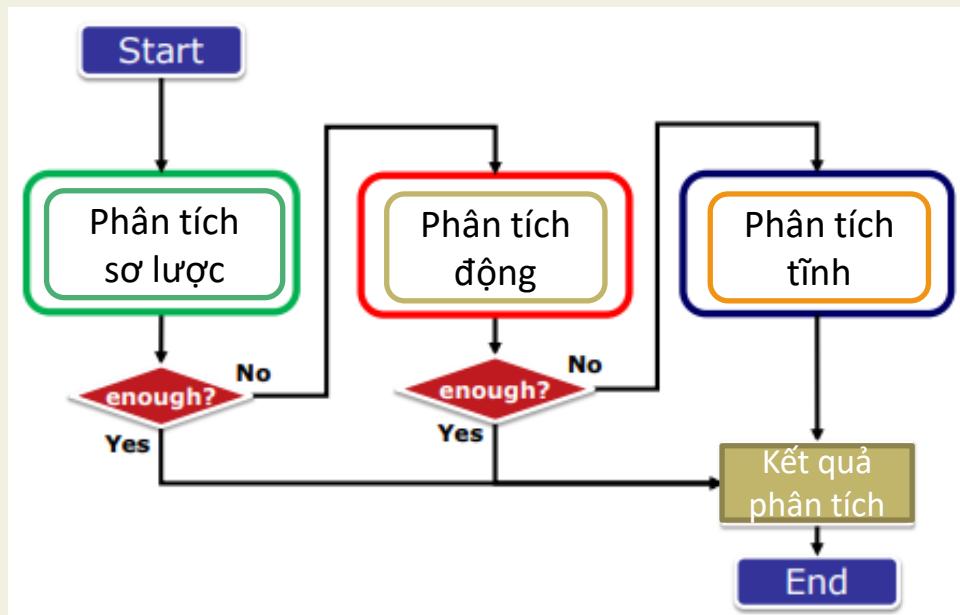


# Phân tích mã độc

	<b>Phân tích sơ lược</b>	<b>Phân tích động</b>	<b>Phân tích tinh</b>
<b>Tổng quan</b>	Thu thập thông tin từ đối tượng bị tấn công mà không cần thực thi mã độc	Thực thi mã độc và giám sát hành vi	Đọc code – dịch ngược và hiểu các chức năng của mã độc
<b>Đầu ra</b>	<ul style="list-style-type: none"> <li>- Hàm băm</li> <li>- Strings</li> <li>- Các thuộc tính file</li> <li>- Thông tin đóng gói</li> <li>- Thông tin từ các AV</li> </ul>	<p>Hoạt động:</p> <ul style="list-style-type: none"> <li>- File system</li> <li>- Registry</li> <li>- Process</li> <li>- Network</li> </ul>	<p>Các chức năng của mã độc.</p> <p>Ví dụ:</p> <ul style="list-style-type: none"> <li>- Bot commands</li> <li>- Encode/Decode methods</li> </ul>
<b>Nguy cơ an toàn</b>	Thấp	Cao	Trung bình
<b>Hiệu quả phân tích</b>	Thấp	Trung bình	Cao

# Giới thiệu về phân tích mã độc

## • Sơ đồ các bước phân tích mã độc



# Giới thiệu về phân tích mã độc

## Một số điểm lưu ý khi phân tích mã độc

### ❖ Thận trọng khi phân tích

- *Một sai lầm có thể dẫn tới hậu quả nghiêm trọng*

### ❖ Cân nhắc khi công bố kết quả

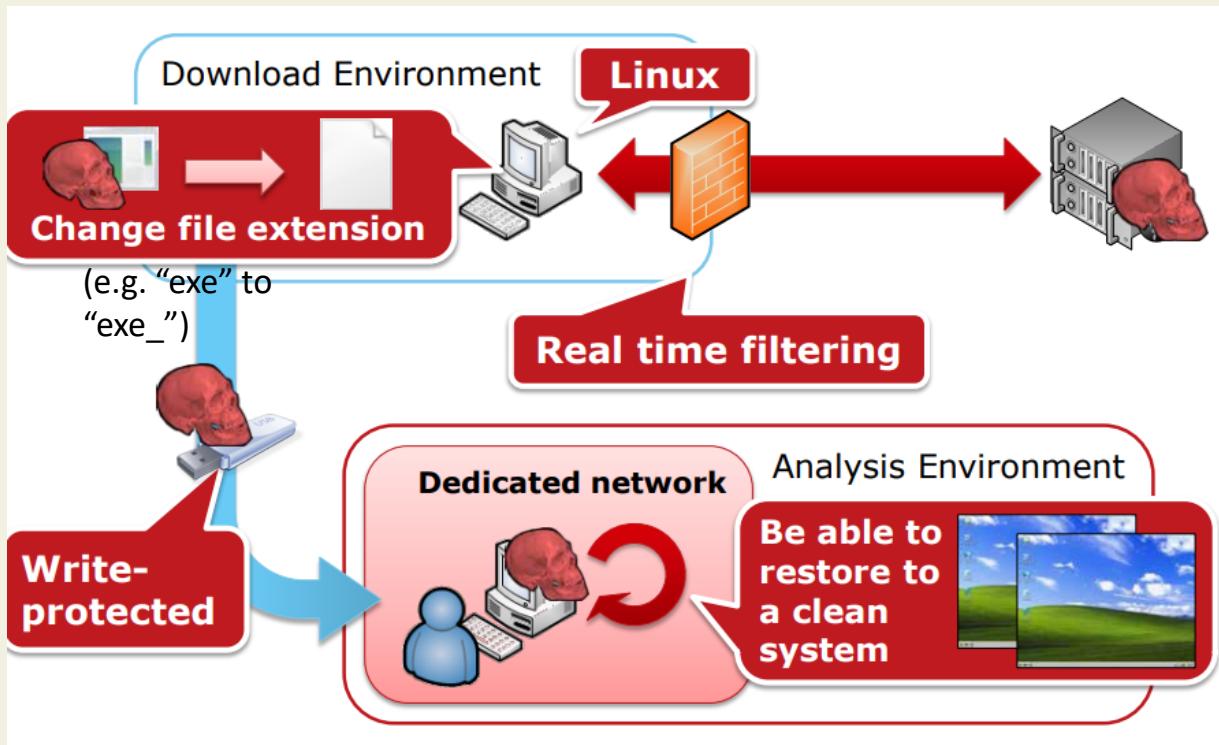
*Kết quả phân tích chi tiết được công bố có thể dẫn tới  
xuất hiện những mã độc mạnh hơn*

### ❖ Xây dựng môi trường phân tích an toàn

- *Chú ý tới môi trường để tải về mã độc, phân tích  
mã độc và kết quả phân tích*

# Phân tích mã độc

- Môi trường phân tích an toàn

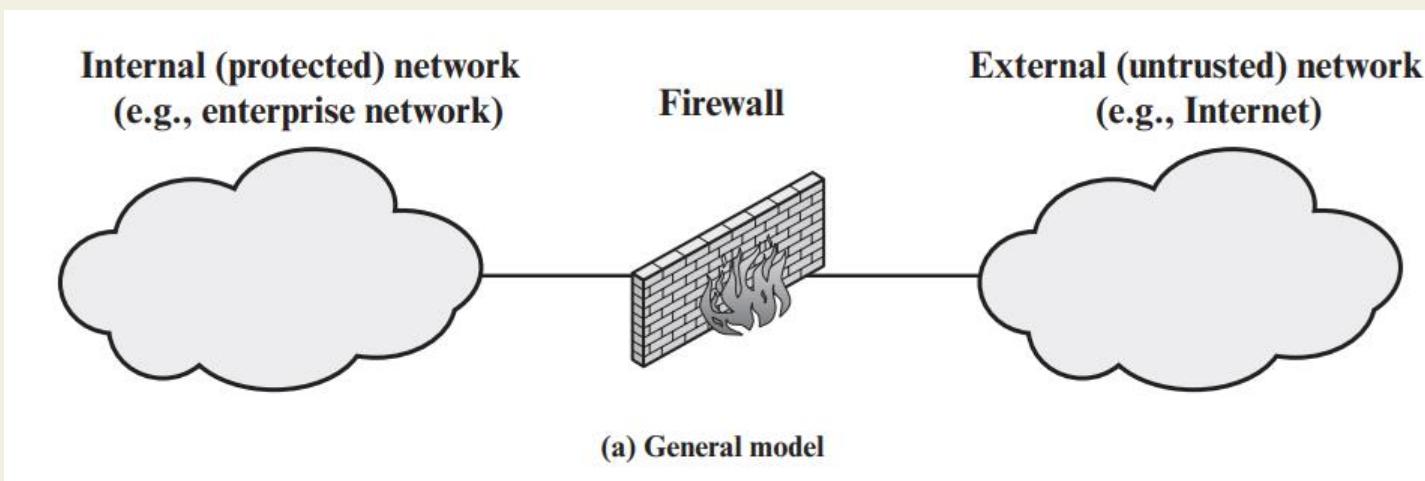


# Câu hỏi:

- Câu 1: Tìm hiểu ít nhất 5 loại mã độc nguy hiểm trên thế giới? (Tên mã độc, tổ chức/cá nhân phát tán/tạo, mức thiệt hại bao nhiêu, số lượng máy tính, quốc gia bị nhiễm, cơ chế lây nhiễm, ...)
- Câu 2: Kể tên các tổ chức, doanh nghiệp tại Việt Nam có job về phân tích mã độc?

# Tường lửa - firewall

- **Khái niệm:** Firewall là một công cụ phần cứng hoặc phần mềm hoặc cả 2 được tích hợp vào hệ thống để chống lại sự truy cập trái phép, ngăn chặn virus... để đảm bảo nguồn thông tin nội bộ được an toàn, tránh bị kẻ gian đánh cắp thông tin.
- Nói ngắn gọn và dễ hiểu hơn thì Firewall chính là ranh giới bảo mật giữa bên trong và bên ngoài của một hệ thống mạng máy tính.



# Tường lửa - firewall

- **Vai trò:** Firewall giúp kiểm soát luồng thông tin giữa Intranet và Internet, chúng phát hiện và phán xét những hành vi được truy cập và không được truy cập vào bên trong hệ thống, đảm bảo tối đa sự an toàn thông tin.

Tính năng chính của dòng thiết bị này có thể được tóm tắt ở những gạch đầu dòng dưới đây:

- Cho phép hoặc vô hiệu hóa các dịch vụ truy cập ra bên ngoài, đảm bảo thông tin chỉ có trong mạng nội bộ.
- Cho phép hoặc vô hiệu hóa các dịch vụ bên ngoài truy cập vào trong.
- Phát hiện và ngăn chặn các cuộc tấn công từ bên ngoài.
- Hỗ trợ kiểm soát địa chỉ truy cập (bạn có thể đặt lệnh cấm hoặc là cho phép).
- Kiểm soát truy cập của người dùng.
- Quản lý và kiểm soát luồng dữ liệu trên mạng.

# Tường lửa - firewall

- **Vai trò:**

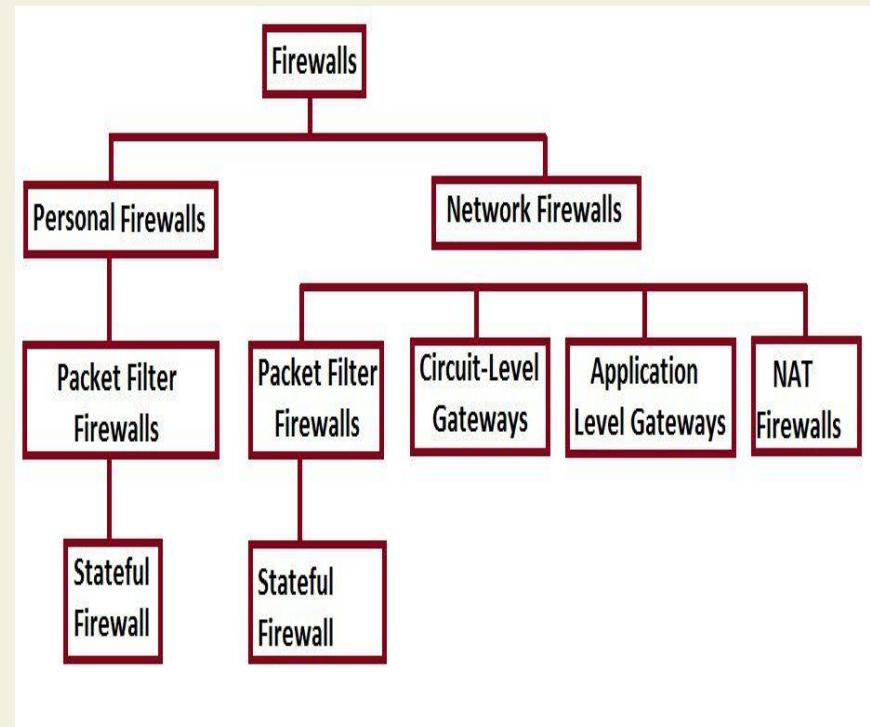
- Xác thực quyền truy cập.
- Hỗ trợ kiểm soát nội dung thông tin và gói tin lưu chuyển trên hệ thống mạng.
- Lọc các gói tin dựa vào địa chỉ nguồn, địa chỉ đích và số Port ( hay còn công), giao thức mạng.
- Người quản trị có thể biết được kẻ nào đang cố gắng để truy cập vào hệ thống mạng.
- Firewall hoạt động như một Proxy trung gian.
- Bảo vệ tài nguyên của hệ thống bởi các mối đe dọa bảo mật.
- Cân bằng tải: Bạn có thể sử dụng nhiều đường truyền internet cùng một lúc, việc chia tải sẽ giúp đường truyền internet ổn định hơn rất nhiều.

# Tường lửa - firewall

- Phân loại

**Personal Firewall:** Loại này được thiết kế để bảo vệ một máy tính trước sự truy cập trái phép từ bên ngoài.

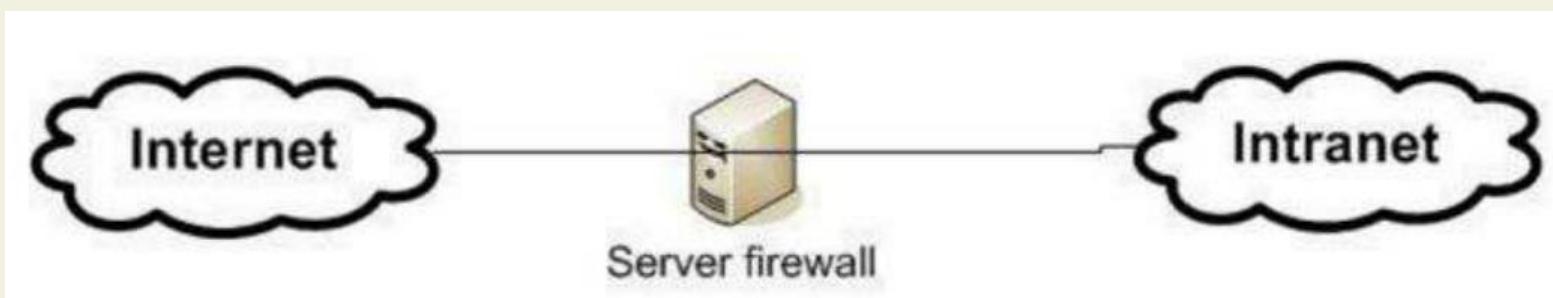
**Network Firewalls:** Được thiết kế ra để bảo vệ các host trong mạng trước sự tấn công từ bên ngoài.



# Tường lửa - firewall

Sản phẩm Firewall được ứng dụng trong thực tế

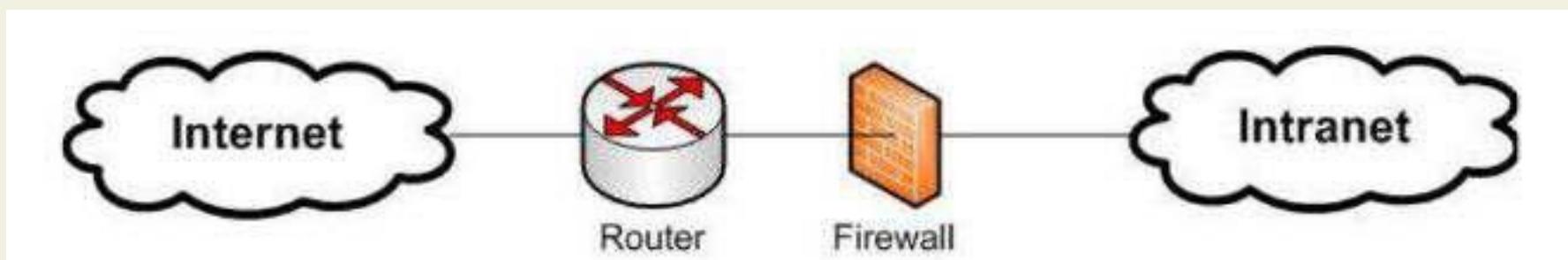
- **Software Firewalls:** Hay còn gọi là Firewall mềm, đây là loại Firewall được tích hợp trên hệ điều hành, nó bao gồm các sản phẩm như: SunScreen firewall, Check Point NG, IPF, Linux's IPTables, Microsoft ISA server ...



# Tường lửa - firewall

Sản phẩm Firewall được ứng dụng trong thực tế

- **Appliance Firewalls:** Hay còn gọi là Firewall cứng. Đây là loại Firewall cứng được tích hợp sẵn trên các phần cứng chuyên dụng, thiết kế này dành riêng cho Firewall. Một số Firewall cứng như Cisco PIX, WatchGuard Fireboxes, NetScreen firewall, SonicWall Appliances, Nokia firewall...



Ngoài ra còn có **Integrated firewalls**: Hay còn gọi là Firewall tích hợp. Ngoài chức năng cơ bản của Firewall ra thì nó còn đảm nhận các chức năng khác ví dụ như VPN, phát hiện và chống xâm nhập từ bên ngoài, lọc thư rác, chống lại virus...

# Ôn tập