An toàn thông tin

Giáo viên: Lê Quốc Anh

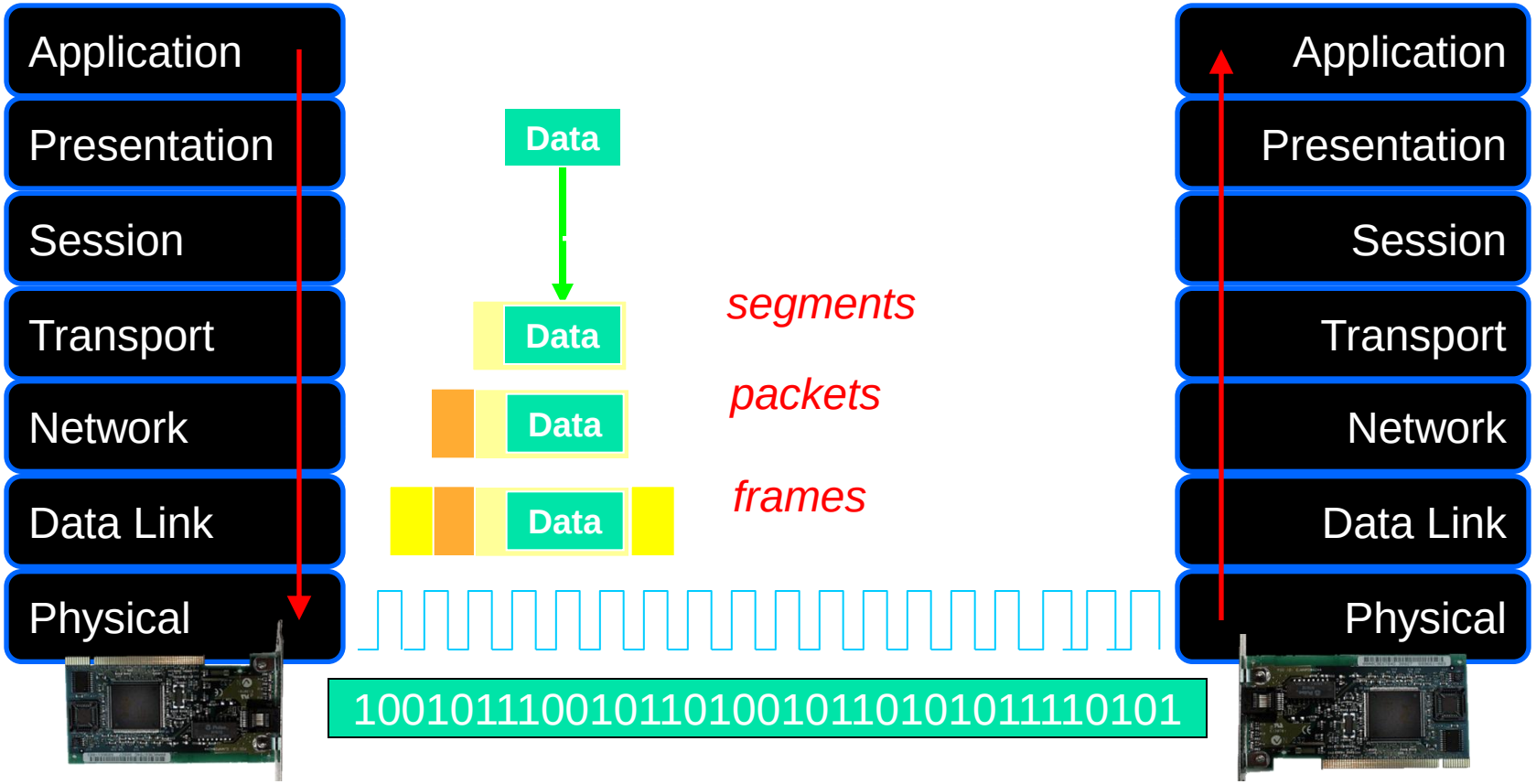
Tài liệu tham khảo

1. Giáo trình an toàn và bảo mật thông tin. Trường đại học Hàng Hải

   http://www.mediafire.com/?abdxi97upa2nqxf

2. An toàn thông tin, tác giả PGS.TS Thái Hồng Nhị, TS Phạm Minh Việt NXB khoa học và kỹ thuật.

Chương 1: Tổng quan về an toàn và bảo mật thông tin



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 ngƣời 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 chung của an toàn bảo mật thông tin

- **Tính bí mật** *(Confidentiality)*: - Đảm bảo rằng thông tin không bị truy cập bất hợp pháp
  - Thuật ngữ *privacy* thường được sử dụng khi dữ liệu được bảo vệ có liên quan tới các thông tin mang tính cá nhân.
- **Tính toàn vẹn** *(Integrity):* - Đảm bảo rằng thông tin không bị sửa đổi bất hợp pháp.
- **Tính sẵn dùng** *(availability):* - Tài sản luôn sẵn sàng được sử dụng bởi nhưng người có thẩm quyền.
- **Tính xác thực** *(Authentication):* - Đảm bảo rằng dữ liệu nhận được chắc chắn là dữ liệu gốc ban đầu
- **Tính không thể chối bỏ** *(Non-repudation):* - Đảm bảo rằng người gửi hay người nhận dữ liệu không thể chối bỏ trách nhiệm sau khi đã gửi và nhận thông tin.

5. 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 bao gồm hai quá trình:** 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

5. 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

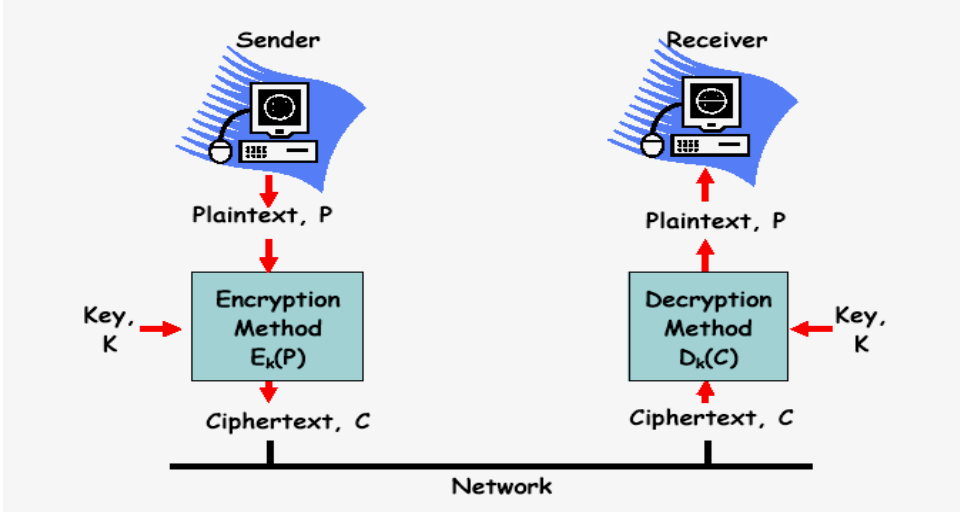


Figure 2: The basic cryptographic techniques

5. An toàn thông tin bằng mật mã

- **Có 2 phương thức mã hoá cơ bản: thay thế và hoán vị**:
  - ✓ **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.

6. Hệ mật mã

- **Vai trò của 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.

6. 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.

6. Hệ mật mã

- **Các thành phần của một 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.

Đối với mỗi $k \in K$ có một quy tắc mã $e_K: P \rightarrow C$ và một quy tắc giải mã tương ứng $d_K \in D$.

Với mỗi $e_K: P \rightarrow C$ và $d_K: C \rightarrow P$ là những hàm mà

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

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

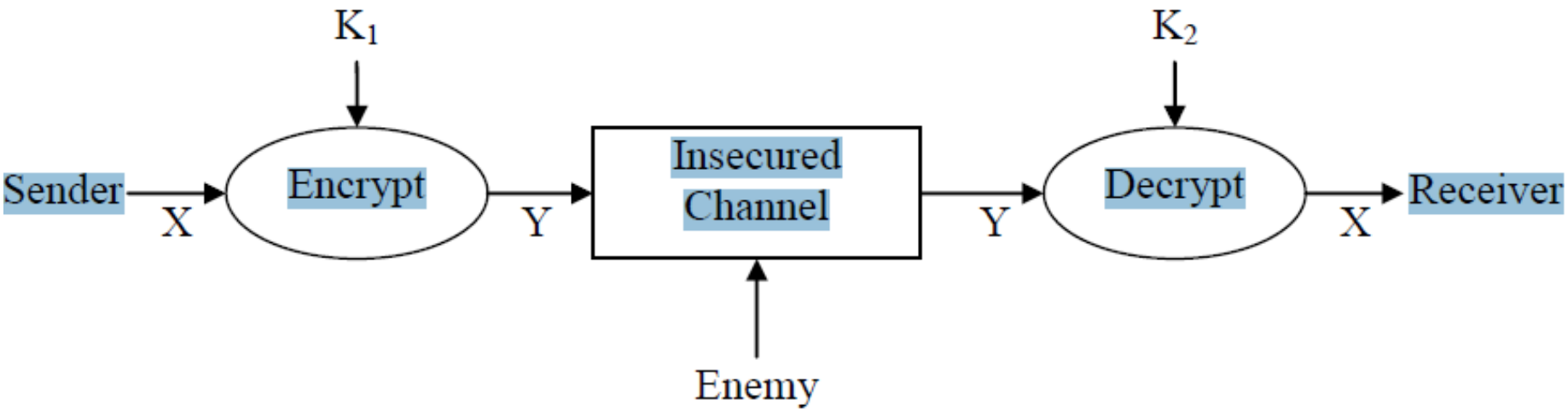
7.Tiêu chuẩn đánh giá hệ mật mã

- **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.

7.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.

8. 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

8. 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.

9. 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

Chương 2: 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:

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.

I. Hệ mã hóa cổ điển:

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.

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

I. Hệ mã hóa cổ điển:

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:

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:

A	B	C	D	...	L	M	N	...	W	X	Y	Z
0	1	2	3	...	11	12	13	...	22	23	23	25

I. Hệ mã hóa cổ điển:

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

- Do đó chẳng hạn xâu “ANGLES” sẽ được mã hóa thành “DQJOHV”.

I. Hệ mã hóa cổ điển:

- 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:

Cho $P = C = Z_n$

$K = \{(a,b) \in \mathbf{Z}_n \times \mathbf{Z}_n : \gcd(a,n) = 1\}$

Với mỗi khóa $k = (a,b) \in K$, định nghĩa:

$e_k(x) = (ax+b) \bmod n$ và $d_k(x) = (a^{-1}(y-b)) \bmod n$ với $x, y \in \mathbf{Z}_n$

$E = \{e_k, k \in K\}$ và $D = \{D_k, k \in K\}$

-giải mã chính xác thông tin ???

-e_k phải là song ánh

$$\forall y \in Z_n, \exists! x \in Z_n, ax + b \equiv y (\bmod n)$$

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

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

I. Hệ mã hóa cổ điển:

<u>Bài Tập:</u>

- a = 5, b = 3: y = 5x + 3 (mod 26).

Mã hoá: ANTOANTHONGTIN → ?

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:

Chọn số nguyên dương m. Định nghĩa $P = C = K = (\mathbf{Z}_n)^m$

$$K = \{(k_1, k_2, \ldots, k_m) \in (Z_n)^m\}$$

Với mỗi khóa $k = (k_1, k_2, \ldots, k_m) \in K$, định nghĩa:

$$e_k(x_1, x_2, \ldots, x_m) = ((x_1 + k_1) \bmod n, (x_2 + k_2) \bmod n, \ldots, (x_m + k_m) \bmod n)$$

$$d_k(y_1, y_2, \ldots, y_m) = ((y_1 - k_1) \bmod n, (y_2 - k_2) \bmod n, \ldots, (y_m - k_m) \bmod n)$$

với $x, y \in (\mathbf{Z}_n)^m$.

I. Hệ mã hóa cổ điển:

<u>Ví dụ</u>: $m = 6$ và keyword là ***CIPHER***

-Suy ra, khóa $k = (2, 8, 15, 7, 4, 17)$

-Cho bản rõ: ***thiscryptosystemisnotsecure***

19	7	8	18	2	17
2	8	15	7	4	17
21	15	23	25	6	8

24	15	19	14	18	24
2	8	15	7	4	17
0	23	8	21	22	15

18	19	4	12	8	18
2	8	15	7	4	17
20	1	19	19	12	9

13	14	19	18	4	2
2	8	15	7	4	17
15	22	8	25	8	19

20	17	4
2	8	15
22	25	19

-Vậy bản mã là: **“vpxzgiaxivwoubttmjpwizitwzt”**

I. Hệ mã hóa cổ điển:

Bài tập 3.14: Cho hệ mã Vigenere có M = 6. Mã hóa xâu P = "THIS IS MY TEST" người ta thu được bản mã là "LLKJML ECVVWM".

a) Hãy tìm khóa mã hóa đã dùng của hệ mã trên.

b) Dùng khóa tìm được ở phần trên hãy giải mã bản mã C = "KLGZWT OMBRVW".

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

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

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 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) mod 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) * $\begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix}$ = $\begin{pmatrix} 3 & 15 \end{pmatrix}$ = “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:

Bài tập 3.12: Cho hệ mã Hill có M = 2.

a) Ma trận A = $\begin{bmatrix} 15 & 13 \\ 7 & a \end{bmatrix}$ được sử dụng làm khóa cho hệ mã trên. Hãy tìm tất cả các khóa có thể sử dụng của hệ mã trên.

b) Giả sử người ta sử dụng hệ mã trên để mã hóa bản rõ P = "MARS" và thu được bản mã là "YARH". Hãy thực hiện giải mã với bản mã là C = "MANNTF" và đưa ra bản rõ.

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".

Nhận xét: Hạn chế của phương pháp này là toàn bộ các ma trận ký tự phải được sinh để mã hoá và giải mã.

I. Hệ mã hóa cổ điển:

3. *Hoán vị các ký tự của bản rỏ 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 rỏ: $M = m_1 m_2 \ldots m_d m_{d+1} \ldots m_{2d}$

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 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

Chương II. Chuẩn mã dữ liệu DES

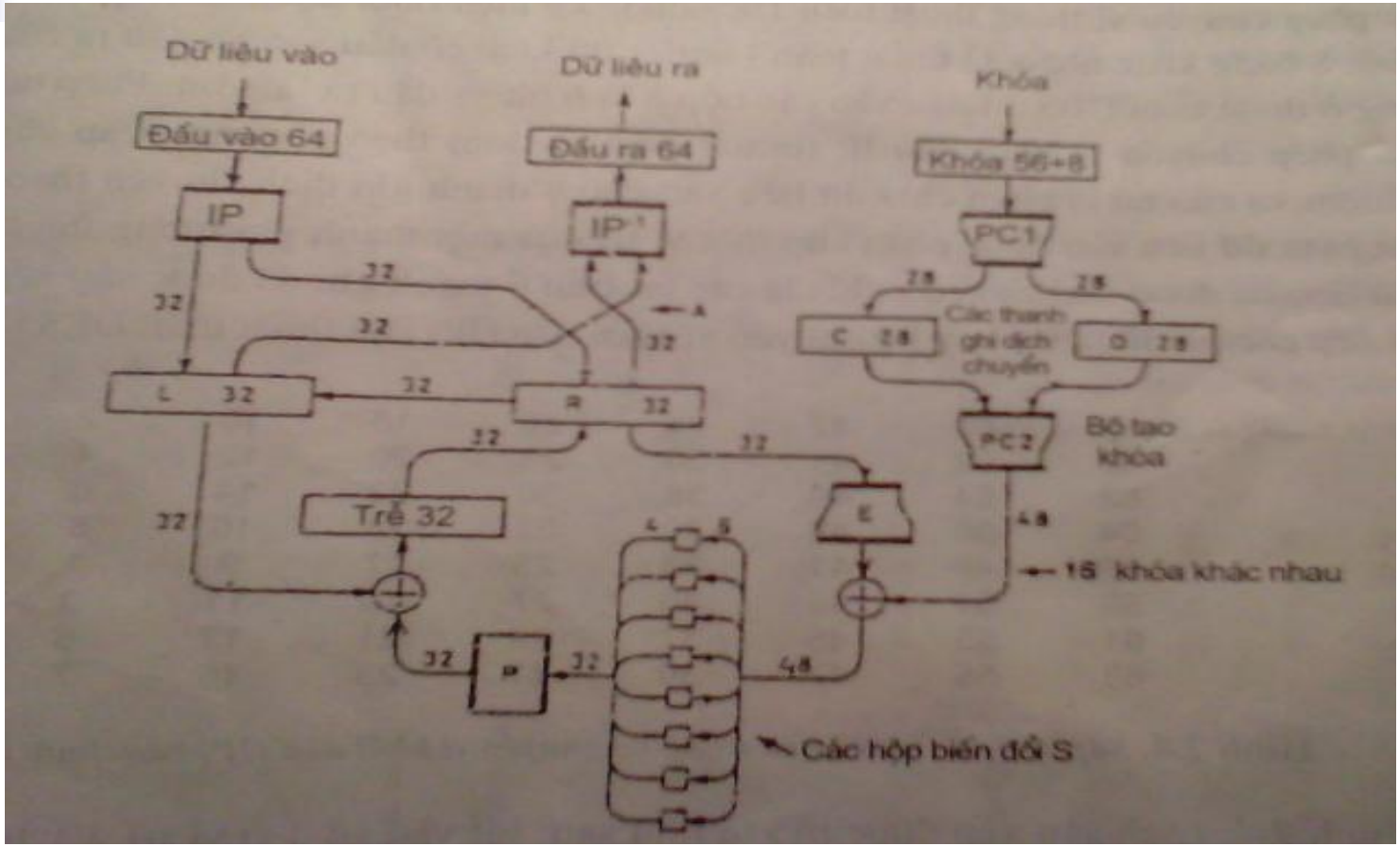
1.Giới thiệu chung về DES

- Ngày 13/5/1973 ủy ban quốc gia về tiêu chuẩn của Mỹ công bố yêu cầu về hệ mật mã áp dụng cho toàn quốc. Điều này đã đặt nền móng cho chuẩn mã hóa dữ liệu, hay là DES.
- Lúc đầu DES được công ty IBM phát triển từ hệ mã Lucifer, công bố vào năm 1975.
- Sau đó DES được xem như là chuẩn mã hóa dữ liệu cho các ứng dụng.

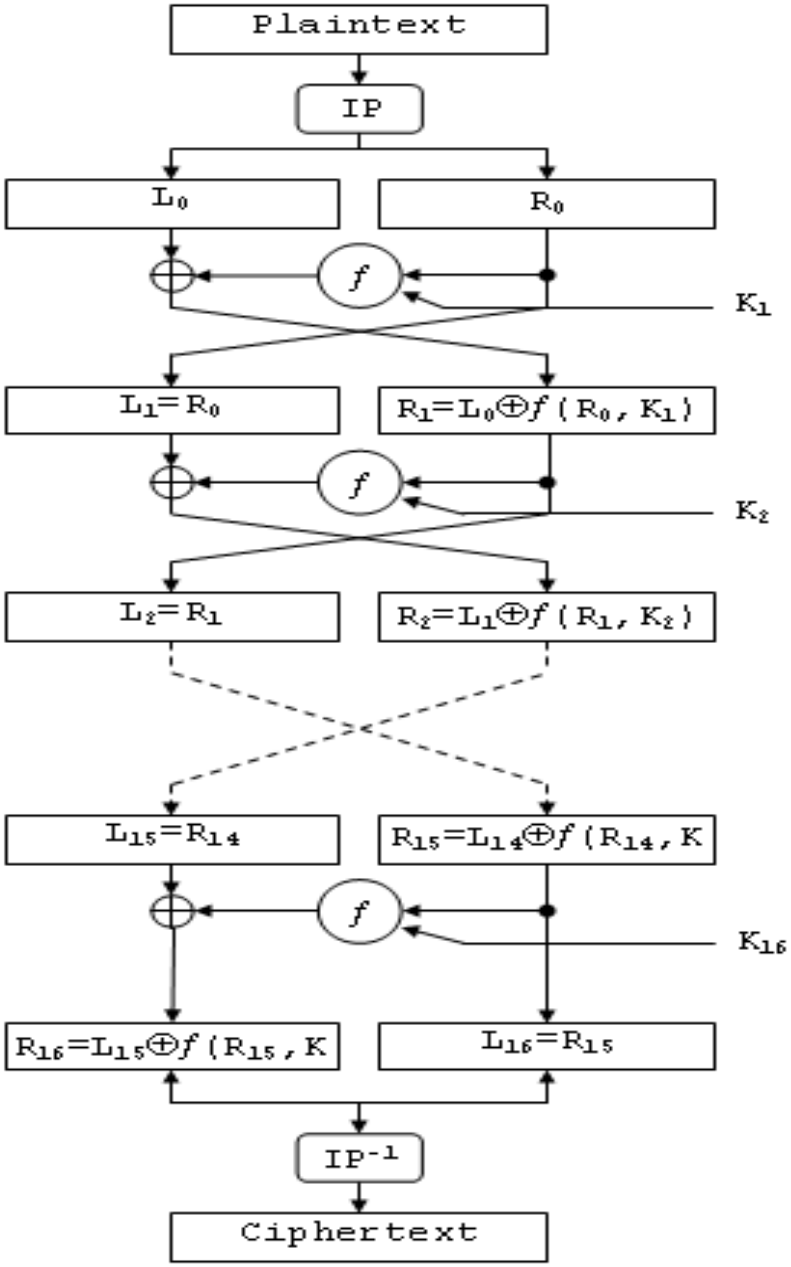
2. Đặc điểm của thuật toán DES

- DES là thuật toán mã hóa khối, độ dài mỗi khối là 64 bit
- Khóa dùng trong DES có độ dài toàn bộ là 64 bit. Tuy nhiên chỉ có 56 bit thực sự được sử dụng; 8 bit còn lại chỉ dùng cho việc kiểm tra.
- DES xuất ra bãn mã 64 bit.
- Thuật toán thực hiện 16 vòng
- Mã hoá và giải mã được sử dụng cùng một khoá.
- DES được thiết kế để chạy trên phần cứng.

3. Mô tả thuật toán

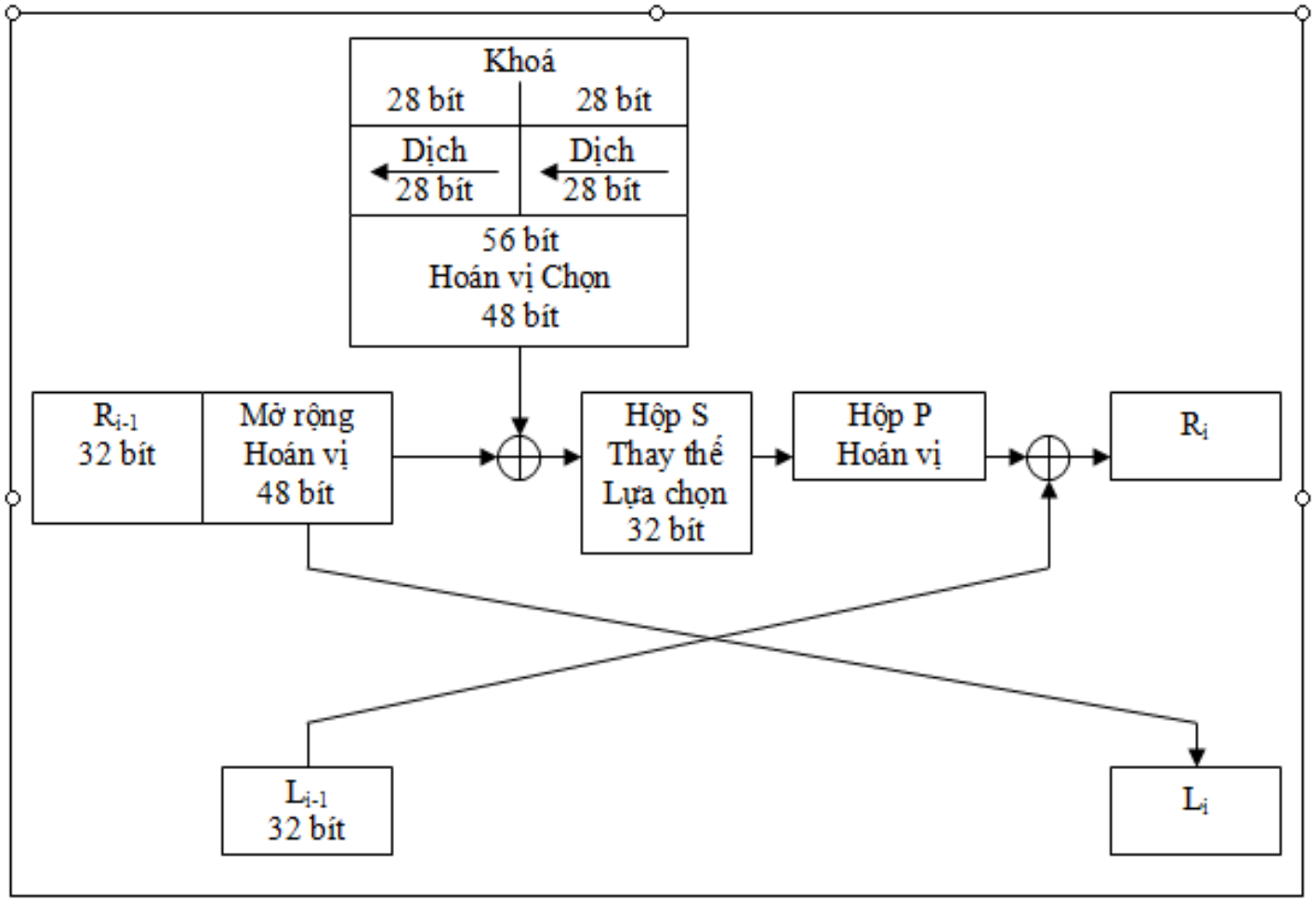


Cấu trúc logic của thuật toán DES



Lưu đồ thuật toán DES

3. Mô tả thuật toán



Một vòng lặp DES

3. Mô tả thuật toán

Thuật toán được thực hiện trong 3 giai đoạn:

1. Cho bản rõ x (64bit) được hoán vị khởi tạo IP (Initial Permutation) tạo nên xâu bit x_0.

   $x_0=IP(x)=L_0R_0$

   L_0 là 32 bit đầu tiên của x_0.
   R_0 là 32 bit cuối của x_0.

3. Mô tả thuật toán

Bộ chuyển vị IP

Hoán vị khởi đầu nhằm đổi chỗ khối dữ liệu vào , thay đổi vị trí của các bít trong khối dữ liệu vào. Ví dụ, hoán vị khởi đầu chuyển bít 1 thành bít 58, bít 2 thành bít 50, bít 3 thành bít 42,...

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

3. Mô tả thuật toán

2. 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.

3. Mô tả thuật toán

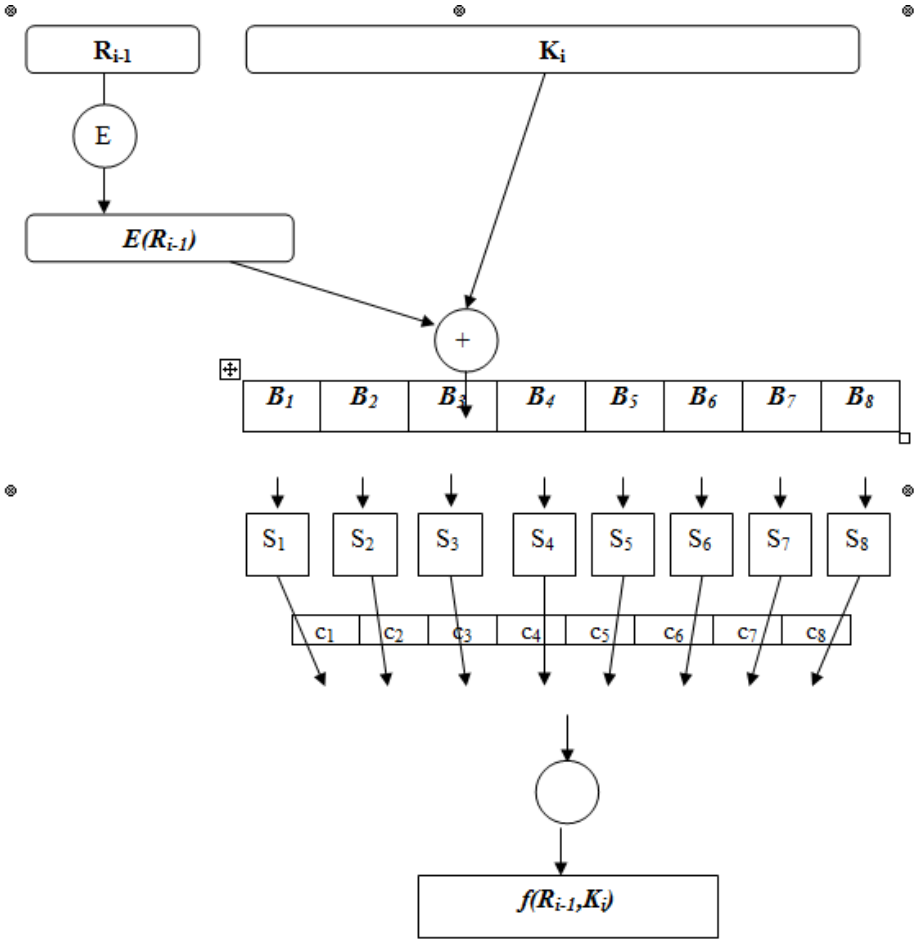
3. Tại vòng thứ 16, R16 đổi chỗ cho L16. Sau đó ghép 2 nửa R16 L16 cho đi qua hoán vị nghịch đảo của hoàn vị IP sẽ tính được bản mã. Bản mã cũng có độ dài 64 bít.

Hoán vị 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

3. Mô tả thuật toán

Hàm F



3. Mô tả thuật toán

Hàm F

Hàm f lấy đối số đầu vào là xâu nhập R_{i-1} (32 bit) đối số thứ hai là K_i (48 bit) và tạo ra xâu xuất có độ dài 32 bit. Các bước sau được thực hiện.

1. Đối số đầu R_{i-1} sẽ được "mở rộng" thành xâu có độ dài 48 bit tương ứng với hàm mở rộng E cố định. $E(R_i)$ bao gồm 32 bit từ R_i, được hoán vị theo một cách thức xác định, với 16 bit được tạo ra 2 lần.

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

Hàm mở rộng E

3. Mô tả thuật toán

2. Tính $E(R_{i-1}) \oplus K_i$ kết quả được một khối có độ dài 48 bit. Khối này sẽ được chia làm 8 khối $B=B_1B_2B_3B_4B_5B_6B_7B_8$. Mỗi khối này có độ dài là 6 bít.

3. Bước kế tiếp là cho các khối Bi đi qua hộp S_i sẽ biến một khối có độ dài 6 bit thành một khối C_i có độ dài 4 bít.

3. Mô tả thuật toán

Hộp S

- Mỗi hộp S-box là một bảng gồm 4 hàng và 16 cột được đánh số từ 0. Như vậy mỗi hộp S có hàng 0,1,2,3. Cột 0,1,2,…,15. Mỗi phần tử của hộp là một số 4 bít. Sáu bít vào hộp S sẽ xác định số hàng và số cột để tìm kết quả ra.
- Mỗi khối Bi có 6 bít kí hiệu là b1, b2, b3, b4, b5 và b6. Bít b1 và b6 được kết hợp thành một số 2 bít, nhận giá trị từ 0 đến 3, tương ứng với một hàng trong bảng S. Bốn bít ở giữa, từ b2 tới b5, được kết hợp thành một số 4 bít, nhận giá trị từ 0 đến 15, tương ứng với một cột trong bảng S.

3. Mô tả thuật toán

Hộp S1

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

Hộp S2

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

3. Mô tả thuật toán

Hộp S3

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

Hộp S4

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

3. Mô tả thuật toán

Hộp S5

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

Hộp S6

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

3. Mô tả thuật toán

Hộp S7

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

Hộp S8

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

3. Mô tả thuật toán

Ví dụ: Ta có B1=011000 thì b_1b_6=00 (xác định r=0), $b_2b_3b_4b_5$=1100 (xác định c=12), từ đó ta tìm được phần tử ở vị trí (0,12) --> S1(B1)=0101 (tương ứng với số 5).

$b_2b_3b_4b_5$=1100

b_1b_6=00

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

Hộp S1

- Mỗi xâu xuất 4 bit của các hộp S được đưa vào các Cj tương ứng: Cj = Sj(Bj) (1<=j<=8).

3. Mô tả thuật toán

4. Xâu bit $C = C_1C_2C_3C_4C_5C_6C_7C_8$ có độ dài 32 bit được hoán vị tương ứng với hoán vị cố định P. Kết quả có $P(C)= f(R_i,K_i)$.

Hoán vị P

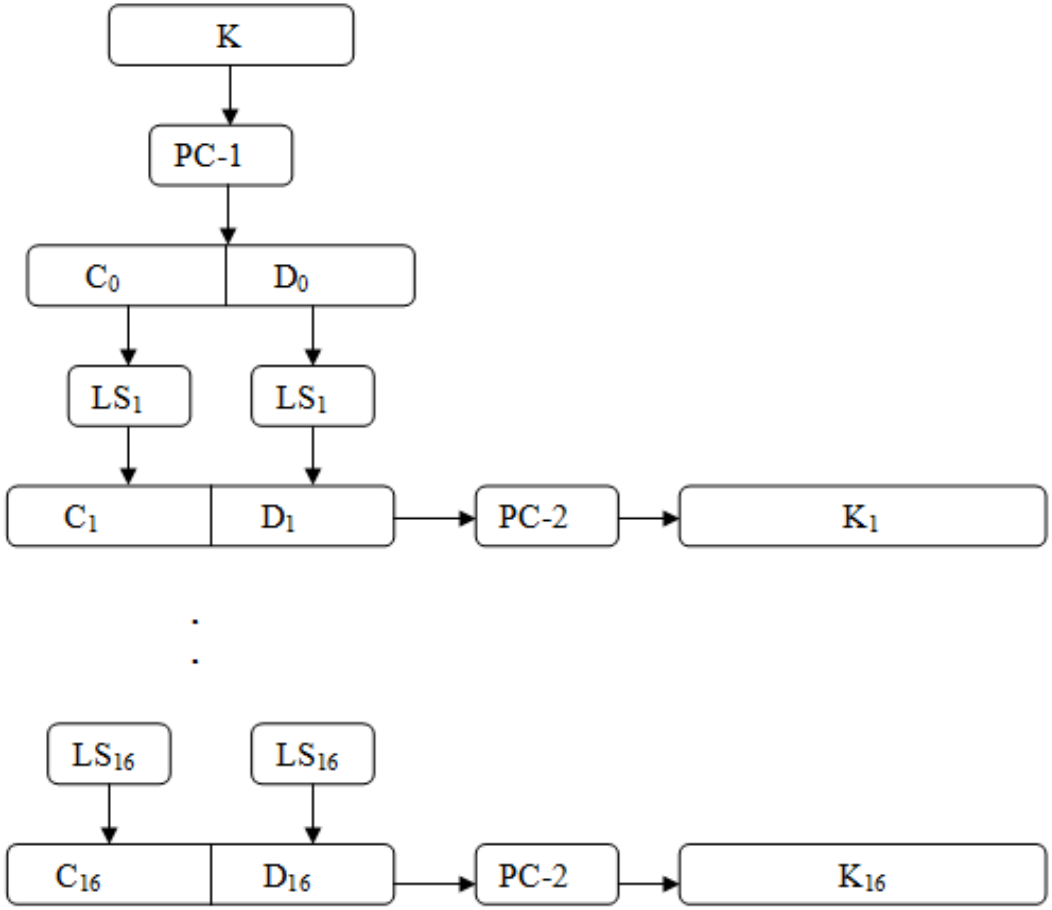
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

3. Mô tả thuật toán

Khóa K

- K là một xâu có độ dài 64 bit trong đó 56 bit dùng làm khóa và 8 bit dùng để kiểm tra sự bằng nhau (phát hiện lỗi).
- Các bit ở các vị trí 8, 16,…, 64 được xác định, sao cho mỗi byte chứa số lẻ các số 1, vì vậy từng lỗi có thể được phát hiện trong mỗi 8 bit.

3. Mô tả thuật toán



Sơ đồ tính khóa $k_1, k_2, \ldots k_{16}$

3. Mô tả thuật toán

Quá trình tạo các khóa con (subkeys) từ khóa K được mô tả như sau:

Cho khóa K 64 bit, loại bỏ các bit kiểm tra và hoán vị các bit còn lại của K tương ứng với hoán vị cố định PC-1. Ta viết PC1(K) = C_0D_0, với C_0 bao gồm 28 bít đầu tiên của PC-1(k) và D_0 là 28 bit còn lại.

Trong đó bảng số bít dịch trái tại mỗi vòng là:

Vòng i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Số bít dịch	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

3. Mô tả thuật toán

Các hoán vị cố định PC-1 và PC-2:

Bảng trật tự khoá (PC-1):

57	49	41	33	25	17	9	1	58	50	42	34	26	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

Bảng trật tự nén(PC-2):

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

Giải mã

- Việc giải mã dùng cùng một thuật toán như việc mã hoá.
- Để giải mã dữ liệu đã được mã hoá, quá trình giống như mã hoá được lặp lại nhưng các chìa khoá phụ được dùng theo thứ tự ngược lại từ K_{16} đến K_1, nghĩa là trong bước 2 của quá trình *mã hoá dữ liệu đầu vào* ở trên R_{i-1} sẽ được XOR với K_{17-i} chứ không phải với K_i.

Ví dụ mã hóa

Mã hóa bản rõ sau trong dạng thập lục phân (Hexadecimal)

0123456789ABCDEF

Sử dụng khóa thập lục phân ***133457799BBCDFF1***

L_0	=	11001100000000001100110011111111
$L_1 = R_0$	=	11110000101010101111000010101010

$E(R_0)$	=	011110100001010101010101011110100001010101010101
K_1	=	000110110000001011101111111111000111000001110010
$E(R_0) \oplus K_1$	=	011000010001011110111010100001100110010100100111
Output S-hộp	=	01011100100001010110101100101 11
$f(R_0,K_1)$	=	00100011010010101010100110111011
$L_2 = R_1$	=	11101111010010100110010101000100

$E(R_1)$	=	011101011110101001010100001100001010101000001001

K_2	=	011110011010111011011001110110111100100111100101
$E(R_1) \oplus K_2$	=	000011000100010010001101111010110110001111101100
Output S-hộp	=	11111000110100000011101010101110
$f(R_1, K_2)$	=	00111100101010111000011110100011
$L_3 = R_2$	=	11001100000000010111011100001001

$E(R_2)$	=	111001011000000000000010101110101110100001010011
K_3	=	010101011111110010001010010000101100111110011001
$E(R_2) \oplus K_3$	=	101100000111110010001000111110000010011111001010
S-box output	=	00100111000100001110000101101111
$f(R_2, K_3)$	=	01001101000101100110111010110000
$L_4 = R_3$	=	10100010010111000000101111110100

$E(R_3)$	=	010100000100001011111000000001010111111110101001
K_4	=	011100101010110111010110110110110011010100011101
$E(R_3) \oplus K_4$	=	001000101110111100101110110111100100101010110100
S-box output	=	00100001111011011001111100111010
$f(R_3, K_4)$	=	10111011001000110111011101001100
$L_5 = R_4$	=	011101110

$E(R_4)$	=	101110101110100100000100000000000000000001000001010
K_5	=	011111001110110000000111111010110101001110101000
$E(R_4) \oplus K_5$	=	110001100000010100000011111010110101000110100010
Xuất S-hộp	=	01010000110010000011000111101011
$f(R_4, K_5)$	=	00101000000100111010110111000011
$L_6 = R_5$	=	10001010010011111010011000110111

$E(R_5)$	=	110001010100001001011111110100001100000110101111
K_6	=	011000111010010100111110010100000111101100101111
$E(R_5) \oplus K_6$	=	101001101110011101100001100000000101110101000000
S-box output	=	010000011111001101001100000111101
$f(R_5, K_6)$	=	100111100100010111001101001011 00
$L_7 = R_6$	=	111010010110011111001101011 01001

$E(R_6)$	=	111101010010101100001111111001011010101101010011
K_7	=	111011001000010010110111111101100001100010111100
$E(R_6) \oplus K_7$	=	000110011010111110111000000100111011001111101111
S-box output	=	000100000111010101000000010101101
$f(R_6, K_7)$	=	100011000000010100011100001001 11
$L_8 = R_7$	=	000001100100101010111010000 10000

$E(R_7)$	=	000000001100001001010101010111110100000010100000

K_8	=	111101111000101000111010110000010011101111111011
$E(R_7) \oplus K_8$	=	111101110100100001101111100111100111101101011011
S-box output	=	011011000001100001111110010101110
$f(R_7, K_8)$	=	001111000000111010000110111111001
$L_9 = R_8$	=	110101010110100101001011100 10000

$E(R_8)$	=	011010101010101101010010101001010111110010100001
K_9	=	111000001101101111101011111011011110011110000001
$E(R_8) \oplus K_9$	=	100010100111000010111001010010001001101100100000
S-box output	=	00010001000011000101011101110111
$f(R_8, K_9)$	=	00100010001101100111110001101010
$L_{10} = R_9$	=	00100100011111001100011001111010

$E(R_9)$	=	000100001000001111111001011000001100001111110100
K_{10}	=	101100011111001101000111101110100100011001001111
$E(R_9) \oplus K_{10}$	=	101000010111000010111110110110101000010110111011
S-box output	=	11011010000001000101001001110101
$f(R_9, K_{10})$	=	01100010101111001001110000100010
$L_{11} = R_{10}$	=	10110111110101011101011110110010
$E(R_{10})$	=	010110101111111010101011111010101111110110100101
K_{11}	=	001000010101111111010011110111101101001110000110
$E(R_{10}) \oplus K_{11}$	=	011110111010000101111000001101000010111000100011
S-box output	=	01110011000001011101000100000001
$f(R_{10}, K_{11})$	=	11100001000001001111101000000010
$L_{12} = R_{11}$	=	11000101011110000011110001111000

$E(R_{11})$	011000001010101111110000000111111000001111110001
K_{12}	011101010111000111110101100101000110011111101001
$E(R_{11}) \oplus K_{12}$	000101011101101000000101100010111110010000011000
S-box output	011110111000101100100110001101 01
$f(R_{11}, K_{12})$	110000100110100011001111111101010
$L_{13} = R_{12}$	011101011011110100011000010110 00

$E(R_{12})$	=	001110101011110111111010100011110000001011110000
K_{13}	=	100101111100010111010001111110101011101001000001
$E(R_{12}) \oplus K_{13}$	=	101011010111100000101011011101011011100010110001
S-box output	=	100110101101000110001011010011 11
$f(R_{12}, K_{13})$	=	110111011011101100101001001000 10
$L_{14} = R_{13}$	=	000110001100001100010101010110 10

$E(R_{13})$	=	000011110001011000000110100010101010101011110100
K_{14}	=	010111110100001110110111111100101110011100111010
$E(R_{13}) \oplus K_{14}$	=	010100000101010110110001011110000100110111001110

S-box output	=	01100100011110011001101011110001
$f(R_{13}, K_{14})$	=	10110111001100011000111001010101
$L_{15} = R_{14}$	=	11000010100011001001011000001101

$E(R_{14})$	=	11100000010101000101100101001010110000001011011
K_{15}	=	10111111100100011000110100111101001111100001010
$E(R_{14}) \oplus K_{15}$	=	01011111110001011101010001110111111111101010001
S-box output	=	10110010111010001000110100111100
$f(R_{14}, K_{15})$	=	01011011100000010010011101101110
$L_{16} = R_{15}$	=	01000011010000100011001000110100

$E(R_{15})$	=	00100000011010100000010000011010010000011010100 0
K_{16}	=	11001011001111011000101100001110000101111110101
$E(R_{15}) \oplus K_{16}$	=	11101011010101111000111100010100010101100101110 1
S-box output	=	10100111100000110010010000101001
$f(R_{15}, K_{16})$	=	11001000110000000100111110011000
R_{16}	=	00001010010011001101100110010101

Ví dụ mã hóa

Kết luận:

Cuối cùng áp dụng IP-1 cho R16L16 ta nhận được bản rõ trong dạng thập lục phân sau: ***85E813540F0AB405***

4. Đặc điểm của hệ mã DES

4.1 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

4. Đặc điểm của hệ mã DES

4.2 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.

4.3 Khóa yếu

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

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

=>**V**iệc 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	$\{1\}^{28}$	$\{0\}^{28}$

4.3 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	$\{1\}^{28}$	$\{10\}^{14}$

5. Triple DES (3DES)

-Hệ mã DES (hay chuẩn mã hóa dữ liệu) với không gian khóa có 2^{55} khóa nên thực tế hiện nay có thể bị thám mã trong một khoảng thời gian ngắn.

-Tìm kiếm một hệ mã mới là điều tất yếu.

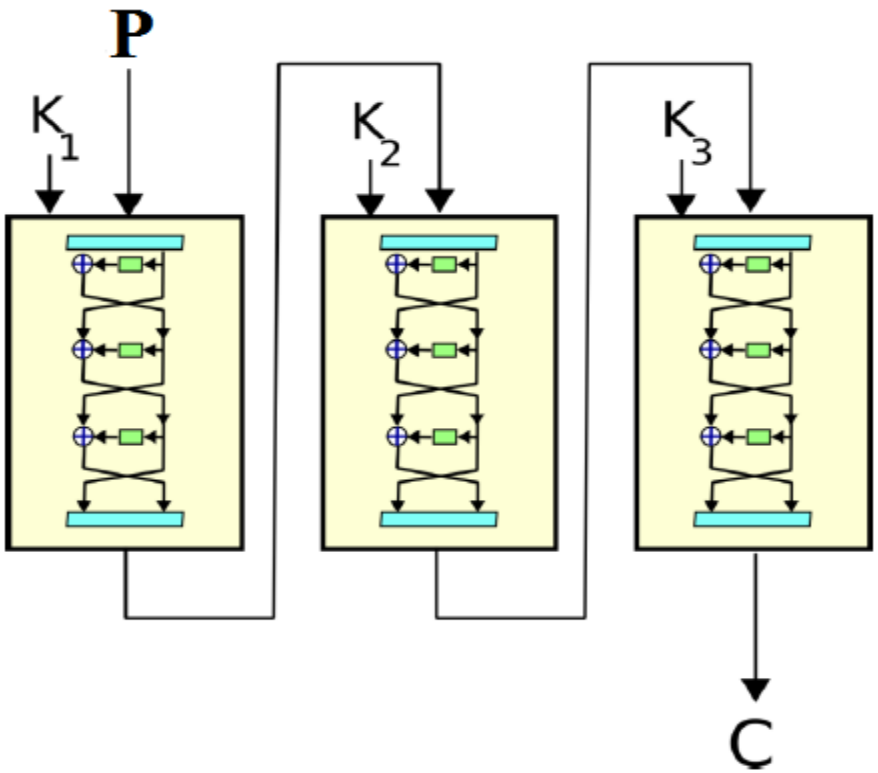
-Tận dụng hệ mã DES, chúng ta đưa ra một phương pháp mã hóa mới bằng cách mã hóa nhiêu lần.

-Chúng ta sử dụng 2 khóa để mã hóa hai lần $C = E_{K2}(E_{K1}(P))$

-Cách này gọi là double DES hay 2DES, khóa của hệ mã theo mô hình này là 112 bit tuy nhiên về mặt lý thuyết thì hệ mã này không an toàn hơn DES.

5. Triple DES (3DES) (tt)

-Chúng ta có thể sử dụng thuật toán DES với 3 khóa để mã hóa được gọi là Triple DES (TDES) hay 3DES



5. Triple DES (3DES) (tt)

-Bản mã C = $DES_{K3}(DES_{K2}(DES_{K1}(M))$, mô hình này gọi là EEE,

-Một biến thể khác của mô hình này gọi là EDE với bước ở giữa sử dụng thuật toán giải mã của DES:

$$C = DES_{K3}(DES_{K2}^{-1}(DES_{K1}(M)).$$

-Khóa của Triple DES là 168 bit. Các chứng minh về mặt lý thuyết và các tấn công đối với Triple DES cho thấy hệ mã này vẫn sẽ còn được sử dụng trong một tương lai dài

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.**

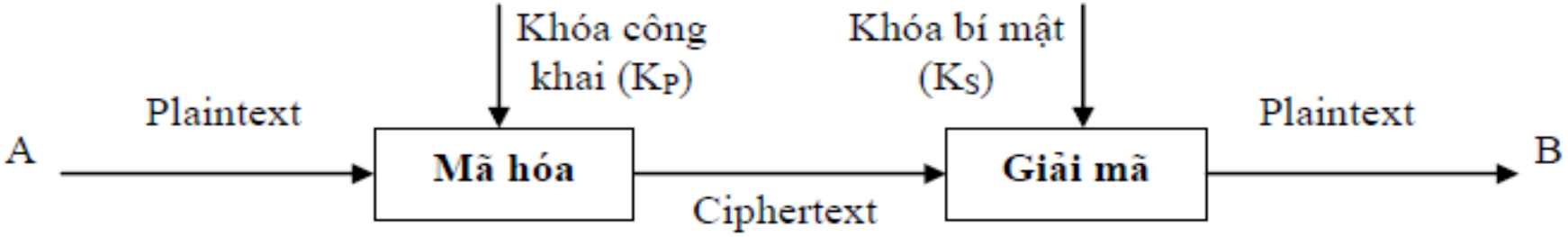
2. Ý tưởng của Diffie & Hellman

- 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)

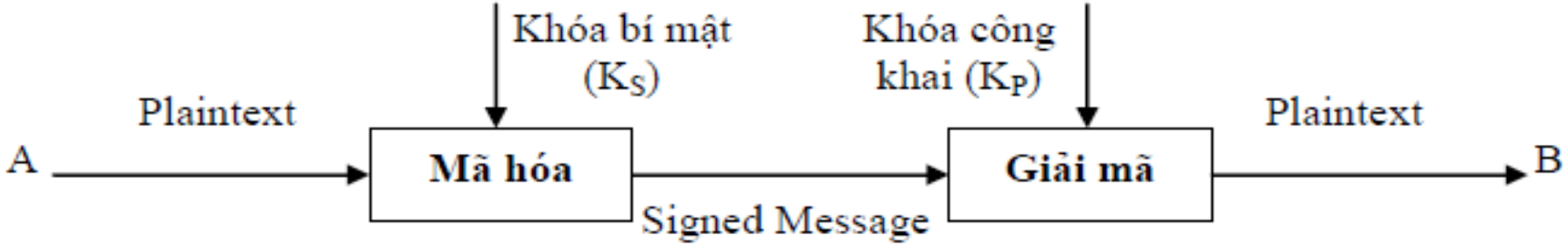
- Các yêu cầu của loại hệ mã này:
  - Việc sinh K_S,K_P phải dễ dàng
  - Việc tính $E(K_P, M)$ là dễ dàng
  - Nếu có $C = E(K_P, M)$ và K_S thì việc tìm bản rõ cũng là dễ
  - Nếu biết K_P thì việc dò tìm K_S là khó
  - Vệc giải mã bản rõ từ mã là rất khó
- Khi A muốn truyền tin cho B, A sẽ sử dụng khóa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

Mô hình (2) được sử dụng cho các hệ chử ký điện tử còn mô hình (1) được sử dụng cho các hệ mã mật

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 D_z, 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.

Bài tập 4.9: Cho hệ mã Knapsack có A = {11, 15, 30, 60}, M = 150 và u = 77.

a) Hãy tìm khóa công khai K_P, và khóa bí mật K_S của hệ mã trên.

b) Để mã hóa các thông điệp viết bằng tiếng Anh người ta dùng một hàm chuyển đổi từ các ký tự thành các xâu nhị phân như sau:

Ký tự	Xâu bít	Ký tự	Xâu bít	Ký tự	Xâu bít	Ký tự	Xâu bít
A	00000	H	00111	O	01110	V	10101
B	00001	I	01000	P	01111	W	10110
C	00010	J	01001	Q	10000	X	10111
D	00011	K	01010	R	10001	Y	11000
E	00100	L	01011	S	10010	Z	11001
F	00101	M	01100	T	10011		
G	00110	N	01101	U	10100		

Khi đó ví dụ xâu ABCD sẽ được chuyển thành 00000 00001 00010 00011 và cắt thành các xâu có độ dài 4 để thực hiện mã hóa. Kết quả thu được bản mã là một dãy các số $\in Z_M$. Hãy thực hiện mã hóa xâu P = "ANTI".

c) Giả sử bản mã thu được là C = <120, 105, 105, 0, 60, 75, 30, 22, 22, 30>. Hãy thực hiện giải mã bản mã trên để thu được thông điệp ban đầu.

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 cần 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:

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

  - $\phi(n)$ 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

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

khi đó ta sẽ có điều cần thiết:

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

- Tóm lại: Nếu đã biết e và
  - Biết PTTSNT của n → tìm được $d = e^{-1} \pmod{\phi(n)}$ tức $X^{ed} = X \pmod{n}$
  - 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 $\varphi(n) = (p-1).(q-1)$ (hàm phi Euler)
4. Chọn số e sao cho $1 < e < \varphi(n)$ và là số nguyên tố cùng nhau với $\varphi(n)$ (tức là: $\gcd(e, \varphi(n)) = 1$, để đảm bảo $e^{-1} \bmod \varphi(n)$ tồn tại duy nhất)
5. Tính số $d = e^{-1} \bmod \varphi(N)$

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

Bài tập

Cho hệ mã RSA có p = 29, q = 43, e = 11.

a) Hãy tìm khóa công khai K_P, và khóa bí mật K_S của hệ mã trên.

b) Để mã hóa các thông điệp được viết bằng tiếng Anh người ta dùng một hàm chuyển đổi các ký tự thành các số thập phân có hai chữ số như sau:

Ký tự	A	B	C	D	E	F	G	H	I	J	K	L	M
Mã hóa	00	01	02	03	04	05	06	07	08	09	10	11	12
Ký tự	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Mã hóa	13	14	15	16	17	18	19	20	21	22	23	24	25

Khi đó ví dụ xâu ABC sẽ được chuyển thành 00 01 02 và sau đó cắt thành các số có 3 chữ số 000 (bằng 0) và 102 để mã hóa. Bản mã thu được là một tập các số $\in Z_N$. Hãy thực hiện mã hóa xâu P = "TAURUS".

c) Giả sử bản mã thu được là C = <1, 169, 1206, 433> hãy thực hiện giải mã để tìm ra thông điệp bản rõ ban đầu.

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 **K** = y^k mod p
  3. Sau đó tính ***cặp ciphertext*** **{C_1, C_2}:**
     - **C_1** = a^k mod p
     - **C_2** = **K** . M mod p
- Như vậy, M đã được mã hóa thành {C_1, C_2}:

$$\mathbf{M \rightarrow \{C_1, C_2\}}$$

- 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}:
    - $\mathbf{C_1} = 5^{36}$ = **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

So sánh thời gian và chi phí cần thiết để phá các khóa có độ dài tương ứng.

Giá (USD)	Độ dài khoá (bit)				
	40	56	64	80	128
100 nghìn	2 giây	35 giờ	1 năm	70000 năm	10^{19} năm
1 triệu	2 giây	3,5 giờ	37 năm	7000 năm	10^{18} năm
100 triệu	2 mili giây	2 phút	9 giờ	70 năm	10^{16} năm
1 tỷ	2 mili giây	13 giây	1 giờ	7 năm	10^{15} năm
100 tỷ	2 micro giây	1 giây	32 giây	24 ngày	10^{13} năm

So sánh độ dài các khoá trong hai hệ mã hoá đối xứng và bất đối xứng với cùng độ an toàn

Độ dài khóa đối xứng (bit)	Độ dài khóa bất đối xứng (bit)
56	384
64	512
80	768
112	1792
128	2034

Mã đối xứng và mã bất đối xứng

Mã đối xứng

- Tốc độ xử lý nhanh
- Mã khóa ngắn
- Khó trao đổi mã khóa

Mã bất đối xứng

- Tốc độ xử lý chậm
- Mã khóa dài
- Trao đổi mã khóa dễ dàng