An toàn và bảo mật thông tin

Giáo viên: TS. Lê Thị Anh

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.

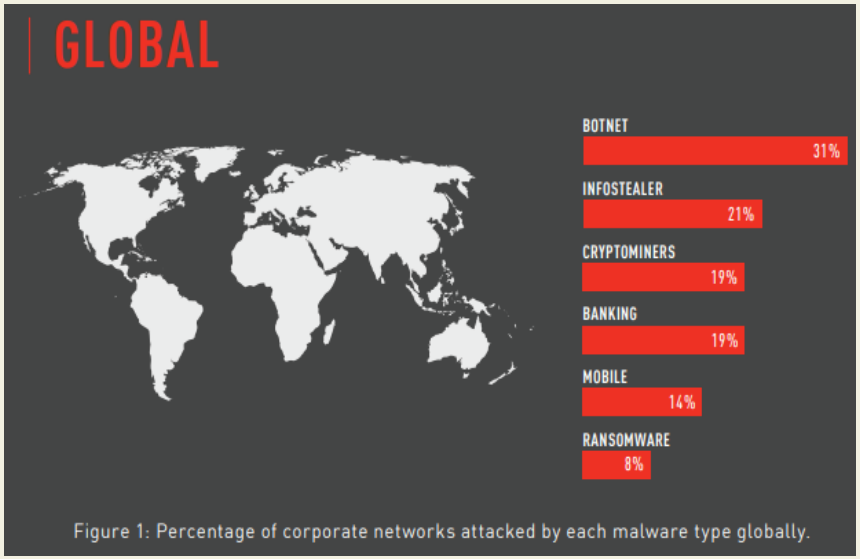


Figure 1: Percentage of corporate networks attacked by each malware type globally.

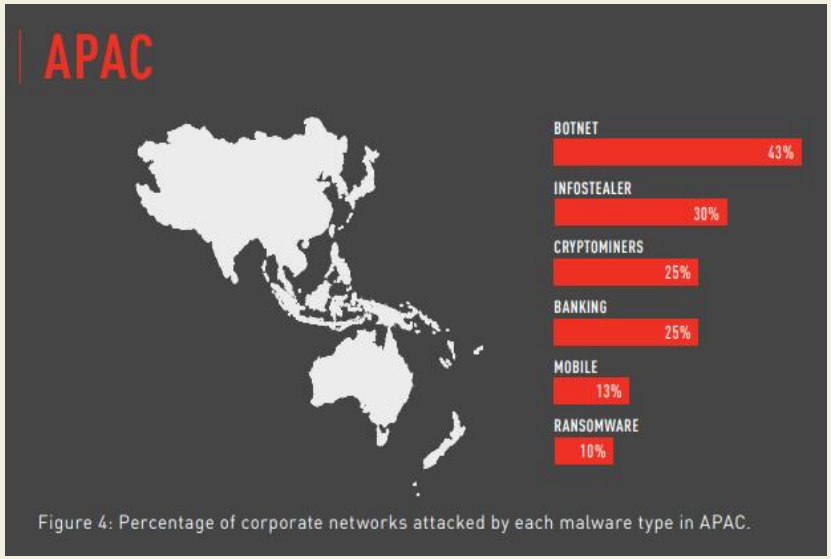


Figure 4: Percentage of corporate networks attacked by each malware type in APAC.

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.

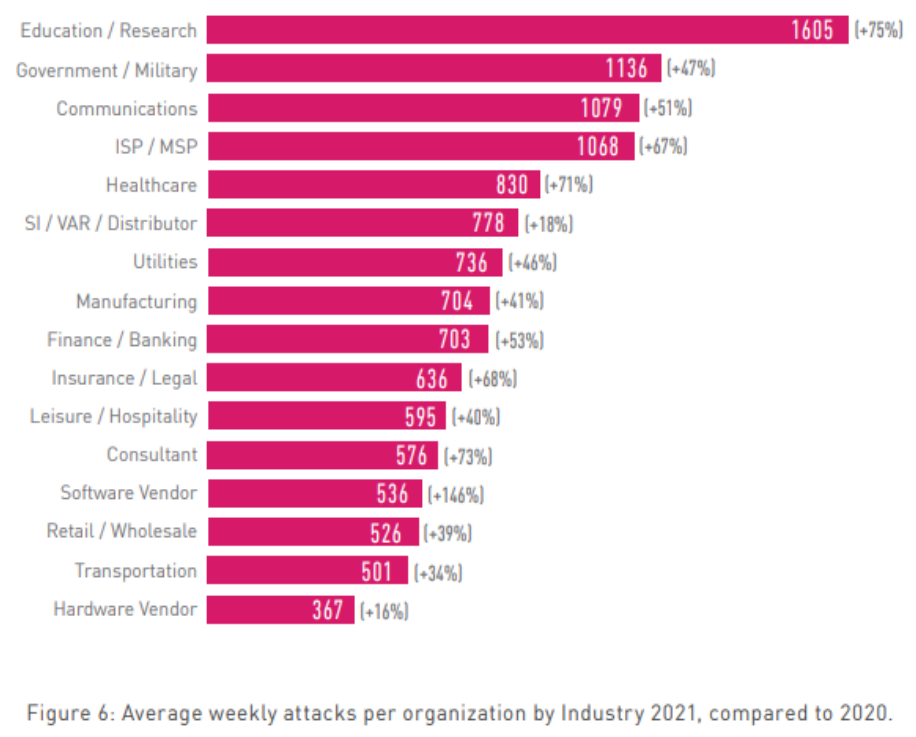
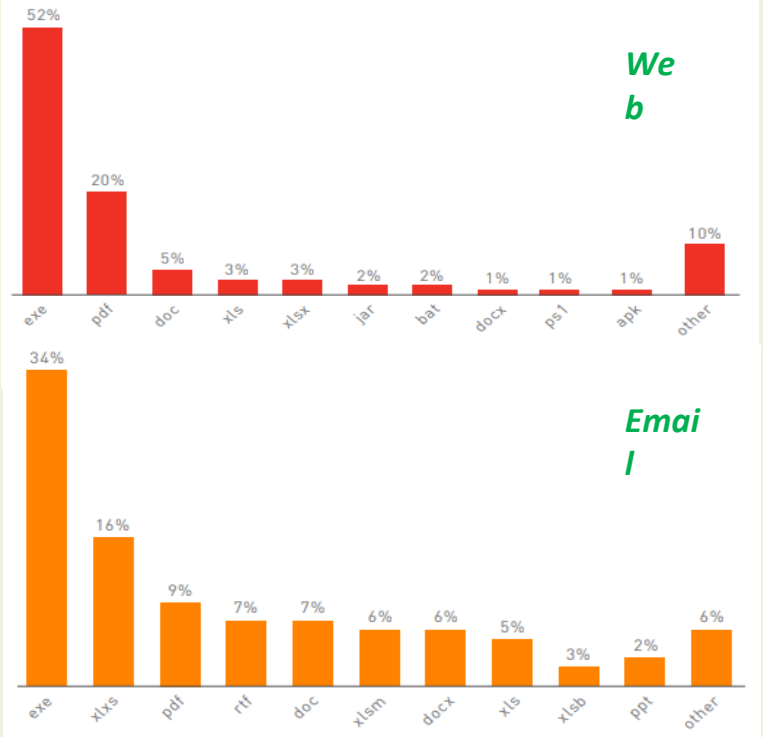


Figure 6: Average weekly attacks per organization by Industry 2021, compared to 2020.



Top malicious file types

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 tư, 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 (toàn 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 Checkpoin từ tháng 1 đến tháng 12 năm 2021 của hãng: https://threatmap.checkpoint.com/

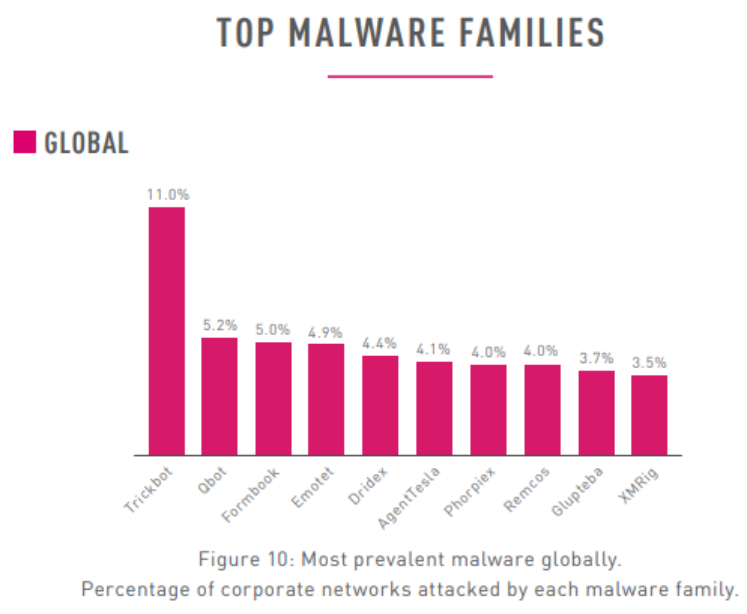


Figure 10: Most prevalent malware globally.
Percentage of corporate networks attacked by each malware family.

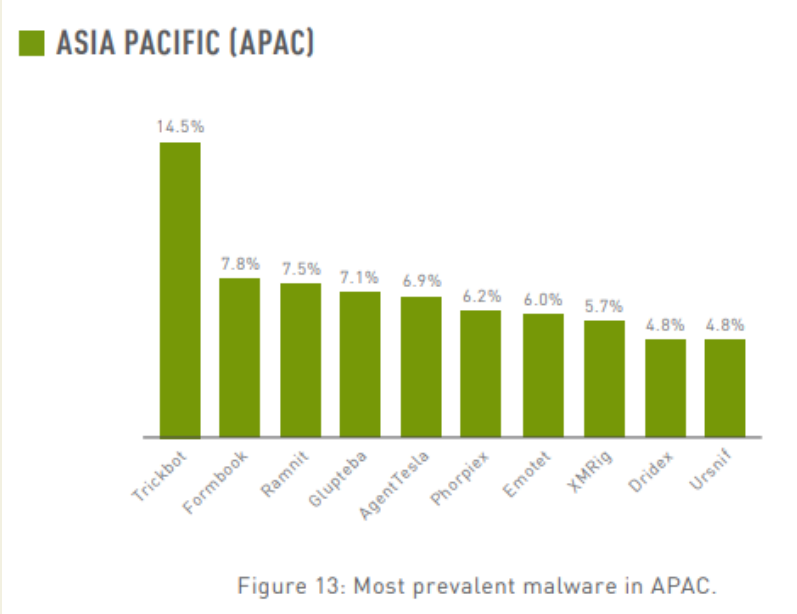


Figure 13: Most prevalent malware in APAC.

1. Tại sao phải bảo vệ thông tin

- ✓Thông tin là một bộ phần 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 doạ 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 ngơời 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 doạ 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 nhơ spyware, adware …) chạy trên hệ thống.

3. Các mối đe doạ 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

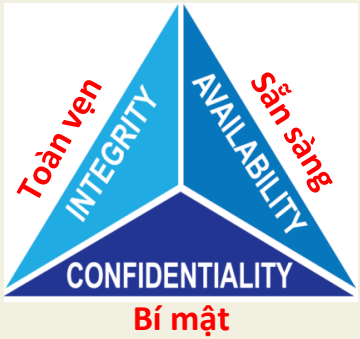
- ***Computer Security*:** The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).

- ***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 vaf hệ thống. Gồm có toàn vẹn về dữ liệu và toàn vẹn của hệ thống:

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

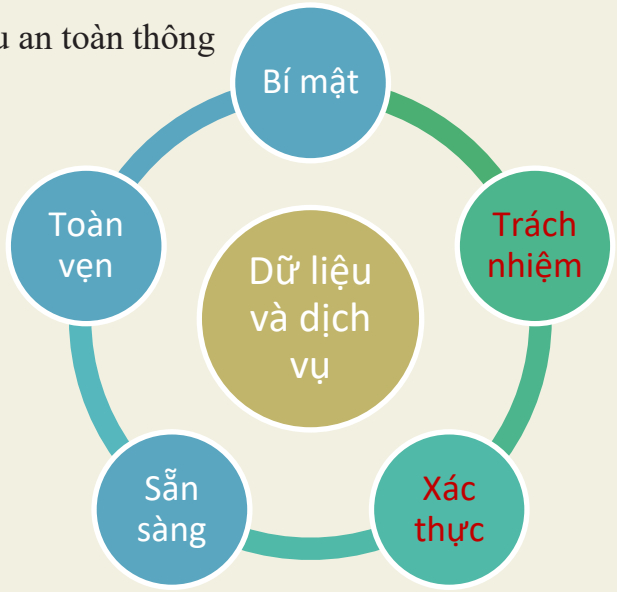
→ Tính toàn vẹn của hệ thống: Đảm bảo rằng một hệ thống thực hiện chức năng dự kiến của nó một cách nguyên vẹn, không bị thao túng trái phép một cách có chủ ý hoặc vô ý.

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 đoán 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:

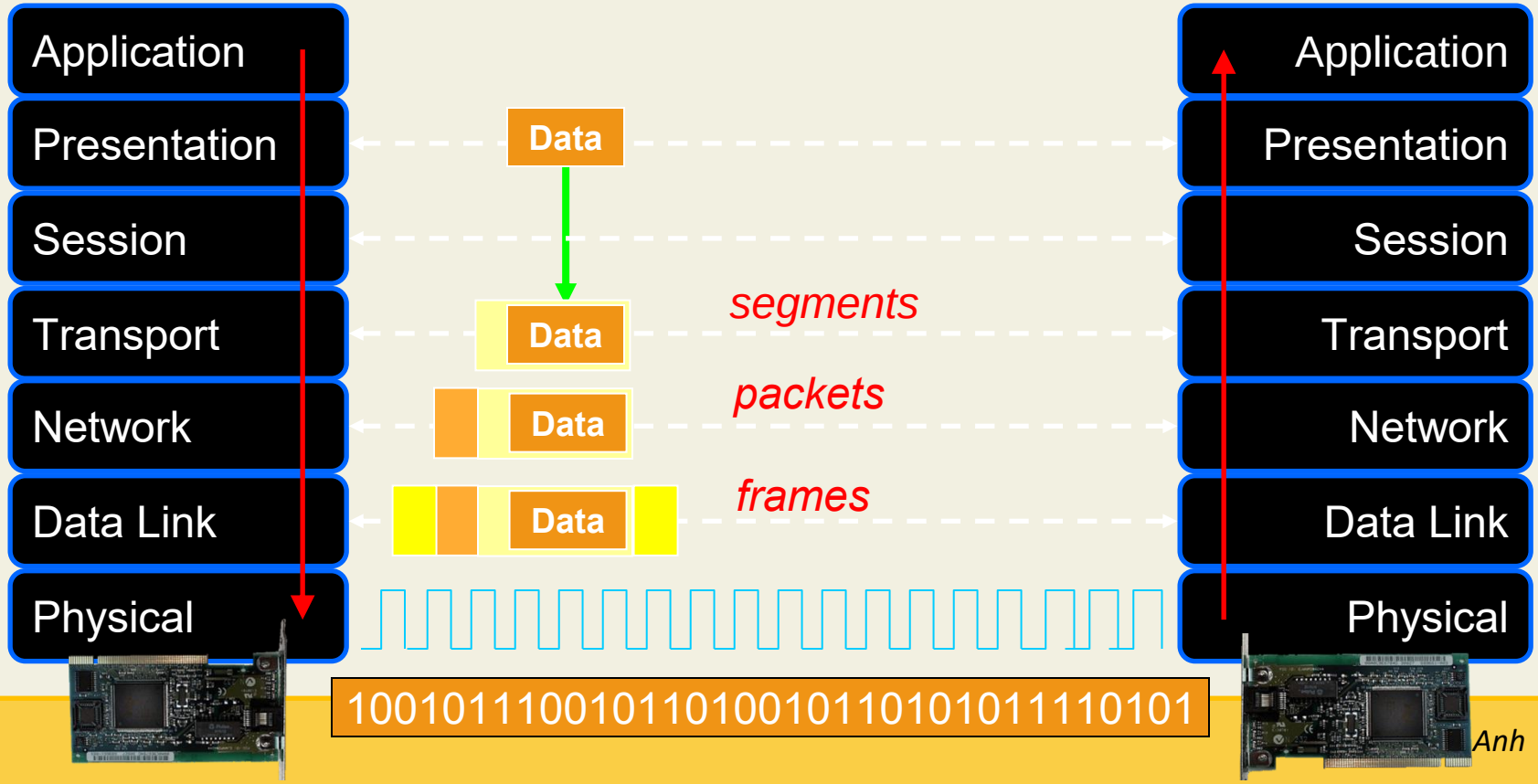
Bí mật



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 lỗi, 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


Application
Presentation
Session
Transport
Network
Data Link
Physical
Data
Data
Data
Data
segments
packets
frames
Application
Presentation
Session
Transport
Network
Data Link
Physical
10010111001011010010110101011110101

6. Các loại tấn công an toàn thông tin

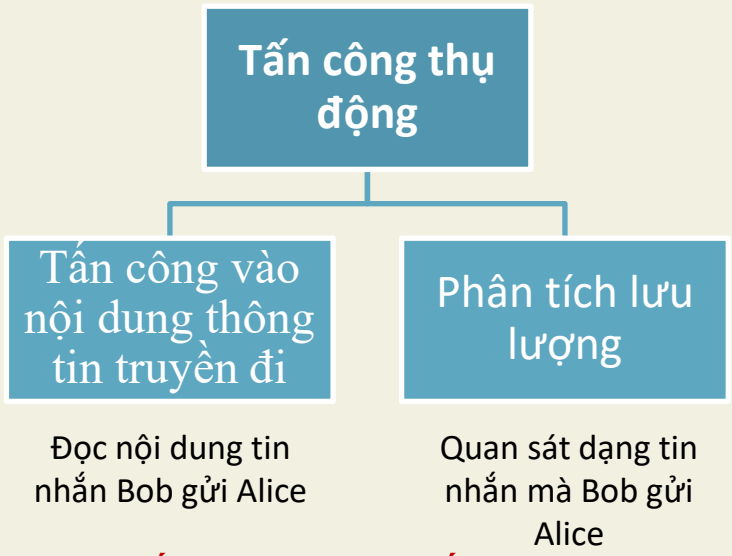
Tấn công an toàn

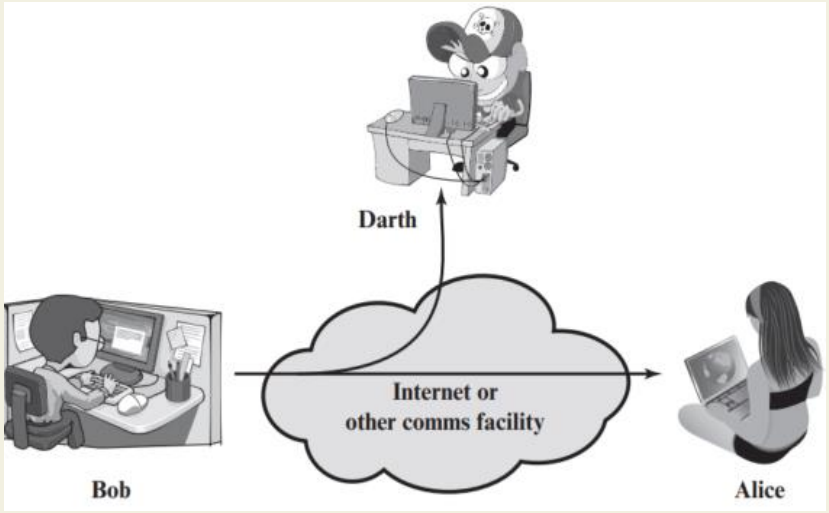
Có 2 loại hình tấn công an ninh chính được sử dụng trong cả X.800 (đây là kiến trúc bảo mật cho hệ thống OSI đươc ITU quy định), tiêu chuẩn RFC 4949 (RFC viết tắt của Request for comment, bao gồm các thuật ngữ bảo mật Internet).

- Tấn công thụ động: là cuộc tấn cống 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 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.



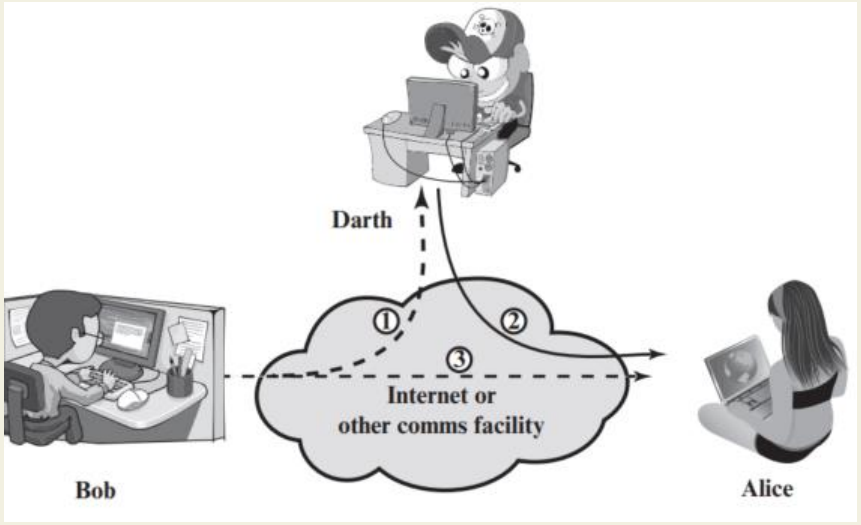


Các cuộc tấn công thụ động rất khó phát hiện do không tạo ra sự thay đổi gì về dữ liệu

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

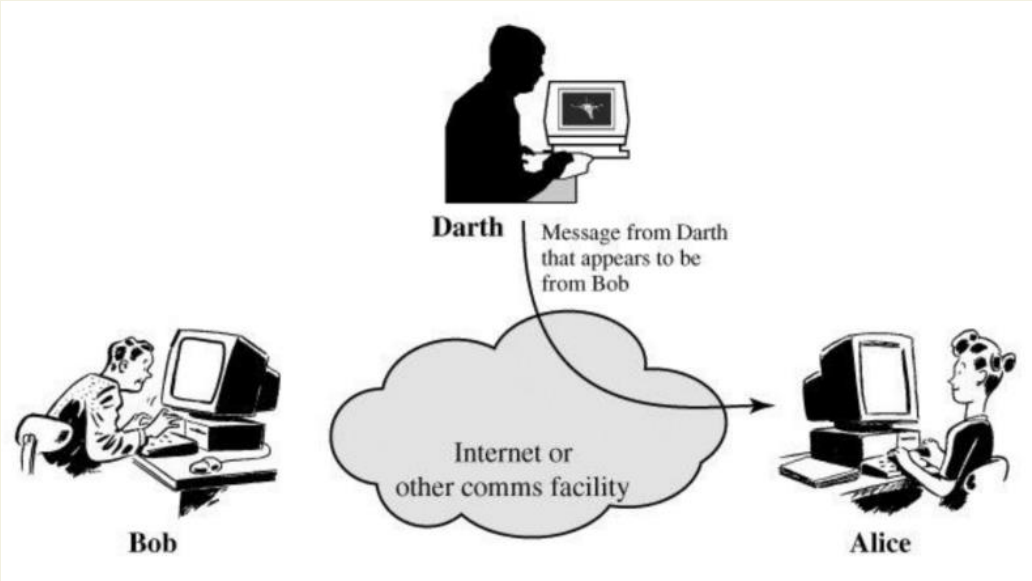
Tấn công chủ động: liên quan đến việc sửa đổi luồng dữ liệu hoặc tạo luồng giả và có thể được chia thành 4 loại: giả mạo, phát lại, sửa đổi thông tin, và từ chối dịch vụ.



6. Các loại tấn công an toàn thông tin

Tấn công chủ động

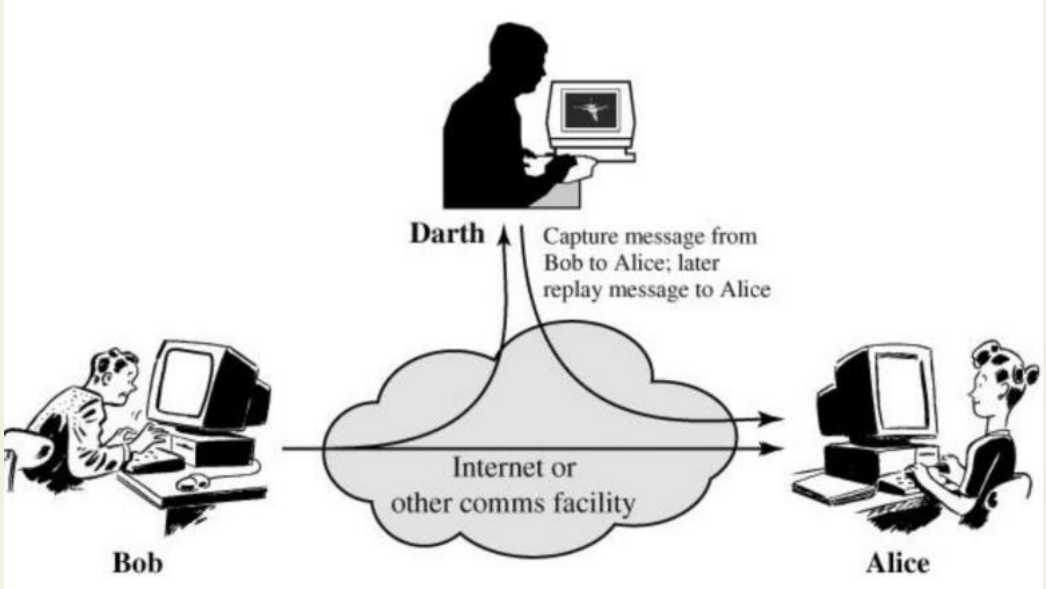
Giả mạo: Diễn ra khi một thực thể giả vờ một thực thể khác (2) – tin nhắn từ Darth tới Alice nhưng lại giả vờ là từ Bob



6. Các loại tấn công an toàn thông tin

Tấn công chủ động

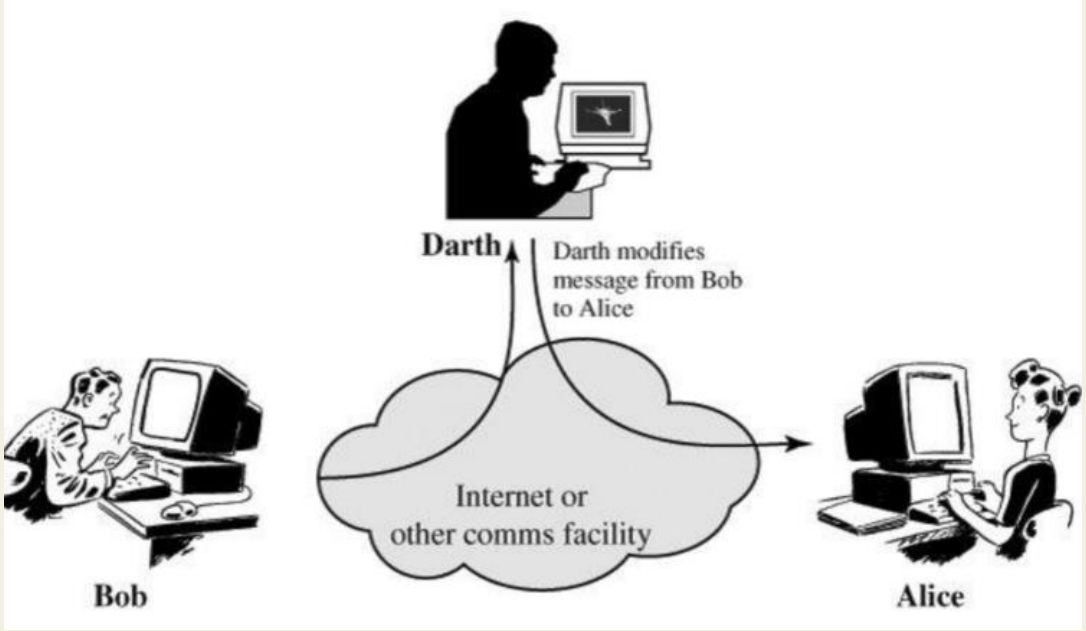
- **Phát lại**: Liên quan đến việc nắm bắt thụ động dữ liệu và truyền lại sau đó tạo ra hiệu ứng không xác thực (1,2,3) – Darth bắt gói tin từ Bob tới Alice; sau đó phát lại tin nhắn tới Alice



6. Các loại tấn công an toàn thông tin

Tấn công chủ động

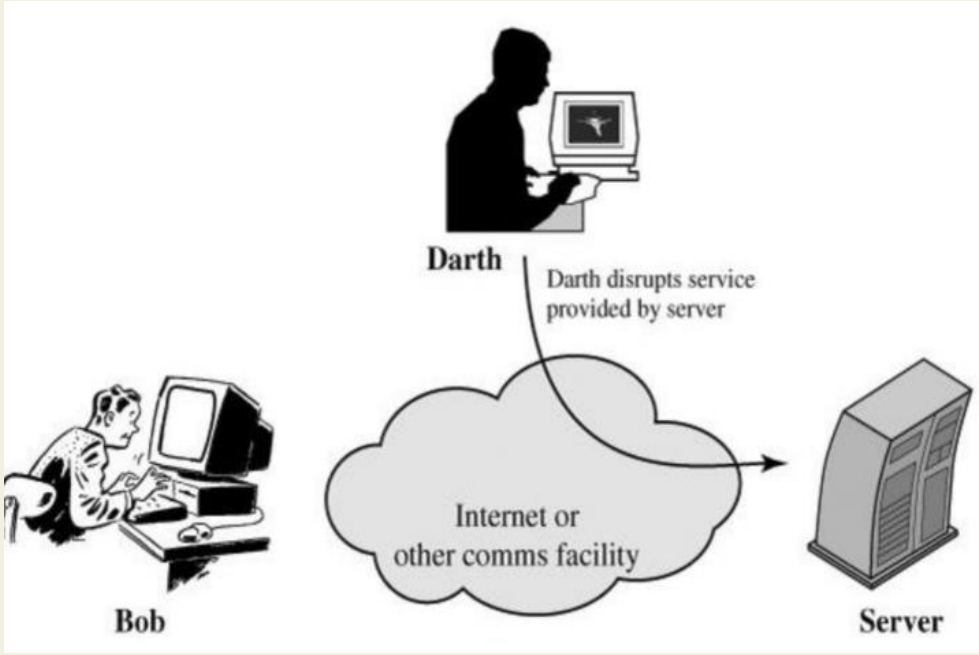
- **Sửa đổi**: tin nhắn bị sửa lại một phần hoặc tin nhắn gửi đi bị trễ để tạo ra hiệu ứng không xác thực (1, 2) – Darth sửa tin nhắn mà Bob gửi cho Alice.



6. Các loại tấn công an toàn thông tin

Tấn công chủ động

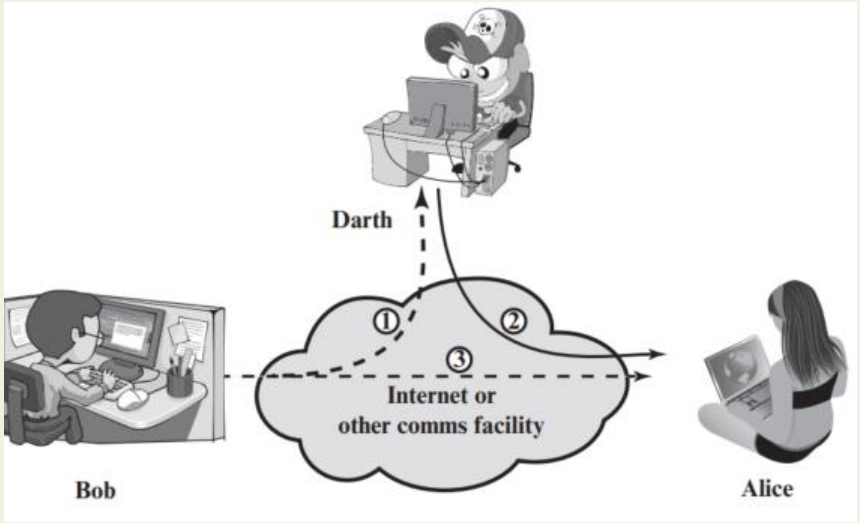
- **Từ chối dịch vụ**: là ngăn chặn hoặc cản trở việc sử dụng hoặc quản lý các phương tiện truyền thông (3 – Darth sẽ ngắt dịch vụ được cung cấp bởi máy chủ).



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

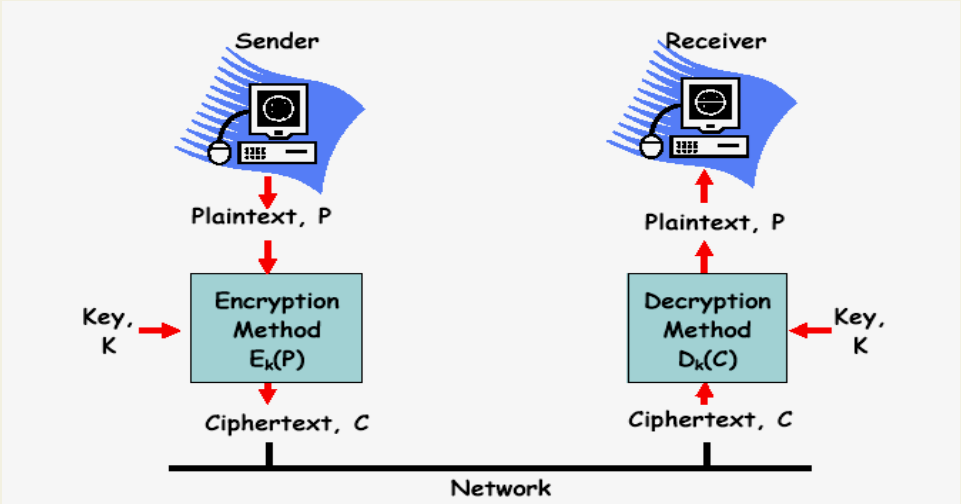


Figure 2: The basic cryptographic techniques

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 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 mã mật là bộ 5 (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.

$d_K(e_K(x))=x$ 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.

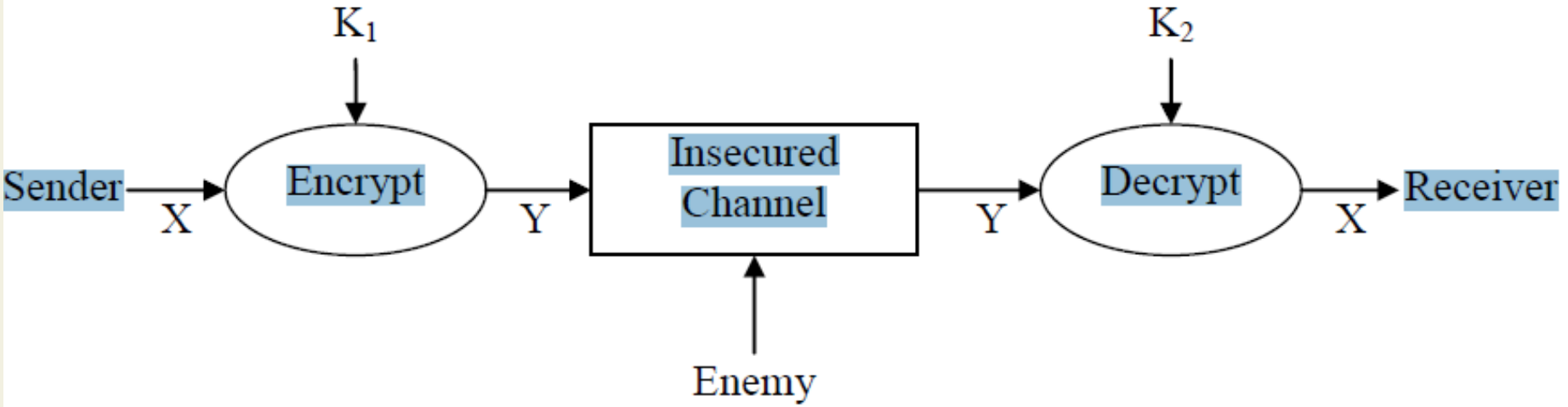
    Nếu thời gian cần thiết dùng để phá vỡ một thuật toán là quá lâu thì thuật toán đó tạm thời được coi là an toàn.

    Nếu lượng dữ liệu cần thiết để phá vỡ một thuật toán quá lơn 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 hoá) 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 hoá 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 mã mối quan hệ sau:

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

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 mod n* là phần dư của phép chia *a* cho *n*. *Ví dụ: 13 mod 4 =1*

Hai số nguyên a và b được gọi là đồng dư modulo với n nếu (a mod n) = (b mod n) và được ký hiệu như sau: a ≡b (mod n). Ví dụ 13 ≡5 (mod 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	$(x + y)\ mod\ n = (y + x)\ mod\ n$ $(x \times y)\ mod\ n = (y \times x)\ mod\ n$
Kết hợp	$[(x + y) + z]\ mod\ n = [x + (y + z)]\ mod\ n$ $[(x \times y) \times z]\ mod\ n = [x \times (y \times z)]\ mod\ n$
Phân phối	$[x \times (y + z)]\ mod\ n = [(x \times y) + (x \times z)]\ mod\ n$
Số đối (-x)	Với mỗi số nguyên $x \in Z_n$ tồn tại số y sao cho $x + y \equiv 0\ (mod\ n)$
Identities	$(0 + x)\ mod\ n = x\ mod\ n$ $(1 \times x)\ mod\ n = x\ mod\ n$

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 *gcd(a,b)*
  - Hai số a và b được gọi là nguyên tố cùng nhau nếu *gcd(a,b) = 1*
  - Thuật toán Ơclit 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:

    gcd(a, b) = gcd (b, a mod b)

    Ví dụ: *gcd(55,22)?*

Chương 2: Cơ sở toán học

Đoạn chương trình sau minh họa cài đặt thuật toán Ơclit 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;
 }
}
```

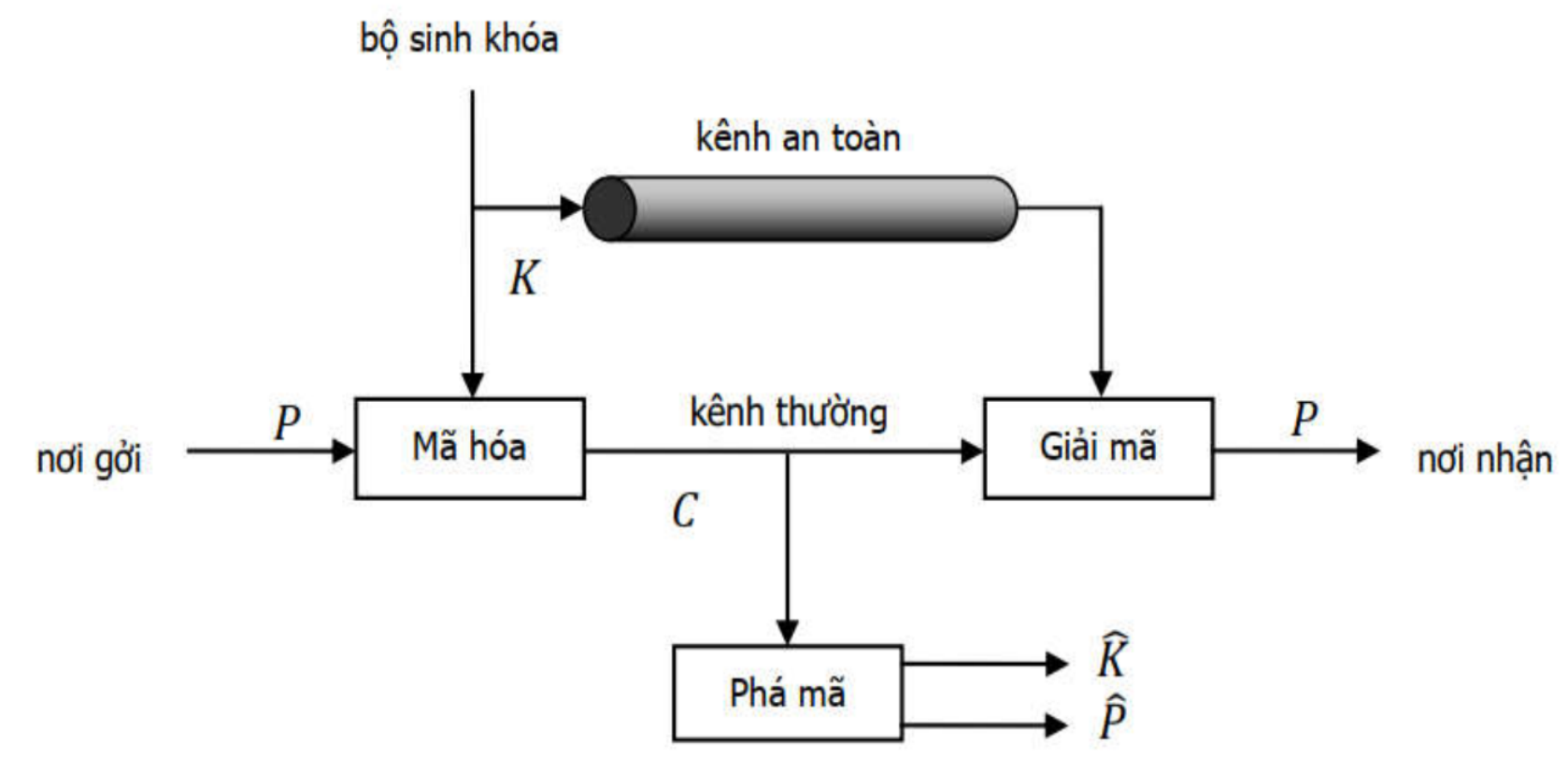
Số nguyên tố: Số nguyên $p > 1$ được gọi là số nguyên tố nếu nó chỉ có ước số là ±1 và ±\. Ví dụ 2 là số nguyên tố vì nó chỉ có các ước số là ±1 và ±2.

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			

73
79
83
89
97

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 hoá thay thế :**

Hệ mã hoá thay thế là hệ mã hoá 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 củ

d. *Thay thế đa sơ 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 hoá:

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

$P = C = Z_{26}$, $K = \{(a,b) \in Z_{26} \times Z_{26}$, ước chung lớn nhất của a và 26 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 cổ điển:

Ví dụ: Giả sử $P = C = Z_{26}$.

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

-*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 $a, b \in \mathbf{Z}_{26}$.

- 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 ~1.1×10^7

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:

$$y_1 = 11x_1 + 3x_2$$
$$y_2 = 8x_1 + 7x_2.$$

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

I. Hệ mã hóa cổ điển:

Chọn số nguyên dương m. Định nghĩa:

$P = C = (\mathbf{Z}_n)^m$ 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:

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

và $d_k(y) = yk^{-1}$ với $y \in C$.

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

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 = "HELP" và giải mã ngược lại bản mã thu được.

I. Hệ mã hóa cổ 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.

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

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

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 $\det(K) = (k_{11}*k_{22} - k_{21}*k_{12}) \bmod N$ 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 $\det(K) = (15 - 6) \bmod 26 = 9$. $GCD(9, 26) = 1$ nên áp dụng thuật toán Ơclit mở rộng tìm được $\det(K)^{-1} = 3$. Vậy $K^{-1} = 3 * \begin{pmatrix} 5 & 23 \\ 24 & 3 \end{pmatrix} = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix}$.

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 hoá đổi chỗ là hệ mã hoá 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 hoá 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: CAELPOPSEEMHLANPIOSSUCWTITSBIUEMUTERATSGYAERBTX		

I. Hệ mã hóa cổ điển:

Các kỹ thuật đổi chổ:

1. *Đảo ngược toàn bộ bản rỏ*: 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 3×5 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 "IEICSELTEENNHTR".

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:

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

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

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

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

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	Từ	Mã hoá
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 5x5 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 dược 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

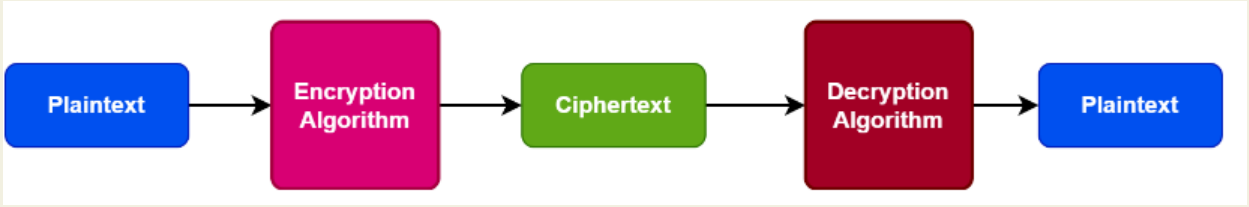
– Các trường hợp khác, mỗi chữ cái được thay bởi chữ cái khác cùng hàng, trên cột chữ cái cùng cặp Ví dụ: ET sẽ thay bằng KL

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)

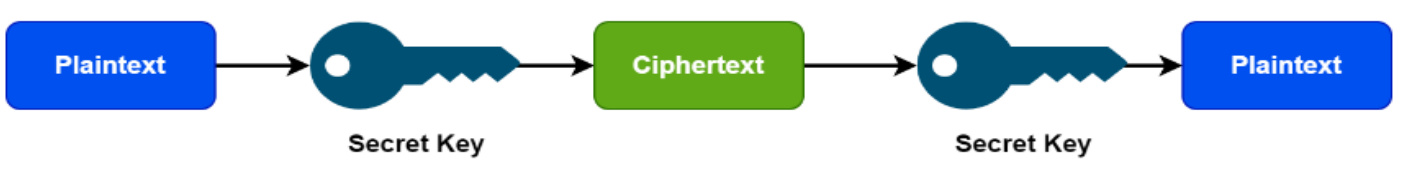
Mật mã học giúp bảo đảm các yếu tố sau cho dữ liệu:

- **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.

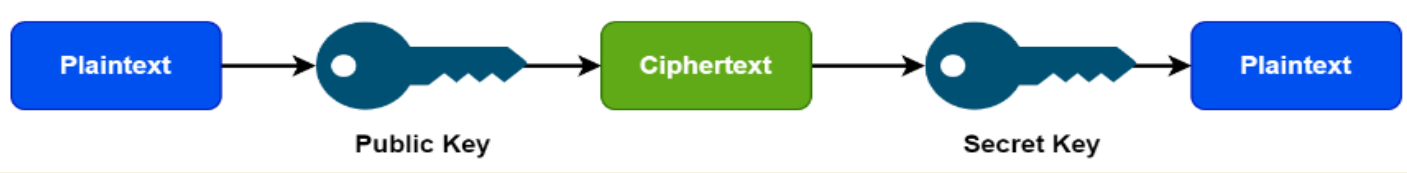
I. Mã hóa (nhắc lại)

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

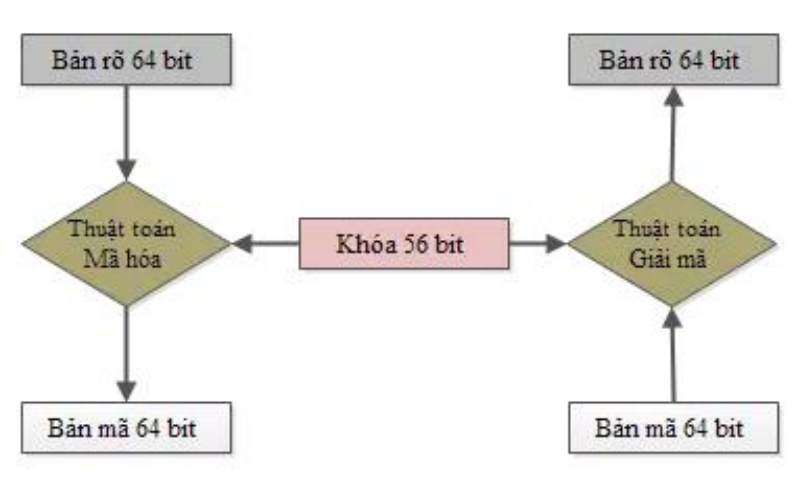
I. Mã hóa (Nhắc lại)

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



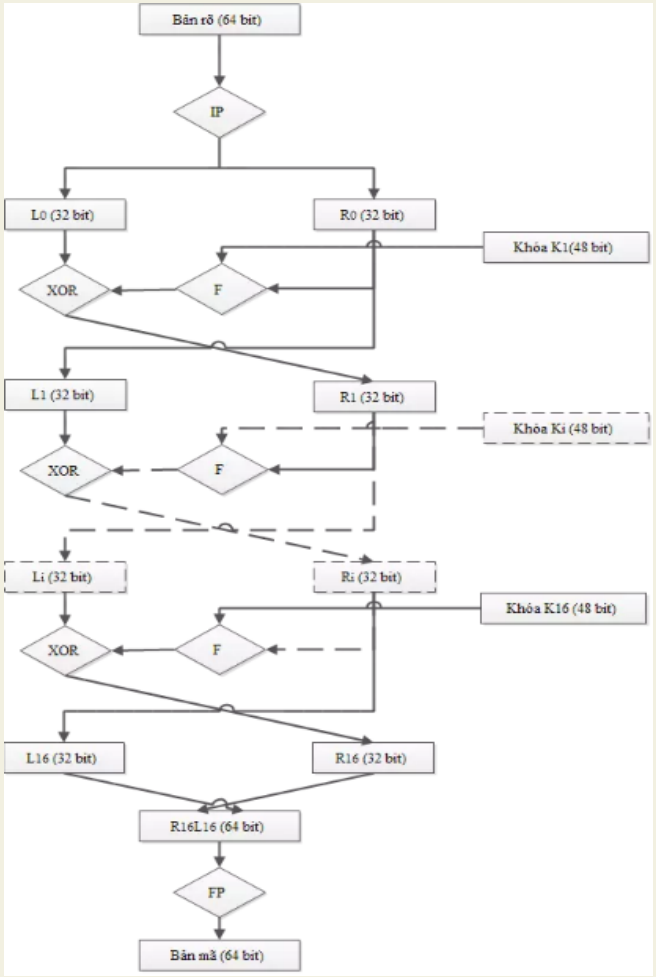
2.3. Mật mã DES (Data Encryption Standard)

- **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),… (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:

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

- 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:

$L_i=R_{i-1}$

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

với:

$\oplus$ là phép XOR của hai xâu bit:

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

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

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)

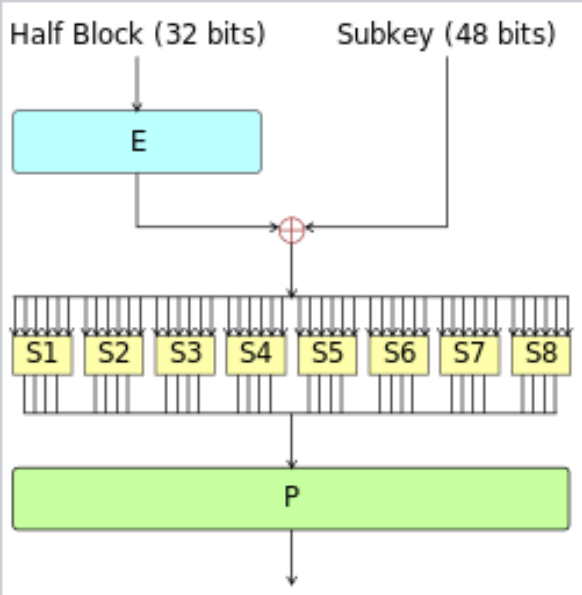
- IP là một phép hoán vị vị trí của các ký tự trong mỗi từ 64 bít, từ vị trí thứ nhất đến vị trí thứ 64.
- Bảng dưới đây cho ta phép hoán vị IP, với cách biểu diễn là bít thứ nhất của IP(x) là bít thứ 58 của từ x (có 64 bít), bít thứ hai của IP(x) là bít thứ 50 của x,…
- Bảng của phép hoán vị IP^{-1}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

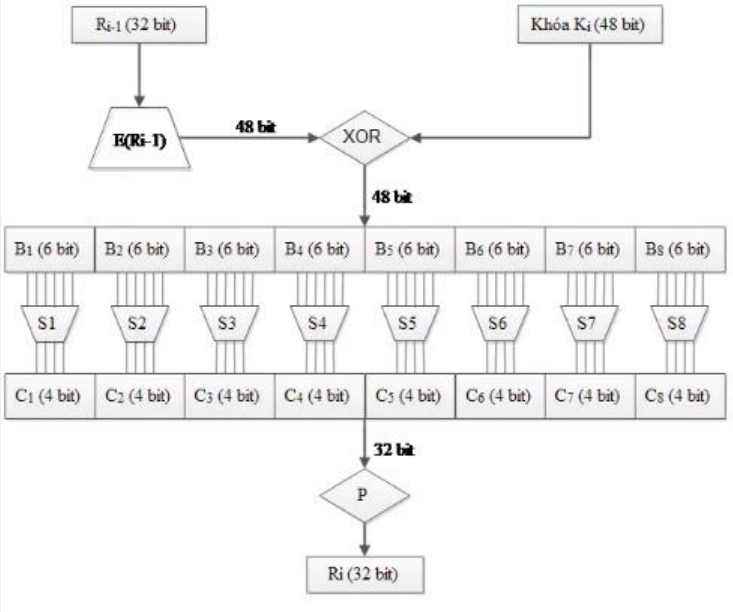
IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
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:



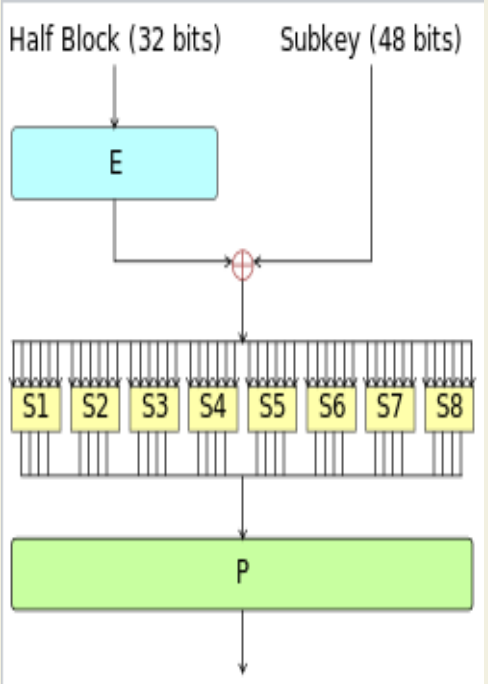


2.1. Mật mã DES (Data Encryption Standard)

- **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:

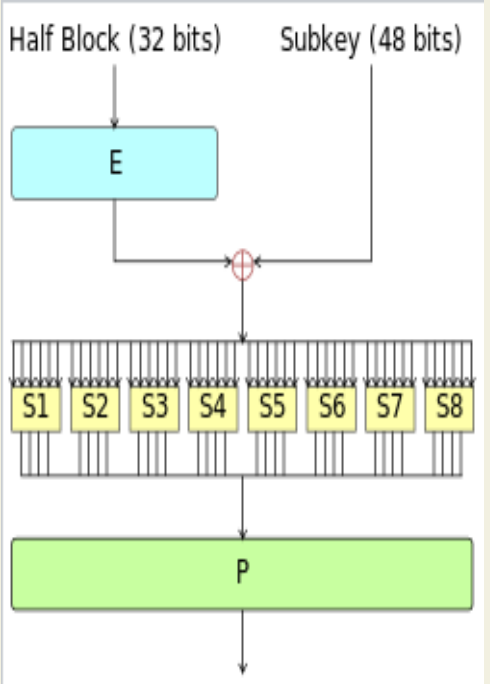
Phép hoán vị "mở rộng" E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

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



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 Bj 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 (hàng r=10_b+1=3, cột s=0111_b+1 =8)

S_2(011000) = 12_d=1100_b (hàng r=00_b+1=1, cột s=1100_b+1 =13)

S_3(100110) =?

S là bí mật rất quan trọng trong bảo đảm tính bí mật của DES

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

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													
1	13	0	11													
2	1	4	11													
3	6	11	13													

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

2.1. Mật mã DES (Data Encryption Standard)

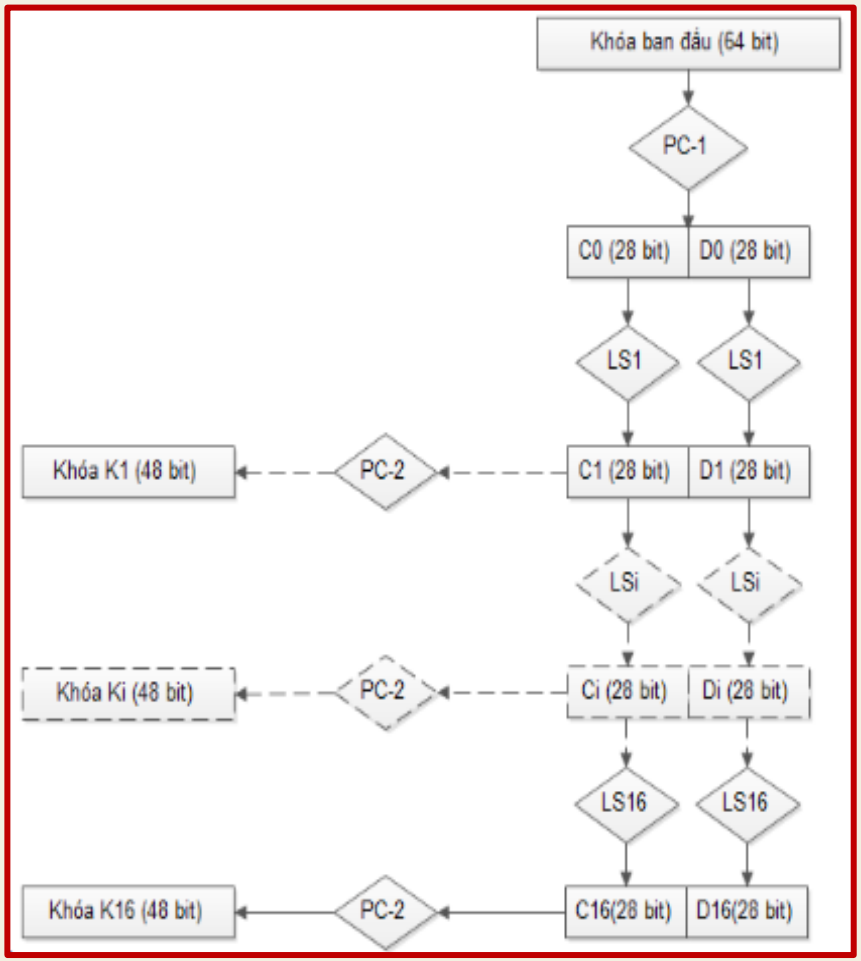
Phép hoán vị P trong sơ đồ của hàm f được cho ở bảng dưới đây:
Mỗi 4 bít đầu ra của các hộp S-box sẽ được ghép lại, theo thứ tự các hộp và được đưa vào hộp P-box. P đơn giản chỉ là phép hoán vị các bít với nhau.

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**: $K_1, K_2, \ldots, K_{16}$
- Các khóa con đều được sinh ra từ khóa chính của DES bằng thuật toán sinh khóa con (thuật toán G)
- LS: left shilft
- Khóa mật mã K là một từ 56 bít, ta chia thành 8 khối, mỗi khối 7 bít, ta cho thêm mỗi khối 7 bít đó một bít kiểm tra tính chẵn lẻ vào vị trí cuối để được một từ 64 bít, 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_i, 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,….,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},…,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 bít → 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} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} 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 ½ 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 $\overline{U}$ Là phần tử bù của U (ví dụ 0100101 là phần bù của 1011010) thì DES có tính chất sau:

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

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

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:

$$K_1 = K_2 = ... = K_{15} = K_{16}$$

=>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:

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

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

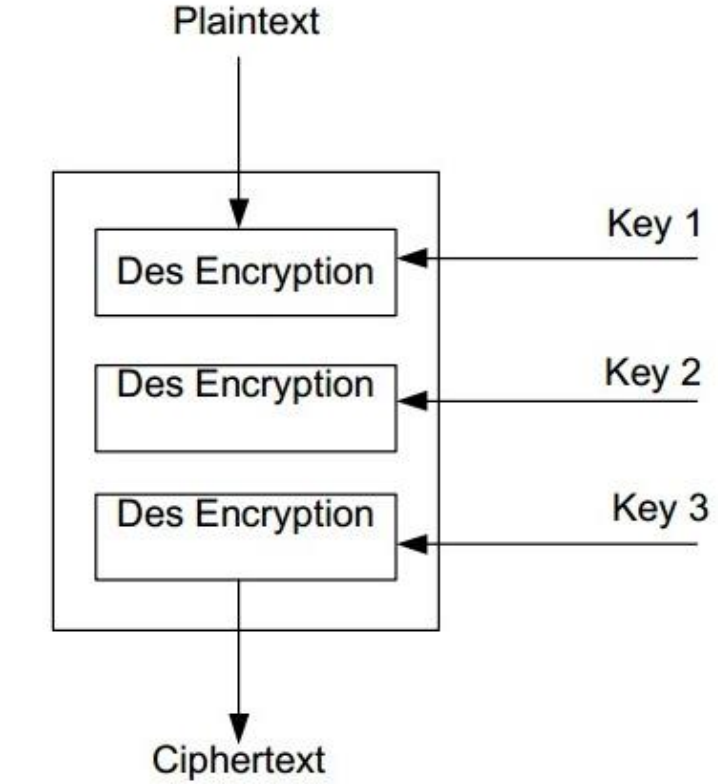
C_0	D_0	Semi-weak key (Hex)								C_0	D_0
$\{01\}^{14}$	$\{01\}^{14}$	01FE	01FE	01FE	01FE	FE01	FE01	FE01	FE01	$\{10\}^{14}$	$\{10\}^{14}$
$\{01\}^{14}$	$\{10\}^{14}$	1FE0	1FE0	0EF1	0EF1	E01F	E01F	F10E	F10E	$\{10\}^{14}$	$\{01\}^{14}$
$\{01\}^{14}$	$\{0\}^{28}$	01E0	01E0	01F1	01F1	E001	E001	F101	F101	$\{10\}^{14}$	$\{0\}^{28}$
$\{01\}^{14}$	$\{1\}^{28}$	1FFE	1FFE	0EFE	0EFE	FE1F	FE1F	FE0E	FE0E	$\{10\}^{14}$	$\{1\}^{28}$
$\{0\}^{28}$	$\{01\}^{14}$	011F	011F	010E	010E	1F01	1F01	0E01	0E01	$\{0\}^{28}$	$\{10\}^{14}$
$\{1\}^{28}$	$\{01\}^{14}$	E0FE	E0FE	F1FE	F1FE	FEE0	FEE0	FEF1	FEF1	$\{1\}^{28}$	$\{10\}^{14}$

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 khóa 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.



Mô tả 3DES

2.2. Mật mã 3-DES (Triple DES)

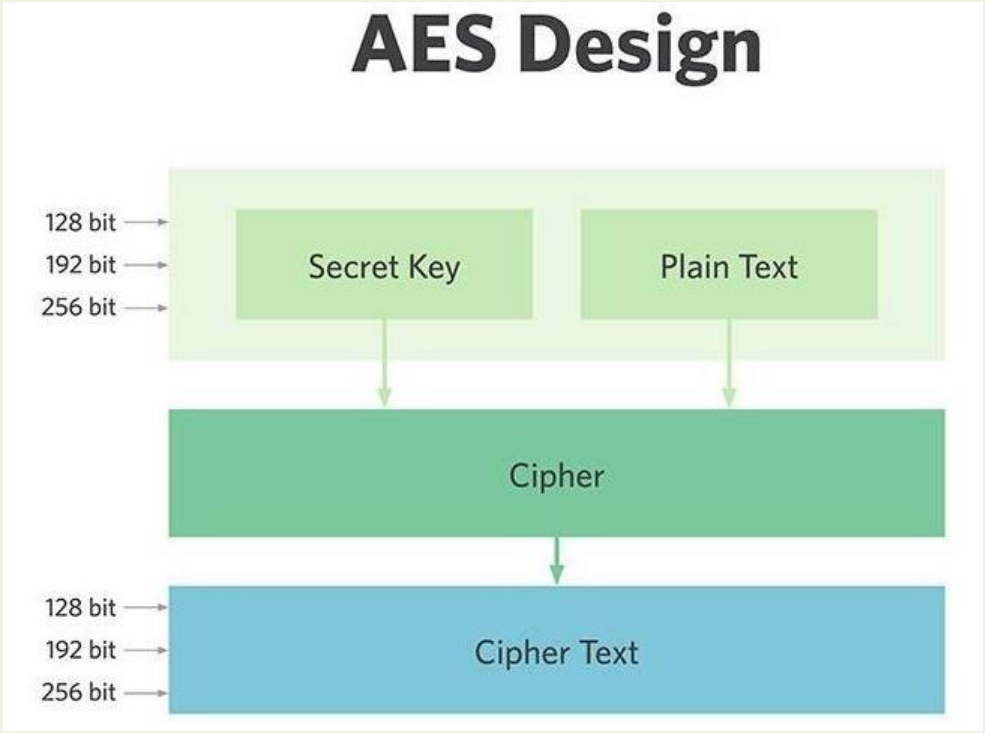
- ***Ưu và nhược điểm của 3DES***

- **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.

CƠ SỞ TOÁN HỌC VÀ MÃ HÓA CÔNG KHAI

Chương 3: Mật mã khóa công khai (*Public Key Cryptosystems*)

1. **Mã Hóa Khóa Bí Mật và Nhược Điểm**

•Mã hóa khóa bí mật chỉ sử dụng MỘT khóa trong cả quá trình mã hóa và giải mã (đối xứng).

- Khóa này phải được giữ bí mật.
- Các nhược điểm của khóa bí mật:
  - Cần có kênh an toàn để trao đổi khóa.
  - Trên môi trường mạng có N người dùng, thì cần N(N-1)/2 khóa để N(N-1)/2 cặp người trao đổi thông tin (tổ hợp chập 2 của N phần tử) -> Cần quá nhiều khóa cho nên việc quản lý khóa phức tạp

1. Mã Hóa Khóa Bí Mật và Nhược Điểm (tt)

- Không thể thiết lập được chữ ký điện tử

->Sử dụng quy trình **mã hóa khóa công khai.**

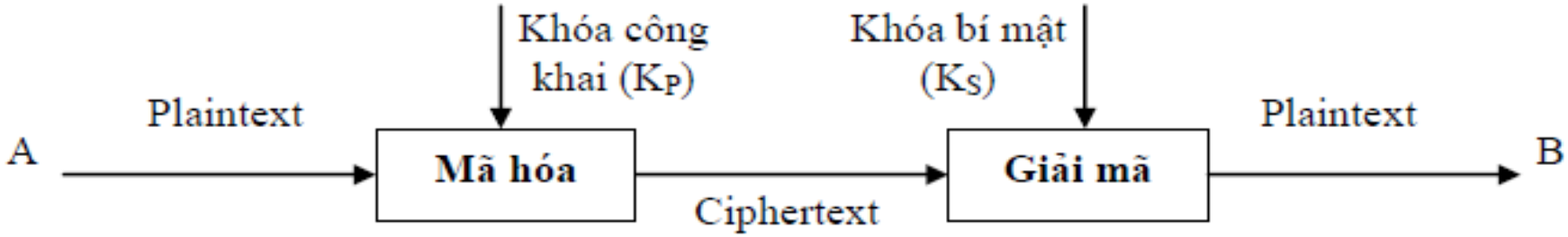
2. Ý tưởng của Diffie & Hellman

- Diffie & Hellman (1975-76) đã đề xuất một loại hệ mã với nguyên tắc mới, được gắn với một NSD nhất định chứ không phải là gắn với một cuộc truyền tin giữa một cặp NSD.
  - mỗi user có hai khoá: một khoá bí mật (K_S) và một khoá công khai (K_P) -- tự do phổ biến công khai.
  - Khoá thứ nhất gắn liền với giải mã, còn khoá thứ hai với sinh mã.
- Với các hệ mã khóa công khai việc phân phối khóa sẽ trở nên dễ dàng hơn qua các kênh cung cấp khóa công cộng, số lượng khóa hệ thống quản lý cũng sẽ ít hơn (là n khóa cho n người dùng).
- Các dịch vụ mới như chữ ký điện tử, thỏa thuận khóa cũng được xây dưng dựa trên các hệ mã này.

2. Ý tưởng của Diffie & Hellman(tt)

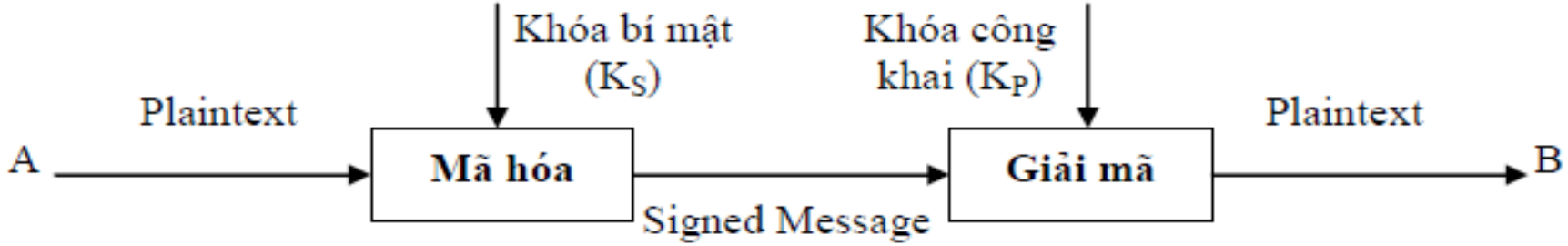
- Các yêu cầu của loại hệ mã này:
  - Việc sinh K_S,K_P phải dễ dàng
  - Việc tính E(K_P, M) là dễ dàng
  - Nếu có C = E(K_P, M) và K_S thì việc tìm bản rõ cũng là dễ
  - Nếu biết K_P thì việc dò tìm K_S là khó
  - Vệc giải mã bản rõ từ mã là rất khó
- Khi A muốn truyền tin cho B, A sẽ sử dụng khóa K_P của B để mã hóa tin tức và truyền bản mã tới cho B, B sẽ sử dụng khóa bí mật của mình để giải mã và đọc tin:

3. Mô hình hệ mã hóa PKC



Mô hình sử dụng 1 của các hệ mã khóa công khai PKC

Ciphertext = $E(K_P, Plaintext)$,Plantext = $D(K_S, E(K_P, Plaintext))$ (1)



Mô hình sử dụng 2 của các hệ mã khóa công khai PKC

3. Mô hình hệ mã hóa PKC(tt)

->Các Thuật toán PKC là bất đối xứng vì quá trình mã hóa và giải mã dùng các khóa khác nhau, hay vai trò của người gửi và nhận không tương đương: Người mã hóa thông điệp hoặc kiểm tra chữ ký không thể giải mã hoặc tạo nên chữ ký.

4. Nguyên tắc cấu tạo một hệ PKC

- Một hệ mã PKC có thể được tạo dựng trên cơ sở sử dụng một hàm kiểu one - way (1 chiều). Một hàm f được gọi là one-way nếu:
  - Đối với mọi X tính ra Y = f(X) là dễ dàng.
  - Khi biết Y rất khó để tính ra X. Hay việc tìm f^{-1} là khó
- Ví dụ. Cho n số nguyên tố p1, p2, ...pn ta có thể dễ dàng tính được N = p1 * p2 * ... * pn, tuy nhiên khi biết N, việc tìm các thừa số nguyên tố của nó là khó khăn hơn rất nhiều

4. Nguyên tắc cấu tạo một hệ PKC(tt)

- Cần một hàm one-way đặc biệt, trang bị một trap-door (cửa bẫy), sao cho nếu biết trap-door này thì việc tính X khi biết f(X) (tức là đi tìm nghịch đảo của f) là dễ, còn ngược lại thì khó
- Một hàm one-way có trap door như thế -> một hệ mã PKC
  - Lấy E_z (hàm sinh mã) là hàm one- way có trap-door.
  - Trap- door chính là khoá mật,mà nếu biết nó thì có thể dễ dàng tính được cái nghịch đảo của E_Z tức là biết Dz, còn nếu không biết thì rất khó tính được.

5. An Toàn Của Mã Hóa Khóa Công Khai

- Độ an toàn của thuật toán mã hóa khóa công khai phụ thuật vào độ khó của bài toán ngược (tính $f^{-1}(y)$ khi không có thông tin bổ sung (khóa bí mật)).
- Thám mã bằng phương pháp vét cạn khóa về mặt lý thuyết là luôn luôn có thể thực hiện được.
- Nhưng trên thực tế các khóa sử dụng là quá lớn cho việc vét cạn (>512bit).
- Để chống lại một số phương pháp thám mã tiên tiến khác, cần phải sử dụng các khóa rất lớn (>>512 bit).
- Do vậy việc cài đặt thuật toán khóa công khai chậm hơn nhiều so với thuật toán khóa bí mật.

6. Một số hệ mã khóa công khai

6.1 Hệ mã knapsack

- 1978, hai ông Merkle - Hellman đã đề xuất một thuật toán mã hoá PKC dựa trên bài toán xếp ba lô:

Bài toán xếp ba lô tổng quát như sau:

Cho M, N và $A_1, A_2, ..., A_N$ là các số nguyên dương tìm các số x_i không âm sao cho:

$$M = \sum_{i=1}^{N} x_i * A_i$$

Vecto $A = (A_1, A_2, ..., A_N)$ được gọi là vecto xếp balô còn vectơ $X = (x_1, x_2, ..., x_N)$ là vectơ nghiệm.

Một trường hợp riêng đá ng quan tâm của bài toán xếp ba lô tổng quát là trường hợp mà $x_i \in \{0, 1\}$. Khi đó ta có bài toán xếp ba lô 0, 1.

6.1 Hệ mã knapsack (tt)

Vecto xếp ba lô siêu tăng : Trong trường hợp vecto $(A_1, A_2, ..., A_N)$ được sắp lại thành $(A'_1, A'_2, ..., A'_N)$ sao cho:

$\forall$ i ta có: $\sum_{j<i} A'_j < A'_i$ thì vecto $(A_1, A_2, ..., A_N)$ được gọi là vecto xếp balo siêu tăng.

Khi $(A_1, A_2, ..., A_N)$ là một vecto xếp balo siêu tăng ta có ngay tính chất: $M >= A'_i \ \forall$ i. Do đó việc giải bài toán xếp ba lô 0/1 trở nên dễ dàng hơn rất nhiều.

6.1 Hệ mã knapsack (tt)

Cách xây dựng:

1. Chọn 1 vecto siêu tăng $A' = (a'_1, a'_2, ..., a'_N)$, chọn 1 số $M > 2 * a'_N$, chọn ngẫu nhiên 1 số $u < M$ và $(u, M) = 1$

2. Xây dựng Vecto $A = (a_1, a_2, ..., a_N)$ trong đó $a_i = (a'_i * u) \bmod M$

3. Khóa: $K_P = (A, M)$, $K_S = (u, u^{-1})$

4. Không gian các bản rõ là không gian mọi dãy N bit

$$P = (x_1, x_2, ..., x_n).$$

Mã hóa: $C = (\sum_{i=1}^{N} a_i * x_i) \bmod M$

Giải mã: tính $C' = C * u^{-1} \bmod M$ sau đó giải bài toán xếp ba lô 0/1 với A', C' từ đó tìm được $P = (x_1, x_2, ..., x_n)$.

6.1 Hệ mã knapsack (tt)

Ví dụ 1: Cho hệ mã Knapsack có $A' = (2, 3, 6, 12, 25)$, $N = 5$, $M = 53$, $u = 46$, $u^{-1} = 15$.

a) Hãy tìm các khóa của hệ mã trên

b) Mã hóa và giải mã bản mã tương ứng của bản rõ M = 01001.

Nhận xét chung về hệ mã PKC

- Kể từ năm 1976, nhiều giải pháp cho PKC đã được nêu ra nhưng khá nhiều trong số đó đã bị phá vỡ hoặc bị đánh giá là không thực dụng do dung lượng tính toán lớn hoặc thông tin nở ra quá lớn khi mã hoá.
- Một hệ thống PKC có thể đáp ứng 2 mục đích:
  - Bảo mật thông tin và truyền tin.
  - Chứng thực và chữ ký điện tử.
- Hai thuật toán đáp ứng các ứng dụng trên thành công nhất là RSA và El-Gamal.
- Nói chung PKC chậm, không thích hợp cho on-line encryption.
  - Cần khi yêu cầu tính an toàn cao và chấp nhận tốc độ chậm.
  - Ngoài ra người ta thường sử dụng kết hợp PKC và SKC***(symmetric key cryptosystems)***:
    - dùng PKC để tạo khóa bí mật thống nhất chung giữa hai bên truyền tin để thực hiện pha truyền tin chính bằng SKC sau đó.

6.2 Hệ mã RSA

- RSA là một thuật toán mã hóa khóa công khai.
- Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa.
- RSA đang được sử dụng phổ biến trong thương mại điện tử và được cho là đảm bảo an toàn với điều kiện độ dài khóa đủ lớn.
- RSA được ***Ron Rivest***, ***Adi Shamir*** và ***Len Adleman*** giới thiệu năm 1977 tại Học viện Công nghệ Massachusetts (MIT).
- RSA dựa trên tính khó của bài toán phân tích các số lớn ra thừa số nguyên tố
  - Biết một số nguyên tố nhân chúng với nhau để thu được một hợp số là dễ còn biết hợp số, phân tích nó ra thừa số nguyên tố là khó.

Ý tưởng(Motivation)

- Ý tưởng của các nhà phát minh là gắn các thuật toán sinh mã và mã hoá với phép toán lấy luỹ thừa trên trường $Z_n = \{0,1,2,..n-1\}$.
  - Chẳng hạn, việc sinh mã cho tin X sẽ được thực hiện qua: $Y = X^e \bmod(n)$
  - Còn việc giải mã: $X = Y^d \bmod(n)$
  - Do đó e và d phải được chọn sao cho: $X^{ed} = X \pmod n$
- Trong đó X là đoạn tin, e là khóa công khai được sử dụng để mã hóa, Y là đoạn tin đã được mã hóa, d là khóa bí mật dùng để giải mã

Hiện thực ý tưởng

- Người ta đã tìm được cách xây dựng cặp số (e,d) này trên cơ sở công thức như sau:

$$X^{\phi(n)} = 1 \ (mod\ n)$$ (định lý Ơ - le)

  - $\phi(n)$ là số các thuộc Z_n mà nguyên tố cùng nhau với n.
  - $\phi(n)$ có thể tính được khi đã biết công thức phân tích thừa số nguyên tố của n, cụ thể là nếu đã biết n = p*q (p,q là nguyên tố) thì $\phi(n) = (p-1)(q-1)$.
- Người ta chọn e*d sao cho chia $\phi(n)$ dư 1, hay

$$d = e^{-1} \ (mod\ \phi\ (n)),$$

khi đó ta sẽ có điều cần thiết:

$$X^{ed} = X^{k.\phi(n)+1} = (X^{\phi(n)})^d.X = 1.X = X \quad (mod\ n)$$

- Tóm lại: Nếu đã biết e và
  - Biết PTTSNT của n ➔ tìm được $d = e^{-1} \ (mod\ \phi(n))$ tức $X^{ed} = X \ (mod\ n)$
  - Nếu không biết PTTSNT của n thì **rất khó**.

Thuật toán RSA

Thuật toán RSA bao gồm 3 bước:

1. Tạo khóa
2. Mã hóa
3. Giải mã

1. Tạo khóa

Để tạo cặp khóa, thực hiện các bước sau:

1. Chọn 2 số nguyên tố lớn: p và q, với p # q và p, q >120 chữ số. Lựa chọn ngẫu nhiên và độc lập.
2. Tính số n = p.q
3. Tính số $\phi(n) = (p-1).(q-1)$ (hàm phi Euler)
4. Chọn số e sao cho $1 < e < \phi(n)$ và là số nguyên tố cùng nhau với $\phi(n)$ (tức là: $\gcd(e, \phi(n)) = 1$, để đảm bảo $e^{-1} \bmod \phi(n)$ tồn tại duy nhất)
5. Tính số $d = e^{-1} \bmod \phi(N)$

Khi đó, ta có khóa là các bộ số:

Khóa công khai: $K_u = \{e,n\}$

Khóa cá nhân: $K_r = \{d,p,q\}$

2. Mã hóa

- Thông điệp ban đầu: m (0 < m < n)
- Sử dụng khóa công khai K_u = {e, n} để tính thông điệp mã hóa (ciphertext): c

$$\mathbf{c = m^e \bmod n}$$

3. Giải mã

- Người nhận dùng khóa bí mật $K_r = \{d,p,q\}$ để tính lại thông điệp gốc m từ thông điệp đã mã hóa c:

$$\mathbf{m = c^d \bmod n}$$

Ví dụ

- Ta chọn hai số nguyên tố p và q ,với p= 5 và q = 7
- Tính n = p*q = 5*7 = 35.
- ϕ(n) = (p- 1)*(q-1) = (5-1)(7-1) = 24
- Tiếp đến chọn e thoả 1< e< n

-> chọn e= 5.

- Tìm d ,sao cho e*d $\equiv$ 1 mod(24)

-> Tính được d = 29.

- Do đó: Public key =(n,e) =(35,5)

  Private key = (d,p,q) =(29,5,7)

Ví dụ (tt)

Áp dụng mã hóa chuỗi sau: ***secure***

Ta có bảng sau:

Nôi dung	vị trí	M^e	Nội dung bị mã hoá
S	19	246099	24
E	5	3125	10
C	3	243	33
U	21	4084101	21
R	18	1889568	23
e	5	3125	10

Ví dụ (tt)

Giải mã chuỗi ***secure***

Nội dung bị mã hoá	$M = c^d \bmod n$	Dữ liệu gốc
24	19	S
10	5	E
33	3	C
21	21	u
23	18	R
10	5	e

6.3 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 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 $\mathbf{K} = y^k \bmod p$
  3. Sau đó tính ***cặp ciphertext*** $\{C_1, C_2\}$:
     - $C_1 = a^k \bmod p$
     - $C_2 = \mathbf{K} \cdot M \bmod p$
- 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 {C1, C2} 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 mod p**
   - *(vì C_1^x mod p = $a^{k.x}$ mod p = y^k mod p = K)*
2. Tính **K^{-1} mod p**
3. Tính **M = C_2 . K^{-1} mod p**

Ví Dụ Mã Hóa ElGamal

- Tạo khóa:
  1. Chọn **p** = 97, **a** = 5,
  2. Chọn **x** = **58**,
  3. Tính: **y** = 5^{58} = **44** 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** mod 97
  - Tính cặp ciphertext {C_1, C_1}:
    - $\mathbf{C_1} = 5^{36} = \mathbf{50}$ mod 97,
    - $\mathbf{C_2}$= 75.3 = **31** mod 97
- Giải mã: {**50, 31**}:
  - Tính **K** = 50^{58} = 75 mod 97,
  - Tính $\mathbf{K^{-1}}$ = 22 mod 97
  - Tính thông điệp ban đầu: **M** = $\mathbf{C_2 . K^{-1}}$ **mod p** = 31.22 mod 97= **3** mod 97