BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

ĐỖ VĂN DŨNG

NGHIÊN CỨU ĐỀ XUẤT THUẬT TOÁN MÃ HÓA VĂN BẢN CÓ ĐỘ BẢO MẬT CAO TRÊN CƠ SỞ MẬT MÃ TRUYỀN THỐNG

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH HỆ THỐNG THÔNG TIN
MÃ SỐ: 60 48 01 04

NGƯỜI HƯỚNG DẪN KHOA HỌC:
TS. HỒ VĂN CANH

LỜI CAM ĐOAN

Tôi cam đoan luận văn này là do bản thân tự nghiên cứu và thực hiện theo sự hướng dẫn khoa học của ***TS. Hồ Văn Canh***

Tôi hoàn toàn chịu trách nhiệm về tính pháp lý quá trình nghiên cứu khoa học của luận văn này.

Hải Phòng, ngày tháng 10 năm 2017

Người Cam đoan.

Đỗ Văn Dũng

LỜI CẢM ƠN

Trước tiên tôi bày tỏ lời cảm ơn chân thành đến các Thầy, cô giáo đã giảng dạy, hướng dẫn và giúp đỡ tôi trong thời gian học tập và nghiên cứu hoàn thành luận văn này.

Xin được bày tỏ lòng biết ơn sâu sắc tới Thầy giáo ***TS Hồ Văn Canh*** đã tận tình hướng dẫn, giúp đỡ và đóng góp cho tôi nhiều ý kiến quý báu để hoàn thành luận văn này.

Xin chân thành cảm ơn các Thầy, Cô giáo Trường Đại Học Dân Lập Hải Phòng , đặt biệt là các thầy cô trong khoa CNTT đã giảng dạy, giúp đỡ và tạo điều kiện thuận lợi cho tôi trong thời gian học tập tại Trường.

Cuối cùng, xin chân thành cảm ơn gia đình và bạn bè đã động viên, quan tâm, giúp đỡ tôi hoàn thành khóa học và luận văn.

MỤC LỤC

MỞ ĐẦU

Ngày nay trong mọi hoạt động của con người thông tin đóng một vai trò quan trọng không thể thiếu. Xã hội càng phát triển nhu cầu trao đổi thông tin giữa các thành phần trong xã hội ngày càng lớn. Mạng máy tính ra đời đã mang lại cho con người rất nhiều lợi ích trong việc trao đổi và xử lý thông tin một cách nhanh chóng và chính xác. Chính từ những thuận lợi này đã đặt ra cho chúng ta một câu hỏi, liệu thông tin đi từ nơi gửi đến nơi nhận có đảm bảo tuyệt đối an toàn, ai có thể đảm bảo thông tin của ta không bị truy cập bất hợp pháp. Thông tin được lưu giữ, truyền dẫn, cùng sử dụng trên mạng lưới thông tin công cộng có thể bị nghe trộm, chiếm đoạt, xuyên tạc hoặc phá huỷ dẫn đến sự tổn thất không thể lường được. Đặc biệt là đối với những số liệu của hệ thống ngân hàng, hệ thống thương mại, cơ quan quản lý của chính phủ hoặc thuộc lĩnh vực quân sự được lưu giữ và truyền dẫn trên mạng.

Các kỹ thuật đảm bảo an toàn thông tin cho thông tin liên lạc số được chia thành 2 loại. Đó là mật mã (Cryptography), giấu tin mật (Steganography) và thủy phân số (Watermarking). Mỗi loại có những ứng dụng và mục tiêu khác nhau nhưng đề đảm bảo an toàn cho việc truyền tin mật trên kênh không an toàn.

Các kỹ thuật Cryptography và Steganography nói chung được dùng để truyền những thông tin nhạy cảm giữa hai hay nhiều thực thể trong cùng một nhóm với nhau. Tuy nhiên giữa chúng có những sự khác nhau.

Cryptography sử dụng những phép biến đổi toán học để mã hóa bản thông điệp, biến mỗi thông điệp đọc được có nghĩa thành một dãy giả ngẫu nhiên, mà người ta gọi là bản mã, để truyền trên mạng công cộng đến người nhận có chủ đích. Đó là khi hai người thí dụ như la người A và B liên lạc với nhau thì mặc dù người C không đọc được nội dung thông tin nhưng người C rõ ràng là biết giữa hai người A và B có ý đồ ‘đen tối’ nào đó.

Ngược lại, với Steganography thì người C không thể biết giữa hai người A và B đang có sự liên lạc truyền thông tin mật cho nhau. Để đảm bảo được điều này, hai người A và B sử dụng một vật trung gian số ở đây là đa phương tiện số (Multimedia) cụ thể như: audio, video, hoặc images…

Con thủy vân số (Watermarking) về nguyên lý tương tự như Steganography nhưng có khác nhau về mục đích ứng dụng. Mục tiêu của Watermarking là những thông tin được nhúng trong ảnh phải đảm bảo sao cho Watermarking không thể bị dịch chuyển mà không pháp hủy chính ảnh mang tin đó. Watermarking thường được ứng dụng trong các lĩnh vực như bảo vệ bản quyền.

Để đảm bảo được mức độ an toàn cao, trước khi giấu tin vào các Multimedia, người ta đã mã hóa dữ liệu cần giấu đó bằng các thuật toán mã hóa truyền thống. Do tầm quan trọng như vậy nên em đã chọn đề tài "Nghiên cứu đề xuất thuật toán mã hóa văn bản có độ bảo mật cao trên cơ sở mật mã truyền thống".

Nội dung của luận văn gồm ba chương và phần kết luận.

Chương 1: Tổng quan về các hệ mật mã.

- Chương này giới thiệu một số thông tin tổng quan về các hệ mật mã, trình bày các lý thuyết mật mã, mật mã truyền thống, mật mã khóa công khai.

Chương 2: Một số phương pháp tấn công hệ mật mã truyền thống.

- Chương này giới thiệu một số phương pháp tấn công hệ mật mã truyền thống. Trên cơ sở đó, học viên đưa ra một số nhược điểm của hệ mật mã truyền thống.

Chương 3: Đề xuất thuật toán nhằm nâng cao độ an toàn cho hệ mật mã truyền thống.

- Chương này sẽ dựa trên cơ sở đã nghiên cứu ở chương 2 để đưa ra thuật toán nâng cao độ an toàn. So sánh thuật toán cũ và mới để thấy được độ an toàn bảo mật của văn bản đã được mã hóa? Đề xuất thuật toán, xây dựng thuật toán, cài đặt thuật toán và thử nghiệm. Đánh giá kết quả, hướng nghiên cứu tiếp và kết luận.

CHƯƠNG 1: TỔNG QUAN VỀ CÁC HỆ MẬT MÃ

1.1. Tổng quan về lý thuyết mật mã.

1.1.1. Một số khái niệm cơ bản.

a, Các mô hình mã hóa có chung một số thuật ngữ như sau:

- Bản rõ: Là nội dung của thông điệp cần gửi đi và cần được bảo vệ an toàn. Nó có thể là xâu các bít, các file văn bản, các file có cấu trúc.
- Mã hoá: Là quá trình biến đổi bản rõ thành những dãy ký tự không đọc được có nghĩa trước khi gửi đến người nhận đích thực..
- Bản mã: Là kết quả thu được khi mã hóa bản rõ theo một thuật toán mã hóa nào đó.
- Giải mã: Là quá trình xử lý ngược, tiến hành giải mã bản mã để thu lại bản rõ. Ví dụ: Mã hóa văn bản có nội dung là "ABC" với luật mã là tịnh tiến vòng 1 đơn vị đối với mã ASCII của mỗi kí tự.

Vậy ta có:

Bản rõ: "ABC"

Mã hóa: Thực hiện mã hóa theo luật mã.

Biến đổi các kí tự thành các số theo mã ASCII của kí tự đó.

A ↔ 65, B ↔ 66, C ↔ 67

Thu được các mã mới sau khi tịnh tiến là: 66 – 67 – 68 Biến đổi các mã mới thành kí tự.

Bản mã: "BCD".

Giải mã: Thu được bản rõ là "ABC".

b, Hệ mật mã.

Hệ mật mã là một bộ gồm 5 thành phần (P, C, K, E, D),trong đó,

P (Plaintext): là tập hợp hữu hạn các bản rõ.

C (Ciphertext): là một tập hữu hạn các bản mã.

K (Key): là một tập hữu hạn các khóa có thể.

E (Encrytion): là tập các hàm lập mã.

D (Decrytion): là tập các hàm giải mã.

Chúng ta đã biết một thông báo thường được xem là bản rõ. Người gửi sẽ có nhiệm vụ mã hóa bản rõ đó bằng một thuật toán mã hóa nào đó để cho ra kết quả được gọi là bản mã. Và bản mã này sẽ được gửi đi trên đường truyền không an toàn tới người nhận. Người nhận giải mã bản mã để tìm hiểu nội dung của bản rõ.

Với mỗi $k \in K$, có một hàm lập mã $e_k \in E$, $e_k : P \rightarrow C$, và một hàm giải mã $d_k \in D$, $d_k : C \rightarrow P$ sao cho: $d_x (e_k(x)) = x, \quad \forall x \in P$

c, Những tính năng của hệ mật mã.

- Cung cấp một mức cao về tính bảo mật, toàn vẹn, chống chối bỏ và xác thực.
- Tính bảo mật: Bảo đảm bí mật cho nội dung thông báo và dữ liệu bằng nhờ các kỹ thuật mã hóa.
- Tính toàn vẹn: Bảo đảm với các bên rằng bản tin không bị thay đổi trên đường truyền tin.
- Chống chối bỏ: Có thể xác nhận rằng tài liệu đã đến từ ai đó, ngay cả khi họ cố gắng từ chối nó.
- Tính xác thực: Cung cấp hai dịch vụ:

➢ Nhận dạng nguồn gốc của một thông báo, đảm bảo rằng nó là đúng sự thực.

➢ Kiểm tra định danh của người đang đăng nhập hệ thống, tiếp tục kiểm tra đặc điểm của họ trong trường hợp ai đó cố gắng kết nối và giả danh là người sử dụng hợp pháp.

1.1.2. Cơ sở toán học của lý thuyết số.

a, Tính chia hết của các số nguyên, thuật toán Euclide [3].

Ta ký hiệu Z là tập hợp các số nguyên, Z = {.....,-2,-1,0,1,2,....}, và Z^+ là tập hợp các số nguyên không âm, Z^+= {0,1,2,.....}.

- Tính chia hết của số nguyên

Tập hợp Z là đóng kín đối với các phép cộng, trừ và nhân, nhưng không đóng kín đối với phép chia: chia một số nguyên cho một số nguyên không phải bao giờ cũng được kết quả là một số nguyên. Vì vậy, trường hợp chia hết, tức khi chia số nguyên a cho số nguyên b được thương là một số nguyên q, $a = b.q$, có một ý nghĩa đặc biệt. Khi đó, ta nói a chia hết cho b, b chia hết bởi a, a là bội số của b, b là ước số của a, và ký hiệu là $b \mid a$. Dễ thấy ngay rằng số 1 là ước số của mọi số nguyên bất kỳ, số 0 là bội số của mọi số nguyên bất kỳ, mọi số nguyên a là ước số, đồng thời là bội số, của chính nó.

Cho hai số nguyên bất kỳ a và b, $b > 1$. Thực hiện phép chia a cho b ta sẽ được hai số q và r sao cho

$$a = b \,.\, q + r \,,\ 0 \leq r < b \,.$$

Số q được gọi là số thương của phép chia a cho b, ký hiệu a div b, và số r được gọi là số dư của phép chia a cho b, ký hiệu a mod b.

Thí dụ: 25 div 7 = 3 và 25 mod 7 = 4, -25 div 7 = -4 và -25 mod 7 = 3.

Một số nguyên d được gọi là ước số chung của hai số nguyên a và b nếu $d \mid a$ và $d \mid b$. Số nguyên d được gọi là ước số chung lớn nhất của a và b nếu $d > 0$, d là ước số chung của a và b, và mọi ước số chung của a và b đều là bế hơn hay bằng d. Ta ký hiệu ước số chung lớn nhất của a và b là gcd(a, b). Thí dụ gcd(12, 18) = 6, gcd(-18, 27) = 3.

Dễ thấy rằng với mọi số nguyên dương a ta có gcd(a, 0) = a, ta cũng sẽ qui ước xem rằng gcd(0, 0) = 0.

Định lý 1.1.2: *Nếu $b \neq 0$ và $b \mid a$ thì gcd(a, b) = b.*

Nếu $a = b \cdot q + r$ *thì* $gcd(a, b) = gcd(b, r)$.

Một số nguyên m được gọi là bội số chung của a và b nếu $a \mid m$ và $b \mid m$. Số m được gọi là bội số chung nhỏ nhất của a và b , và được ký hiệu là lcm(a , b), nếu m là bội số chung của a và b và mọi bội số chung của a và b đều lớn hơn hoặc bằng m . Thí dụ lcm(14,21) = 42.

Với hai số nguyên dương a và b bất kỳ ta có quan hệ

$$\text{lcm}(a, b) \cdot \gcd(a, b) = a \cdot b.$$

Từ định lý 1.1.2 ta suy ra thuật toán sau đây thực hiện việc tìm ước số chung lớn nhất của hai số nguyên bất kỳ:

Thuật toán Euclide tìm ước số chung lớn nhất:

INPUT: hai số nguyên không âm a và b , với $a \geq b$.

OUTPUT: ước số chung lớn nhất của a và b.

1. Trong khi còn $b > 0$, thực hiện:

1.1. đặt $r \leftarrow a \bmod b$, $a \leftarrow b$, $b \leftarrow r$.

2. Cho ra kết quả (a).

Thí dụ: Dùng thuật toán Euclide tìm gcd(18, 12), ta lần lượt được các giá trị gán cho các biến a, b và r như sau:

$18 = 1 \cdot 12 + 6$

$12 = 2 \cdot 6 + 0$

a	b	r
18	12	
12	6	6
6	0	0

Thuật toán Euclide mở rộng: Thuật toán Euclide mở rộng. Thuật toán này nhằm xác định 3 số nguyên x, y, d sao cho: $mx + ny = d$, trong đó m, n là hai số nguyên cho trước với giả thiết $m \geq n$. Nội dung thuật toán như sau: Cho 3 véc - tơ ($a1$, $a2$, $a3$), ($b1$, $b2$, $b3$), ($c1$, $c2$, $c3$); Các bước tiến hành như sau:

Bước1. ($a1$, $a2$, $a3$) ← (1, 0, m), ($b1$, $b2$, $b3$) ← (0, 1, n);

Bước 2. Nếu $b3$=0 thì thuật toán dừng và ($a1$, $a2$, $a3$) là đáp số;

Bước 3. Đặt q = [$a3$/ $b3$]; và ($c1$, $c2$, $c3$) ← ($a1$, $a2$, $a3$) -q($b1$, $b2$, $b3$); ($a1$, $a2$, $a3$) ← ($b1$, $b2$, $b3$); ($b1$, $b2$, $b3$) ←($c1$, $c2$, $c3$) và đi đến bước 2. Trong đó [X] là phần nguyên của số X, nghĩa là [X] là số nguyên lớn nhất nhưng không vượt quá X.

Thí dụ: Dùng thuật toán Euclide mở rộng cho các số a= 4864 và b= 3458, ta lần lượt được các giá trị sau đây cho các biến a, b, q, r, x, y, x_1 , x_2 , y_1 , y_2 (sau mỗi chu trình thực hiện hai lệnh 3.1 và 3.2):

a	b	q	r	x	y	x_1	x_2	y_1	y_2
4864	3458					0	1	1	0
3458	1406	1	1406	1	-1	1	0	-1	1
1406	646	2	646	-2	3	-2	1	3	-1
646	114	2	114	5	-7	5	-2	-7	3
114	76	5	76	-27	38	-27	5	38	-7
76	38	1	38	32	-45	32	-27	-45	38
38	0	2	0	-91	128	-91	32	128	-45

Ta dễ thử lại rằng sau mỗi lần thực hiện chu trình gồm hai lệnh 3.1 và 3.2, các giá trị x, y, r thu được luôn thoả mãn $4864 \cdot x + 3458 \cdot y = r$, và do đó khi kết thúc các vòng lặp (ứng với giá trị b= 0), thực hiện tiếp lệnh 4 ta được kết quả d = 38, x = 32 và y= -45, cặp số (32, -45) thoả: $4864 \cdot 32 + 3458 \cdot (-45) = 38$.

b, Số nguyên tố và nguyên tố cùng nhau.

Số nguyên tố là số nguyên dương chỉ chia hết cho 1 và chính nó.

Thí dụ: 2, 3, 5, 7, 11, 17, ...

Hệ mật mã thường sử dụng các số nguyên tố ít nhất là lớn hơn 10^{150}.

Hai số m và n được gọi là nguyên tố cùng nhau, nếu ước số chung lớn nhất của chúng bằng Ký hiệu: gcd (m, n) = 1.

Thí dụ: 9 và 14 là hai số nguyên tố cùng nhau.

c, Đồng dư thức.

- Cho a và b là các số nguyên n là số nguyên dương. Khi đó a được gọi là đồng dư với b theo modulo n, ký hiệu là $a \equiv b \pmod{n}$, nếu a, b chia cho n có cùng số dư. n được gọi là modulo của đồng dư.

Kí hiệu: $a \equiv b \pmod{n}$

Thí dụ: $11 \equiv 5 \pmod 3$ vì 11 và 5 khi chia cho 3 đều dư số dư là 2.

- Tính chất đồng dư

Cho $a, a_1, b, b_1, c \in Z$. Ta có các tính chất sau:

$a \equiv b \bmod n$ nếu và chỉ nếu a và b có cùng số dư khi chia cho n

Tính phản xạ: $a \equiv a \bmod n$

Tính đối xứng: Nếu $a \equiv b \bmod n$ thì $b \equiv a \bmod n$

Tính giao hoán: Nếu $a \equiv b \bmod n$ và $b \equiv c \bmod n$ thì $a \equiv c \bmod n$

Nếu $a \equiv a_1 \bmod n$, $b \equiv b_1 \bmod n$ thì $a + b \equiv (a_1 + b_1) \bmod n$ và $a \cdot b \equiv (a_1 \cdot b_1) \bmod n$

- Lớp tương đương:

Lớp tương đương của số nguyên a là tập hợp các số nguyên đồng dư với a theo modulo n.

Cho $n > 1$ cố định , và a, b là hai số nguyên cho trước. Nếu a - b chia hết cho n , thì ta ký hiệu $a \equiv b \bmod n$. Vì vậy mỗi số nguyên a là đồng dư theo modulo n với duy nhất một số nguyên trong khoảng từ 0 đến $n - 1$ và được gọi là thặng dư nhỏ nhất của a theo modulo n. Cũng vì vậy, a và b cùng thuộc một lớp tương đương. Do đó b có thể đơn giản được sử dụng để thể hiện lớp tương đương theo modulo (n).

d, Không gian Z_n và Z_n*.

- Không gian Z_n (các số nguyên theo modulo n)

Không gian các số nguyên theo modulo n: Z_n là *tập hợp các* số nguyên không âm nhỏ hơn n. Tức là $Z_n = \{0, 1, 2, \ldots n - 1\}$. Tất cả các phép toán trong Z_n đều được thực hiện theo modulo n.

Thí dụ: $Z_{10} = \{0,1,2,3,.., 9\}$

Trong Z_{10} : $6 + 7 = 3$, bởi vì $6 + 7 = 13 \equiv 3 \pmod{10}$.

- Không gian Z_n^*

Là tập hợp các số nguyên $p \in Z_n$, nguyên tố cùng n.

Tức là: $Z_n^* = \{ p \in Z_n \mid \gcd(n, p) = 1\}$, $\varphi(n)$ là số phần tử của Z_n^*

Nếu n là một số nguyên tố thì: $Z_n^* = \{ p \in Z_n \mid 1 \leq p \leq n - 1\}$

Thí dụ: $Z_2 = \{0, 1\}$ thì $Z_2^* = \{1\}$ vì $\gcd(1, 2) = 1$.

e, Phần tử nghịch đảo.

- Định nghĩa: Cho $a \in Z_n$. Nghịch đảo của a theo modulo n là *số nguyên* $x \in Z_n$ sao cho $a \cdot x \equiv 1 \pmod{n}$. Nếu x tồn tại thì đó là giá trị duy nhất $x \in Z_n$, và a được gọi là khả nghịch. Nghịch đảo của a ký hiệu là a^{-1} (đối với phép toán nhân)

- Tính chất:

Cho $a, b \in Z_n$. Phép chia a cho b theo modulo n là tích của a và b theo modulo n, và chỉ được xác định khi b có nghịch đảo theo modulo n.

Cho $a \in Z_n$, a là khả nghịch khi và chỉ khi $\gcd(a, n) = 1$.

Giả sử $d = \gcd(a, n)$. Phương trình đồng dư $a \cdot x = b \bmod n$ có nghiệm x nếu và chỉ nếu d chia hết cho b, trong trường hợp các nghiệm d nằm trong khoảng 0 đến $n - 1$ thì các nghiệm đồng dư theo modulo n/d.

Thí dụ: $4^{-1} = 7 \pmod 9$ vì $4 \cdot 7 \equiv 1 \pmod 9$

g, Hàm φ - Euler.

- Định nghĩa: *Cho* $n \geq 1$. $\varphi(n)$ được định nghĩa là số tất cả các số nguyên trong khoảng từ [1; n] nguyên tố cùng nhau với n vàđược gọi là hàm phi Euler.
- Tính chất:
  - Nếu p là số nguyên tố thì $\varphi(n) = p - 1$.
  - Hàm phi Euler là hàm có tính nhân:

Nếu $(m, n) = 1$ thì $\varphi(m \cdot n) = \varphi(m)\ \varphi(n)$.

  - Nếu $n = p_1^{e1} p_2^{e2} \ldots p_k^{ek}$ trong đó, p_i^{ei} là các thừa số nguyên tố của n với $e_i \geq 1$, thì :

$$\varphi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\ldots\left(1 - \frac{1}{p_n}\right)$$

h, Độ phức tạp tính toán.

- Thuật toán : Một hệ thống chặt chẽ và rõ ràng các chỉ thị nhằm xác định một dãy thao tác trên dữ liệu đầu vào sao cho: Bất kể dữ liệu vào (input) như thế nào, sau một số hữu hạn bước thực hiện các thao tác đã chỉ ra, ta thu được một kết quả (output) mong muốn.
- Đặc trưng của thuật toán: Tính đơn giản, tính dừng, tính đúng đắn, tính phổ dụng, tính khả thi.
- Các thức mô tả thuật toán: Ngôn ngữ tự nhiên, sơ đồ khối, mã giả
- Thuật toán tất định (deterministic): Với hai bộ dữ liệu vào giống nhau, thuật toán tất định sẽ thi hành các mã lệnh giống nhau và cho kết quả giống nhau.
- Thuật toán ngẫu nhiên (randomized): Với hai bộ dữ liệu vào giống nhau, thuật toán ngẫu nhiên có thể thực hiện theo những mã lệnh khác nhau và cho kết quả khác nhau.
- Thuật toán và giải thuật không có sự phân biệt trong thuật ngữ tiếng Anh (Algorithm). Nhưng chúng ta có thể hiểu như sau:

➢ Thuật toán: Cách thức giải quyết bài toán (thuần túy trên mô hình toán học)

➢ Giải thuật: Thuật toán và cách thức cài đặt trên một cấu trúc dữ liệu cụ thể

Thí dụ: Thuật toán tìm kiếm nhị phân có thể cài đặt dễ dàng trên mảng nhưng không cài đặt được trong danh sách nối đơn.

- Đánh giá thuật toán (giải thuật) tương đương với đánh giá mô hình cài đặt thuật toán đó trên một cấu trúc dữ liệu cụ thể.
- Đánh giá giải thuật: Là việc tìm cách đánh giá, ước lượng nguồn tài nguyên cần phải có khi thực hiện chương trình cài đặt giải thuật đó.

Tài nguyên: thời gian, bộ nhớ, số lượng bộ vi xử lý, tốc độ đường truyền mạng…

Đánh giá chương trình	**Đánh giá giải thuật**
Thực hiện sau khi cài đặt chương trình trên một máy cụ thể	Thực hiện trước khi viết chương trình
Thử chạy với một vài bộ dữ liệu cụ thể, đo thời gian thực hiện, lượng bộ nhớ chiếm dụng trong trường hợp cụ thể	Nhằm xác định tính khả thi của giải thuật, chọn thuật toán tốt nhất để cài đặt

- Có nhiều chỉ tiêu để đánh giá giải thuật nhưng phổ biến nhất là đánh giá *thời gian thực hiện giải thuật.*
- Phân tích thời gian thực hiện giải thuật :

➢ Dữ liệu càng lớn → thời gian sử lý càng chậm.

➢ Dữ liệu kích thước n → thời gian thực hiện $T(n)$ là một hàm xác định dương.

➢ Thực hiện trên mô hình máy tính trừu tượng.

➢ Độc lập với phần cứng cụ thể.

Độ phức tạp tính toán:

- Thời gian thực hiện một thuật toán phụ thuộc vào cỡ (size) của dữ liệu vào:

Thí dụ:- Tìm một đối tượng có trong danh sách N phần tử hay không ?

- Sắp xếp một dãy số gồm N số.

- Bài toán người bán hàng cần thăm N địa điểm.

- Trong các dữ liệu vào cùng một cỡ (N), thời gian chạy của thuật toán cũng thay đổi:

Thí dụ: Tìm xem một đối tượng có trong danh sách N phần tử hay không ?

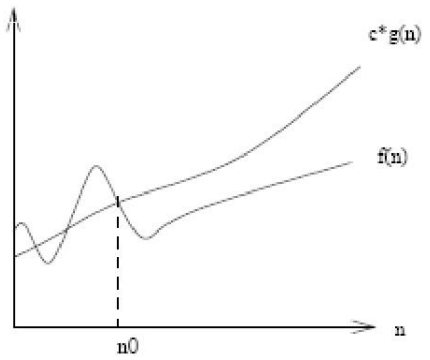
- Đối tượng nằm ở đầu danh sách.

- Đối tượng nằm ở giữa danh sách.

- Đối tượng nằm ở cuối danh sách.

- Biểu diễn thời gian chạy bởi kí hiệu O

Định nghĩa : Giả sử $f(n)$ và $g(n)$ là các hàm thực không âm của đối số nguyên không âm n. Ta nói '$f(n)$ là $O(g(n)$') và viết là : $f(n) = O(g(n))$ nếu tồn tại các hằng số dương c^* và n_0 sao cho $f(n) \leq c^* g(n)$ với mọi $n \geq n_0$.



Thí dụ: Giả sử $f(n) = 5n^3 + 2n^2 + 13n + 6$, ta có :

$f(n) = 5n^3 + 2n^2 + 13n + 6 \leq 5n^3 + 2n^3 + 13n^3 + 6n^3 = 26n^3$

$f(n) = O(n^3)$

Tổng quát, nếu $f(n)$ là một đa thức bậc k của n:

$f(n) = a_k n^k + a_{k-1} n^{k-1} + \ldots + a_1 n + a_0$thì$f(n) = O(n^k)$

Bảng kí hiệu thời gian chạy:

Kí hiệu O lớn	Tên gọi
O(1)	hằng
O($\log n$)	logarit
O(n)	tuyến tính
O($n\log n$)	n$\log n$
O(n^2)	bình phương
O(n^3)	lập phương
O(2^n)	mũ

- Thời gian chạy của các lệnh

➢ Lệnh gán

X = <biểu thức>

Thời gian chạy của lệnh gán bằng thời gian thực hiện biểu thức.

➢ Lệnh lựa chọn

if(điều kiện) →$T_0(n)$

lệnh 1→$T_1(n)$

else

lệnh 2 →$T_2(n)$

Thời gian : $T_0(n)$ + max($T_1(n)$ + $T_2(n)$)

➢ Lệnh lặp : for, while, do – while

Thí dụ : $\sum_i^{X(n)}(T_0(n) + T_i(n))$ với *X(n)* số vòng lặp.

$T_0(n)$ Điều kiện lặp.

$T_i(n)$ Thời gian thực hiện vòng lặp thứ *i*

1.2. Mật mã truyền thống.

1.2.1. Mã chuyển dịch (shift cipher).

Các hệ mật mã dùng phép chuyển dịch nói trong mục này cũng như nhiều hệ mật mã tiếp sau đều có bảng ký tự bản rõ và bảng ký tự bản mã là bảng ký

tự của ngôn ngữ viết thông thường. Vì bảng ký tự tiếng Việt có dùng nhiều dấu phụ làm cho cách xác định ký tự khó thống nhất, nên trong tài liệu này ta sẽ lấy bảng ký tự tiếng Anh để minh hoạ, bảng ký tự này gồm có 26 ký tự, được đánh số từ 0 đến 25 như trình bày ở tiết 1.2.1, ta có thể đồng nhất nó với tập Z_{26}. Như vậy, sơ đồ các hệ mật mã chuyển dịch được định nghĩa như sau:

$$S = (P, C, K, E, D),$$

trong đó $P = C = K = Z_{26}$, các ánh xạ E và D được cho bởi:

với mọi $K, x, y \in Z_{26}$:

$E(K, x) = x + K \bmod 26$,

$D(K, y) = y - K \bmod 26$.

Các hệ mật mã được xác định như vậy là đúng đắn, vì với mọi K, x, y Z_{26} ta đều có:

$$d_K(e_K(x)) = (x + K) - K \bmod 26 = x.$$

Các hệ mật mã chuyển dịch đã được sử dụng từ rất sớm, theo truyền thuyết, hệ mã đó với $K = 3$ đã được dùng bởi J. Caesar từ thời đế quốc La mã, và được gọi là hệ mã Caesar.

Thí dụ: Cho bản rõ hengapnhauvaochieuthubay, chuyển dãy ký tự đó thành dãy số tương ứng ta được:

x = 7 4 13 6 0 15 13 7 0 20 21 0 14 2 7 8 4 20 19 7 20 1 0 24.

Nếu dùng thuật toán lập mật mã với khoá $K = 13$, ta được bản mã là:

y= 20 17 0 19 13 2 0 20 13 7 8 13 1 15 20 21 17 7 6 20 7 14 13 11.

chuyển dưới dạng ký tự thông thường ta được bản mật mã là:

uratncaunhinbpuv rhguhonl.

Để giải bản mật mã đó, ta chỉ cần chuyển nó lại dưới dạng số (để được dãy y), rồi thực hiện thuật toán giải mã, tức trừ từng số hạng với 13 (theo modulo 26), được lại dãy x, chuyển thành dãy ký tự là được bản rõ ban đầu.

Các hệ mật mã chuyển dịch tuy dễ sử dụng, nhưng việc thám mã cũng khá dễ dàng, số các khoá có thể có là 26; nhận được một bản mã, người thám mã chỉ cần thử dùng lần lượt tối đa là 26 khoá đó để giải mã, ắt sẽ phát hiện ra được khoá đã dùng và cả bản rõ!

1.2.2. Mã thay thế (substitution cipher).

Sơ đồ các hệ mật mã thay thế được định nghĩa như sau:

$$S = (P, C, K, E, D),$$

trong đó, $P = C = Z_{26}$, K là tập hợp tất cả các phép hoán vị trên Z_{26}. Các ánh xạ E và D được cho bởi:

$$e_\pi(x) = \pi(x)$$

$$d_\pi(y) = \pi^{-1}(y)$$

với mọi $x \in P$, $y \in C$, $\pi \in K$ là một phép hoán vị trên Z_{26} .

Ta thường đồng nhất Z_{26} với bảng ký tự tiếng Anh, do đó phép hoán vị trên Z_{26} cũng được hiểu là một phép hoán vị trên tập hợp các ký tự tiếng Anh, thí dụ một phép hoán vị được cho bởi bảng:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
x	n	y	a	h	p	o	g	z	q	w	b	t	s	f	l	r	c

s	t	u	v	w	x	y	z
v	m	u	e	k	j	d	i

Với hệ mật mã thay thế có khoá π, bản rõ

$$x = hengapnhauvaochieuthubay$$

sẽ được chuyển thành bản mật mã

$$y = ghsoxlsgxuexfygzhumgunxd.$$

Sơ đồ hệ mật mã có số khoá có thể bằng số các phép hoán vị trên tập Z_{26}, tức là 26! khoá có thể có. Đó là một số rất lớn ($26! > 4.10^{26}$). Do đó, việc duyệt

lần lượt tất cả các khoá có thể để thám mã là không thực tế, ngay cả dùng máy tính. Tuy vậy, có những phương pháp thám mã khác hiệu quả hơn, làm cho các hệ mật mã thay thế không thể được xem là an toàn. Thuật toán giải mã với khoá cho trước sẽ biến *y* thành bản rõ *x*.

1.2.3. Mã apphin.

Sơ đồ các hệ mật mã apphin được định nghĩa như sau: Hệ mật mã apphin là một bộ 5 thành phần: (P , C , K , E , D) ,

trong đó, $P = C = Z_{26}$, $K = \{ (a ,b) \ Z_{26} \text{x} \ Z_{26} | \gcd(a, 26) = 1\}$, các ánh xạ *E* và *D* được cho bởi:

$$e_K(X) = ax + b \bmod 26,$$

$$d_K(Y) = a^{-1}(y - b) \bmod 26,$$

với mọi $x \in P$, $y \in C$, k= $(a, b) \in K$.

Điều kiện gcd(*a*, 26) = 1 để bảo đảm có phần tử nghịch đảo $a^{-1} \bmod 26$ của *a*, làm cho thuật toán giải mã d_K luôn thực hiện được. Có tất cả $\phi(26) = 12$ số $a \epsilon Z_{26}$ nguyên tố với 26, đó là các số:

1, 3, 5, 7 ,9, 11, 15, 17, 19, 21, 23, 25,

và các số nghịch đảo theo mod 26 tương ứng của chúng là

1, 9, 21, 15, 3, 19, 7, 23, 11, 5, 17, 25.

Thí dụ với bản rõ " hengapnhauvaochieuthubay", có dãy số tương ứng là:

x = 7 4 13 6 0 15 13 7 0 20 21 0 14 2 7 8 4 20 19 7 20 1 0 24.

Nếu dùng hệ mật mã apphin với khoá k=(5, 6) ta sẽ được bản mật mã

y = 15 0 19 10 6 3 19 15 6 2 7 6 24 16 15 20 0 2 23 15 2 11 6 22,

chuyển sang dòng ký tự tiếng La tinh, ta được bản mật mã dưới dạng

patkgdtpgchgyqpuacxpclgw.

Vì có 12 số thuộc Z_{26} nguyên tố với 26, nên số các khoá có thể có (do đó, số các hệ mật mã apphin) là bằng 12x26 = 312, một con số không lớn lắm nếu

ta sử dụng máy tính để thực hiện việc thám mã bằng cách duyệt lần lượt tất cả các khoá có thể. Như vậy, mã apphin cũng không còn được xem là mã an toàn!

1.2.4. Mã Vigenere.

Sơ đồ mật mã này lấy tên của Blaise de Vigenere, sống vào thế kỉ 16. Khác với các hệ mật mã đã được trình bày ở trên, Hệ mật mã Vigenere không thực hiện trên từng ký tự một, mà được thực hiện trên từng bộ m ký tự (*m* là số nguyên dương).

Sơ đồ các hệ mật mã Vigenere được định nghĩa như sau: mật mã Vigenere là một bộ 5 thành phần: (*P*, *C*, *K*, *E*, *D*),

trong đó, $P = C = K = Z_{26}^m$, các ánh xạ *E* và *D* được cho bởi:

$e_K(x_1,..., x_m) = (x_1 + k_1,...., x_m + k_m)$ mod 26. Hàm giải mã được cho bởi:

$$d_K(y_1,..., y_m) = (y_1 - k_1,..., y_m - k_m) \text{ mod } 26$$

với mọi $x = (x_1,..., x_m) \in P$, $y = (y_1,..., y_m) \in C$, $k = (k_1,...,k_m) \in K$. với $m \geq 1$

Sơ đồ mã Vigenere có thể được xem là mở rộng của sơ đồ mã chuyển dịch, nếu mã chuyển dịch thực hiện việc chuyển dịch từng ký tự một thì mã Vigenere thực hiện đồng thời từng bộ $m \geq 1$ ký tự liên tiếp. Thí dụ lấy $m = 6$ và $k = (2, 8, 15, 7, 4, 17)$. Để mã mã hóa bản rõ:

hengapnhauvaochieuthubay,

ta cũng chuyển nó thành dãy số và tách thành từng đoạn 6 số liên tiếp:

x = 7 4 13 6 0 15 | 13 7 0 20 21 0 | 14 2 7 8 4 20 | 19 7 20 1 0 24.

(nếu độ dài của x không phải là bội số của 6, ta có thể qui ước thêm vào đoạn cuối của *x* một số phần tử nào đó, chẳng hạn là các số 0, để bao giờ cũng có thể xem là *x* nhóm được thành các đoạn có 6 số liên tiếp). Cộng theo mod 26 các số trong từng đoạn đó với các số tương ứng trong khoá *k* ta sẽ được bản mật mã:

y = 9 12 2 13 4 6 | 15 15 15 1 25 17 | 16 10 22 15 8 11 | 21 15 9 8 4 15

chuyển sang dãy ký tự ta được bản mã là

jmcnegpppbzrqkwpilvpjiep.

Từ bản mã đó, dùng thuật toán giải mã tương ứng ta lại thu được bản rõ ban đầu.

Tập K có tất cả là 26^m phần tử, do đó với mỗi m có tất cả là 26^m hệ mật mã Vigenere khác nhau (với m = 6 thì số đó là 308,915,776), duyệt toàn bộ chừng ấy khoá để thám mã bằng tính thủ công thì khó, nhưng nếu dùng máy tính đủ mạnh thì cũng không đến nỗi khó lắm! Nhưng điều quan trọng là m có thể thay đổi tăng lên để việc thám mã bằng vét cạn là không khả thi.

1.2.5. Mã Hill.

Sơ đồ mật mã này được đề xuất bởi Lester S. Hill năm 1929. Cũng giống như sơ đồ mã Vigenere, các hệ mã này được thực hiện trên từng bộ m ký tự liên tiếp, điều khác là mỗi ký tự của bản mã được xác định bởi một tổ hợp (trên vành Z_{26}) của m ký tự trong bản rõ. Như vậy, khoá sẽ được cho bởi một ma trận cấp m, tức là một phần tử của $k \in Z^{m \cdot m}$. Để phép biến đổi tuyến tính xác định bởi ma trận k có phép nghịch đảo, bản thân ma trận k cũng phải có ma trận nghịch đảo k^{-1} theo mod 26; mà điều kiện cần và đủ để k có nghịch đảo là định thức của nó, ký hiệu det k, nguyên tố với 26. Vậy, sơ đồ mật mã Hill được định nghĩa là bộ 5 thành phần:

$$(P, C, K, E, D),$$

trong đó, $P = C = Z_{26}^m$, $K = \{k \in Z_{26}^m : \gcd(\det k, 26) = 1\}$,

các ánh xạ E và D được cho bởi:

$$e_k(x_1,..., x_m) = (x_1,..., x_m) \cdot k \bmod 26,$$

$$d_k(y_1,..., y_m) = (y_1,..., y_m) \cdot k^{-1} \bmod 26$$

với mọi $x = (x_1,..., x_m) \in P$, $y = (y_1,..., y_m) \in C$, $k \in K$.

Thí dụ : Chọn $m = 2$, và $k = \begin{Bmatrix} 11 & 8 \\ 3 & 7 \end{Bmatrix}$.

Với bộ hai ký tự $x = (x_1, x_2)$ ta có dãy $y = (y_1, y_2) \cdot K$ được tính bởi:

$$y_1 = 11 \cdot 1 + 3 \cdot 2$$

$$y_2 = 8 \cdot 1 + 7 \cdot 2 \bmod 26.$$

Ta lấy lại bản rõ

hengapnhauvaochieuthubay,

ta cũng chuyển nó thành dãy số và tách thành từng đoạn 2 số liên tiếp:

x = 7 4 | 13 6 | 0 15 | 13 7 | 0 20 | 21 0 | 14 2 | 7 8 | 4 20 | 19 7 | 20 1 | 0 24. Lập mật mã cho từng đoạn hai số liên tiếp, rồi nối ghép lại ta được

y = 11 6 |516 | 191 | 8 21 | 8 2 | 23 12 | 4 22 | 23 8 | 0 16 | 22 19 | 15 11 | 20 12.

Từ đó ta được bản mật mã dưới dạng dãy ký tự là:

lgfqtbivicxmewxiaqwtplum.

Chú ý rằng

$$K^{-1} = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}^{-1} (\text{mod } 26) = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix},$$

và giải mã bằng cách nhân từng đoạn hai số liên tiếp của y với K^{-1} ta sẽ được lại dãy x, và từ đó được lại bản rõ.

Với mỗi số m cho trước, số các khoá có thể có là bằng số các ma trận k có det$k$ nguyên tố với 26. Ta không có công thức để tính số đó, tuy biết rằng khi m lớn thì số đó cũng là rất lớn, và tất nhiên việc thám mã bằng cách duyệt lần lượt toàn bộ các hệ mã Hill có cùng số m là không khả thi. Mặc dù vậy, từ lâu người ta cũng đã tìm được những phương pháp thám mã khác đối với hệ mã Hill một cách khá hiệu quả.

1.2.6. Mã hoán vị (chuyển vị - Transposition).

Các hệ mã hoán vị cũng được thực hiện trên từng bộ m ký tự liên tiếp, nhưng bản mật mã chỉ là một hoán vị của các ký tự trong từng bộ m ký tự của bản rõ. Ta ký hiệu S_m là tập hợp tất cả các phép hoán vị của tập hợp $\{1, 2, \ldots, m\}$. Sơ đồ các phép mã hoán vị được cho bởi bộ 5 thành phần:

$$(P, C, K, E, D),$$

trong đó, $P = C = Z_{26}^m$, $K = S_m$, các ánh xạ E và D được cho bởi:

$$e_K(x_1,..., x_m) = (x_{\pi(1)},..., x_{\pi(m)}),$$

$$d_K(y_1,..., y_m) = (Y_{\pi^{-1}(1)}, ..., Y_{\pi^{-1}(m)}),$$

với mọi $x = (x_1,..., x_m) \in P$, $y = (y_1,..., y_m) \in C$, $k = \pi \in S_m$, π^{-1} là hoán vị nghịch đảo của π.

Thí dụ: Chọn $m = 6$ và phép hoán vị S_6 được cho bởi:

$$i = 1\ 2\ 3\ 4\ 5\ 6$$

$$\pi(i) = (\ 3\ 5\ 1\ 6\ 4\ 2\ )$$

Khi đó phép hoán vị π^{-1} sẽ là

$$j = 1\ 2\ 4\ 5\ 6$$

$$\pi^{-1}(j) = (\ 3\ 6\ 1\ 5\ 2\ 4\ ).$$

các ô của bảng theo thứ tự tự nhiên từ trái sang phải và từ trên xuống cho đến hết bản thông điệp. Bây giờ, để nhận được bản mã, ta chỉ việc nhặt các ký tự trong bảng theo thứ tự từ trên xuống và lần lượt từ cột nhỏ nhất cho đến cột lớn nhất. Kết quả là ta có bản mã bằng phương pháp chuyển vị. Bản mã này Với bản rõ hengapnhauvaochieuthubay, tức cũng với

$$x = 7\ 4\ 13\ 6\ 0\ 15\ 13\ 7\ 0\ 20\ 21\ 0\ 14\ 2\ 7\ 8\ 4\ 20\ 19\ 7\ 20\ 1\ 0\ 24.$$

ta sẽ có bản mã tương ứng là:

$$y = 13\ 0\ 7\ 15\ 6\ 4\ \ 0\ 21\ 13\ 0\ 20\ 7\ \ 7\ 4\ 14\ 20\ 8\ 2\ \ 20\ 0\ 19\ 24\ 1\ 7$$

chuyển thành dãy ký tự là: nahpgeavnauhheouicuatybh. Dùng cho từng bộ 6 ký tự liên tiếp của bản mật mã này (tức là của y) phép giải mã d_K ta sẽ thu lại được x và bản rõ ban đầu.

Chú ý rằng mã hoán vị là một trường hợp riêng của mã Hill. Thực vậy, cho phép hoán vị trên $\{1,2,...,m\}$, ta xác định ma trận $k_\pi = (k_{ij})$ với $k_{ij} = 1$ nếu $i = \pi(j)$, và = 0 nếu ngược lại, thì dễ thấy rằng mã Hill với khoá k cho cùng một phép mật mã như mã hoán vị với khoá k. Với mỗi m cho trước, số các khóa có thể có của Hệ mật mã hoán vị là $m!$

Để đơn giản trong thực hành, khi muốn mã hóa một thông điệp x độ dài n tùy ý với khóa chuyển vị $\pi = (k_1, k_2, ..., k_m)$, trong đó, $k_i \in \{ 1, 2, 3,..., m \}$, $i = 1, 2, ...$ m và $m < n$, người ta làm như sau: trước hết, ta lập một bảng (ma trận) [n/m] dòng nếu n chia hết cho m và bằng $[n/m] + 1$ nếu n không chia hết cho m và m cột được viết theo thứ tự $k_1, k_2,..., k_m$. Sau đó, ta viết bản thông điệp x vào thường được viết thành từng nhóm, mỗi nhóm 5 ký tự.

1.3. Thám mã đối với mã Vigenere.

Mã Vigenere có thể coi là mã chuyển dịch đối với từng bộ m ký tự. Khoá mã là một bộ $K= (k_1,..., k_m)$ gồm m số nguyên mod 26. Việc thám mã gồm hai bước: bước thứ nhất xác định độ dài m, bước thứ hai xác định các số $k_1,..., k_m$. Có hai phương pháp để xác định độ dài m: phép thử Kasiski và phương pháp dùng chỉ số trùng hợp.

Phép thử Kasiski (đề xuất từ 1863). Phép thử dựa vào nhận xét rằng hai đoạn trùng nhau của bản rõ sẽ được mã hoá thành hai đoạn trùng nhau của bản mã, nếu khoảng cách của chúng trong văn bản rõ (kể từ ký tự đầu của đoạn này đến ký tự đầu của đoạn kia) là bội số của m. Mặt khác, nếu trong bản mã, có hai đoạn trùng nhau và có độ dài khá lớn (3 chẳng hạn) thì rất có khả năng chúng là mã của hai đoạn trùng nhau trong bản rõ. Vì vậy, ta thử tìm một đoạn mã (có ba ký tự trở lên) xuất hiện nhiều lần trong bản mã, tính khoảng cách của các lần xuất hiện đó, chẳng hạn được $d_1, d_2, ..., d_t$; khi đó ta có thể phán đoán $m = d = \gcd(d_1, d_2,..., d_t)$- ước số chung lớn nhất của $d_1, d_2..., d_t$; hoặc m là ước số của d.

Phương pháp dùng chỉ số trùng hợp: (định nghĩa chỉ số trùng hợp do W.Friedman đưa ra năm 1920).

Định nghĩa 1.3.1. Cho hai dãy $x = x_1, x_2... x_n$ và $y = y_1, y_2,..., y_n$ gồm n ký tự. Ta hãy viết dãy y dưới dãy x theo cách: y_1 dưới x_1; y_2 dưới x_2, v.v, cho đến y_n

dưới x_n. Nếu có tồn tại một cặp $x_i = y_i$ thì ta nói rằng chúng ta đã có một trùng khớp (Coincidence) giữa ký tự x_i với ký tự y_i . Ta có định lý sau đây:

Định lý. [4]

Ký hiệu $f_0, f_1,..., f_{25}$ lần lượt là tần số xuất hiện của $a, b, ... ,z$ trong Z_{26}, ta có:

$$I_C(x) = \frac{\sum_{i=0}^{25}\binom{f_i}{2}}{\binom{n}{2}} = \frac{\sum_{i=0}^{25} f_i(f_i+1)}{n(n+1)}$$

Giả sử x là một dãy ký tự (tiếng Anh). Người ta [1] đã tính được rằng:

$$I_C(x) \approx \sum_{i=0}^{25} p_i^2 = 0{,}065,$$

trong đó, p_i là xác suất của ký tự ứng với số hiệu i cho bởi bảng phân bố xác suất các ký tự. Nếu x là một dãy ký tự hoàn toàn ngẫu nhiên thì ta có [4]:

$$I_C \approx 26.\ (1/26)^2 = 1/26 = 0{,}038.$$

Dựa vào các điều nói trên, ta có phương pháp đoán độ dài m của mã Vigenere như sau: Cho bản mã $y = y_1y_2..., y_n$. Ta viết lại y theo bảng có $m(m \geq 1)$ cột như sau: $y = y_1y_2..... y_{tm}$

$y_{m+1}y_{m+2}..... y_{2m}$

.....

$y_{mk+1}y_{mk+2}..... y_{(mk+1)m}$

nghĩa là viết lần lượt theo các cột m ký tự cho đến hết. Ta ký hiệu $y_1, y_2,..., y_m$ là các xâu ký tự theo m hàng trong bảng đó. Chú ý rằng các ký tự ở mỗi hàng y_i đều thu được từ các ký tự ở văn bản gốc bằng cùng một phép dịch chuyển nếu m đúng là độ dài của khoá, do đó nếu m là độ dài của khoá thì ta có thể hi vọng rằng với mọi i, $1 \leq i \leq m$:

$$I_C(y_i) \approx 0{,}065.$$

Để đoán độ dài *m*, ta lần lượt chia *y* theo cách trên thành *m* = 1, 2, 3... hàng, và tính các $I_C(yi)$ ($1 \le i \le m$), cho đến khi nào được một số *m* mà với mọi *i*, $1 \le i \le m$, đều có $I_C(y_i) \approx 0{,}065$thì ta có thể chắc m là độ dài của khoá.

Thí dụ: Cho bản mã

chreevoahmaeratbiaxxwtnxbeeophbsbqmqeqerbwrvxuoakxaosxxweahbwgjm
mqmnkgrfvgxwtrzxwiaklxfpskautemndemgtsxmxbtuiadngmgpsrelxnjelxvrvp
rtulhdnqwtwdtygbphxtfaljhasvbfxngllchrzbwelekmsjiknbhwrignmgjsglxfeyph
agnbieqjtmrvlcrremndglxrrimgnsnrwchrqhaeyevtaqebbipeewevkakoewadrem
xmtbhhchrtkdnvrzchrclqohpwqaiiwxnrmgwoiifkee.

Dùng phép thử Kasiski, ta nhận thấy rằng chữ r xuất hiện 5 lần, khoảng cách của các lần xuất hiện liên tiếp là 165, 70, 50, 10. Ước số chung của các số đó là 5. Vậy ta có thể phán đoán độ dài khoá mã là 5.

Dùng phương pháp chỉ số trùng hợp, với *m* = 1 ta có một chỉ số trùng hợp là 0,045; với *m* = 2 có hai chỉ số là 0,046 và 0,041; với *m* = 3 có ba chỉ số là 0,043; 0,050 và 0,047; với *m* = 4 có bốn chỉ số là 0,042; 0,039; 0,046 và 0,043; với *m*= 5, ta thu được năm chỉ số là 0,063; 0,068; 0,069; 0,061 và 0,072, đều khá gần với 0,065. Vậy có thể phán đoán độ dài khoá là 5. Cả hai phương pháp cho kết quả như nhau.

Bây giờ đến bước thứ hai là xác định các giá trị $k_1, k_2,...k_m$. Ta cần một khái niệm mới là chỉ số trùng hợp tương hỗ, được định nghĩa như sau:

Định nghĩa 1.3.2. Giả sử $x = x_1x_2... x_n$ và $y = y_1y_2... y_n$ là hai dãy ký tự cùng có độ dài *n*. Chỉ số trùng hợp tương hỗ của *x* và *y*, ký hiệu $MI_C(x, y)$, được định nghĩa là xác suất để cho hai ký tự x_i và y_i tương ứng của hai dãy trùng nhau (đồng tự).

Ký hiệu là tần suất xuất hiện của *a, b,...,z* trong *x* và *y* tương ứng trùng nhau là:

$$MI_C(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=0}^{25} f_i . f_i^{'}}{n.n'} .$$

Bây giờ với m đã xác định, ta viết bản mã y lần lượt theo từng cột để được m hàng $y_1,... y_m$ như ở phần trên. Ta tìm khoá mã $k = (k_1,k_2,...k_m)$. Giả sử x là bản rõ và$x_1,..., x_m$ là các phần bản rõ tương ứng với $y_1,...,y_m$. Ta có thể xem phân bố xác suất của các ký tự trên x, và cũng trên các $x_1,..., x_m$ là xấp xỉ với phân bố xác suất của các ký tự trên văn bản tiếng Anh nói chung. Do đó, xác suất của việc một ký tự ngẫu nhiên của y_i bằng a là, bằng b là, v.v... Và ta có thể đánh giá

$$MI_C(y_i, y_j) \approx \sum_{h=0}^{25} p_{h-k_i} \cdot p_{h-k_j} = \sum_{h=0}^{25} p_h \cdot p_{h+k_i-k_j}.$$

Đại lượng đó chỉ phụ thuộc vào k_i - k_j, ta gọi là dịch chuyển tương đối của y_i và y_j. Ta chú ý rằng biểu thức:

$$\sum_{h=0}^{25} p_h \cdot p_{h+l}$$

có giá trị lớn nhất khi $l = 0$ là 0,065, và có giá trị biến thiên giữa 0,031 và 0,045 với mọi $l > 0$.

Nhận xét rằng y_j phải dịch chuyển $l = k_i - k_j$ bước (hay dịch chuyển l ký tự trong bảng chữ cái) để được y_i, nên nếu ký hiệu y_j^g là dịch chuyển g bước của y_j, thì ta có hi vọng khi tính lần lượt các đại lượng $MI_C(y_i, y_j^g)$ với $0 \le g \le 25$, ta sẽ đạt được một giá trị xấp xỉ 0,065 với $g = l$, và các giá trị khác đều ở khoảng giữa 0,031 và 0,045. Điều đó cho ta một phương pháp để ước lượng các dịch chuyển k_i - k_j , tức là được một số phương trình dạng k_i - k_j = l, từ đó giúp ta tính ra các giá trị $k_1, k_2,..., k_m$.

Trong thí dụ của bản mã đang xét, ta tính được các giá trị $MI_C(y_i, y_j^g)$ với $1 \le i \le j \le 5$, $0 \le g \le 25$, như trong bảng ở trang sau đây (trong bảng đó, ở bên phải mỗi cặp (i, j) là một ngăn gồm có 26 giá trị của $MI_C(y_i, y_j^g)$ ứng với các giá trị của $g = 0,1,2,..., 25$).

Nhìn bảng đó, ta thấy các giá trị $MI_C(y_i, y_j^g)$ xấp xỉ 0.065 (như đã được in đậm và gạch dưới ở trong bảng) ứng với các bộ giá trị (i, j, g) lần lượt bằng (1,2,9), (1,5,16), (2,3,13), (2,5,7), (3,5,20) và (4,5,11).

i	j	Giá trị $MI_C(y_i,y_j^g)$
1	2	.028 .027 .028 .034 .039 .037 .026 .025 .052 .068 .044 .026 .037 .043 .037 .043 .037 .028 .041 .041 .034 .037 .051 .045 .042 .036
1	3	.039 .033 .040 .034 .028 .053 .048 .033 .029 .056 .050 .045 .039 .040 .036 .037 .032 .027 . 037 .036 .031 .037 .055 .029 .024 .037
1	4	.034 .043 .025 .027 .038 .049 .040 .032 .029 .034 .039 .044 .044 .034 .039 .045 .044 .037 .055 .047 .032 .027 .039 .037 .039 .035
1	5	.043 .033 .028 .046 .043 .044 .039 .031 .026 .030 .036 .040 .041 .024 .019 .048 .070 .044 .028 .038 .044 .043 .047 .033 .026 .046
2	3	.046 .048 .041 .032 .036 .035 .036 030 .024 .039 .034 .029 .040 .067 .041 .033 .037 .045 .033 .033 .027 .033 .045 .052 .042 .030
2	4	.046 .034 .043 .044 .034 .031 .040 .045 040 .048 .044 .033 .024 .028 .042 .039 .026 .034 .050 .035 ,032 .040 .056 .043 .028 .028
2	5	.033 .033 .036 .046 .026 .018 .043 .080 .050 .029 .031 .045 .039 .037 .027 .026 .031 .039 .040 .037 .041 .046 .045 .043 .035 .030
3	4	.038 .036 .040 .033 .036 .060 .035 .041 .029 .058 .035 .035 .034 .053 .030 .032 .035 .036 .036 .028 .046 .032 .051 .032 .034 .030
3	5	.035 .034 .034 .036 .030 .043 .043 .050 .025 .041 .051 .050 .035 .032 .033 .033 .052 .031 .027 .030 .072 .035 .034 .032 .043 .027
4	5	.052 .038 .033 .038 .041 .043 .037 .048 .028 .028 .036 .061 .033 .033 .032 .052 .034 .027 .039 .043 .033 .027 .030 .039 .048 .035

Từ đó ta có các phương trình (theo mod 26):

$$k_1 - k_2 = 9 \qquad k_2 - k_5 = 7$$

$$k_1 - k_5 = 16 \qquad k_3 - k_5 = 20$$

$$k_2 - k_3 = 13 \qquad k_4 - k_5 = 11.$$

Hệ phương trình đó chỉ có 4 phương trình độc lập tuyến tính, mà có 5 ẩn số, nên lời giải phụ thuộc một tham số, ta chọn là k_1, và được

$$(k_1, k_2, k_3, k_4, k_5) = (k_1, k_1 + 17, k_1 + 4, k_1 + 21, k_1 + 10) \bmod 26.$$

Thử với các giá trị có thể của k_1 ($0 \le k_1 \le 26$), cuối cùng ta có thể tìm được bản rõ như sau đây với khoá là JANET ($k_1 = 9$):

the almond tree was in tentative blossom the days were longer often ending with magnificent evenings of corrugated pink skies the hunting season was over with hounds and guns put away for six months the vineyards were busy again as the well organized farmers treated their vines and the more lackadaisical neighbors hurried to do the pruning they should have done in november.

1.4. Mật mã khóa công khai.

1.4.1. Hệ mật mã công khai RSA.

Sơ đồ chung của hệ mật mã khoá công khai được cho bởi

$$S = (P, C, K, E, D) \qquad (1)$$

trong , đó P là tập ký tự bản rõ, C là tập ký tự bản mã, K là tập các khoá k, mỗi khoá k gồm có hai phần $k = (k', k'')$, k' là khoá công khai dành cho việc lập mã, còn k'' là khoá bí mật dành cho việc giải mã. Với mỗi ký tự bản rõ $x \in P$, thuật toán lập mã E cho ta ký tự mã tương ứng $y = E(k', x) \in C$, và với ký tự mã y. Thuật toán giải mã D sẽ cho ta lại ký tự bản rõ x: $D(k'', y) = D(k'', E(k', x)) = x$.

Để xây dựng một hệ mật mã khoá công khai RSA, ta chọn trước một số nguyên $n = p \cdot q$ là tích của hai số nguyên tố lớn và khác nhau; chọn một số e sao cho $\gcd(e, (n)) = 1$, và tính số d sao cho

$$e \,.\, d \equiv 1 (\mathrm{mod}\, \phi(n)).$$

Mỗi cặp $k = (k', k'')$, với $k' = (n, e)$ và $k'' = d$ sẽ là một cặp khoá của một hệ mật mã RSA cụ thể cho một người tham gia.

Như vậy, sơ đồ chung của hệ mật mã RSA được định nghĩa bởi danh sách (1), trong đó:

$P = C = Z_n$, n là một số nguyên Blum, tức là tích của hai số nguyên tố;

$K = \{k = (k', k''): k' = (n, e)$ và $k'' = d, \gcd(e, \phi(n)) = 1, e \cdot d \equiv 1 (\mathrm{mod}\, \phi(n))\}$;

E và D được xác định bởi:

$E(k', x) = x^e \bmod n$, với mọi $x \in P$,

$D(k'', y) = y^d \bmod n$, với mọi $y \in C$.

Để chứng tỏ định nghĩa trên là hợp thức, ta phải chứng minh rằng với mọi cặp khoá $k = (k', k'')$, và mọi $x \in P$, ta đều có

$$D(k'', E(k', x)) = x .$$

Thực vậy, do $e \cdot d \equiv 1 \bmod (n)$, ta có thể viết $e \cdot d = t \cdot \phi(n) + 1$. Nếu x nguyên tố với n, thì dùng định lý Euler ta có

$$D(k'', E(k', x)) = x^{ed} \equiv x^{t\phi(n)+1} \equiv x^{t\phi(n)}.x \,(\mathrm{mod}\, n) = x.$$

Nếu x không nguyên tố với n, thì do $n = p \cdot q$, hoặc x chia hết cho p và nguyên tố với q, hoặc x chia hết cho q và nguyên tố với p, và $\phi(n) = (p - 1) \cdot (q - 1)$,trong cả hai trường hợp ta đều có

$$x^{t\phi(n)+1} \equiv x \,(\mathrm{mod}\, p),$$
$$x^{t\phi(n)+1} \equiv x \,(\mathrm{mod}\, q);$$

từ đó ta suy ra $x^{t\phi(n)+1} \equiv x \,(\mathrm{mod}\, n)$, tức là $D(k'', E(k', x)) = x$.

Thí dụ: Giả sử chọn $n = p \cdot q = 2357 \cdot 2551 = 6012707$, ta sẽ có$(n) = (p - 1) \cdot (q - 1) = 2356 \cdot 2550 = 6007800$. Chọn $e = 3674911$, và tính được $d = 422191$ sao cho $e \cdot d \equiv 1 (\mathrm{mod}\, (n))$. Một người dùng A có thể chọn khoá công khai là $k' = (n = 6012707, e = 3674911)$ và giữ khoá bí mật $k'' = d = 422191$. Một đối tác B muốn gửi cho A một thông báo $x = 5234673$, anh ta sẽ dùng khoá công khai của A để tạo bản mật mã $y = x^e = 52346733674911 \bmod 6012707 = 3650502$. A nhận được y, giải mã sẽ được bản rõ $x = 3650502422191 \bmod 6012707 = 5234673$.

1.4.2. Hệ mật mã khoá công khai Rabin.

Sơ đồ hệ mật mã khoá công khai Rabin được cho bởi

$$S = (P, C, K, E, D),$$

trong đó: $P = C = Z_n$, n là một số nguyên Blum, $n = p \cdot q$, với p và q là hai số nguyên tố có tính chất $p \equiv 3(\text{mod } 4)$, $q \equiv 3(\text{mod } 4)$, $K = \{K = (K', K'') : K' = (n, B), K'' = (p, q), 0 \le B \le n - 1\}$, các thuật toán E và D được xác định bởi $E(K', x) = x(x + B) \bmod n$, $D(K'', y) = \sqrt{\frac{B^2}{4} + y} - \frac{B}{2} \bmod n.$

Trong một mạng truyền tin bảo mật với sơ đồ mật mã Rabin, mỗi người tham gia chọn cho mình các yếu tố n, B, p, q để lập nên khoá công khai và khoá bí mật của mình

Ta chú ý rằng với mỗi bộ khoá K, các thuật toán $e_{K'} = E(K', .)$ và $d_{K''} = D(K'', .)$ không lập thành một cặp song ánh, cụ thể là không phải là một đơn ánh, vì nếu w là một căn bậc hai của 1 theo mod n thì $(w(x + \frac{B}{2}) - \frac{B}{2}) = e_{K'}(x)$, mà ta có đến 4 căn bậc hai của 1 theo mod n ,tức là ta có 4 giá trị khác nhau của đối số x cho cùng một giá trị $e_{K'}(x)$.

Bây giờ nói đến thuật toán giải mã $d_{K''} = D(K'', .)$. Đặt $C = B^2/4 + y$, ta có $d_{K''}(y) = \sqrt{C} - B/2 \bmod n$, do đó để có $d_{K''}(y)$, ta cần tính $\sqrt{C} \bmod n$, tức cần giải phương trình $z^2 \equiv C \bmod n$. Phương trình đó tương đương với hệ thống gồm hai phương trình sau đây:

$$\begin{cases} z^2 \equiv C \bmod p, \\ z^2 \equiv C \bmod q. \end{cases} \qquad (2)$$

Vì p và q là các số nguyên tố nên ta có $C^{\frac{p-1}{2}} \equiv 1 \bmod p$, $C^{\frac{q-1}{2}} \equiv 1 \bmod q$. Theo giả thiết, $p \equiv 3(\text{mod } 4)$ và $q \equiv 3(\text{mod } 4)$, nên $\frac{p+1}{4}$ và $\frac{q+1}{4}$ là các số nguyên; và ta có

$$(\pm C^{\frac{p+1}{4}})^2 \equiv C(\text{mod } p), (\pm C^{\frac{q+1}{4}})^2 \equiv C(\text{mod } q).$$

Do đó, phương trình $z^2 \equiv C \bmod n$, hay hệ phương trình (2), có 4 nghiệm theo mod n, tương ứng với 4 hệ phương trình sau đây:

$$\begin{cases} z \equiv C^{(p+1)/4} (\bmod p) \\ z \equiv C^{(q+1)/4} (\bmod q) \end{cases} \quad \begin{cases} z \equiv C^{(p+1)/4} (\bmod p) \\ z \equiv -C^{(q+1)/4} (\bmod q) \end{cases}$$

$$\begin{cases} z \equiv -C^{(p+1)/4} (\bmod p) \\ z \equiv C^{(q+1)/4} (\bmod q) \end{cases} \quad \begin{cases} z \equiv -C^{(p+1)/4} (\bmod p) \\ z \equiv -C^{(q+1)/4} (\bmod q) \end{cases}$$

Cả 4 nghiệm của 4 hệ phương trình đó theo mod n đều được viết chung dưới một ký hiệu là $\sqrt{C} \bmod n$, và vì vậy thuật toán giải mã $d_{K''}(y)$ thực tế sẽ cho ta 4 giá trị khác nhau theo mod n mà bản rõ là một trong 4 giá trị đó. Việc chọn giá trị nào trong 4 giá trị tìm được làm bản rõ là tuỳ thuộc vào những đặc trưng khác của bản rõ mà người giải mã nhận biết (thí dụ bản rõ dưới dạng số phải có biểu diễn nhị phân là mã của một văn bản tiếng Anh thông thường).

Thí dụ: Giả sử $n = 77 = 7 \cdot 11$, $B = 9$ (ở đây $p = 7$, $q = 11$). Ta có

$$e_{K'}(x) = x^2 + 9x \bmod 77,$$

$$d_{K''}(y) = \sqrt{1+y} - 43 \bmod 77,$$

vì $2^{-1} = 39 \bmod 77$, $9 \cdot 2^{-1} = 9 \cdot 39 = 43 \bmod 77$, $B^2 = 4 \bmod 77$, $B^2/4 = 1 \bmod 77$. Với $x = 44$ ta $e_{K'}(x) = 44^2 + 9 \cdot 44 = 2332 = 22 \bmod 77$, bản mã tương ứng với x là $y = 22$. Bây giờ giải mã với bản mã $y = 22$, bằng thủ tục nói trên ta có thể tìm được 4 giá trị của $\sqrt{1+y} = \sqrt{1+22} = \sqrt{23}$ theo mod 77 là 10, 67, 32, 45, từ đó 4 giá trị có thể có của $d_{K''}(y)$ là $d_{K''}(y) = 44, 24, 66, 2$.

Bản rõ nằm trong 4 giá trị đó, trong trường hợp này là 44.

1.4.3. Hệ mật mã khoá công khai ElGamal.

Hệ mật mã ElGamal được T. ElGamal đề xuất năm 1985, dựa vào độ phức tạp của bài toán tính lô ga rit rời rạc, và sau đó đã nhanh chóng được sử

dụng rộng rãi không những trong vấn đề bảo mật truyền tin mà còn trong các vấn đề xác nhận và chữ ký điện tử.

Sơ đồ hệ mật mã khoá công khai ElGamal được cho bởi

$$S = (P, C, K, E, D),$$

trong đó: $P = Z_p^*$, $C = Z_p^* \times Z_p^*$, với p là một số nguyên tố;

$$K = \{K = (K', K'') : K' = (p, \alpha, \beta), K'' = a, \beta \equiv \alpha^a \bmod p\},$$

ở đây α là một phần tử nguyên thuỷ theo mod p, tức của Z_p^*. Các thuật toán lập mã $e_{K'} = E(K', .)$ và giải mã $d_{K''} = D(K'', .)$ được xác định như sau: Với mỗi $x \in P = Z_p^*$,để lập mật mã cho x trước hết ta chọn thêm một số ngẫu nhiên $k \in Z_{p-1}$ rồi tính:

$$e_{K'}(x, k) = (y_1, y_2), \quad \text{với} \begin{cases} y_1 = \alpha^k \bmod p, \\ y_2 = x.\beta^k \bmod p. \end{cases}$$

Với mọi số ngẫu nhiên k bất kỳ, ta đều xem $e_{K'}(x, k)$ là mật mã của x. Và thuật toán giải mã được xác định bởi

$$d_{K''}(y_1, y_2) = y_2.(y_1^a)^{-1} \bmod p.$$

Các phép lập mật mã và giải mã được xác định như vậy là hợp thức, vì ta có với mọi $x \in P = Z_p^*$ và mọi $k \in Z_{p-1}$:

$$d_{K''}(e_{K'}(x, k)) = x.\beta^k.(\alpha^{k.a})^{-1} \bmod p = x.\beta^k.\beta^{-k} \bmod p = x.$$

Ta chú ý rằng trong một mạng truyền thông bảo mật với việc dùng sơ đồ mật mã ElGamal, mỗi người tham gia tự chọn cho mình các tham số p, α, a, rồi tính β, sau đó lập và công bố khoá công khai $K' = (p, \alpha, \beta)$, nhưng phải giữ tuyệt mật khoá bí mật $K'' = a$. Bài toán biết khoá công khai tìm ra khoá bí mật chính là bài toán tính lô ga rít rời rạc, một bài toán khó cho đến nay chưa có một thuật toán nào làm việc trong thời gian đa thức giải được nó.

Thí dụ: Chọn p = 2579, α = 2, a = 765, ta tính $\beta = 2^{765} = 949 \bmod 2579$. Ta có khoá công khai (2579, 2, 949) và khoá bí mật 765. Giả sử để lập mật mã cho x = 1299, ta chọn ngẫu nhiên k = 853, sẽ có $e_{K'}(1299, 853) = (2^{853}, 1299.\ 949^{853}) \bmod 2579 = (453, 2396)$.

Và giải mã ta được lại

$$d_{K''}(453, 2396) = 2396\ .\ (453^{765})^{-1} \bmod 2579 = 1299.$$

Trong chương này chúng ta đã tìm hiểu về cơ sở lý thuyết toán học của các hệ mật mã, cơ sở về mật mã truyền thống và mật mã công khai. Đối với mật mã truyền thống ta thấy có những ưu nhược điểm như sau:

*** Ưu điểm:**

- Mật mã khóa bí mật (mật mã cổ điển) nói chung đơn giản, tức là các yêu cầu về phần cứng không phức tạp, thời gian tính toán nhanh.
- Mật mã khóa bí mật có tính hiệu quả, thông thường tốc độ mã $R_{mã} = 1$ (số bit đầu ra mã hóa bằng với số bit đầu vào).

Từ các ưu điểm này cho thấy mật mã cổ điển dễ sử dụng cho các dịch vụ nhạy cảm với độ trễ và các dịch vụ di động.

*** Nhược điểm:**

- Với mật mã khóa bí mật phải dùng kênh an toàn để truyền khóa, điều này dẫn đến chi phi sẽ cao hơn và việc thiết lập kênh an toàn khó khăn hơn.
- Việc tạo khóa và giữ bí mật khóa phức tạp. Khi làm việc trên mạng nếu dùng mật mã khóa bí mật sẽ phải tạo và lư trữ số lượng khóa nhiều.
- Nếu sử dụng mật mã khóa bí mật sẽ khó xây dựng các dịch vụ an toàn khác như các dịch vụ đảm bảo tính toàn vẹn của dữ liệu, dịch vụ xác thực và chữ ký số. Các dịch vụ này sẽ được thực hiện bởi mật mã khóa công khai.

CHƯƠNG 2: MỘT SỐ PHƯƠNG PHÁP TẤN CÔNG HỆ MÃ TRUYỀN THỐNG

2.1. Các bước cơ bản để tiến hành thám mã.

Bước 1: Phân loại bản mã.

Sau khi nhận được một số bức điện mã, các nhà phân tích mật mã cần phân loại xem những bức điện mã có cùng một loại mã pháp hay không, có cùng một loại khoá mã hay không, mặc dù chúng ta chưa biết được mã pháp phương pháp mã hoá) của các bức điện đó, nhưng chúng vẫn cần được phân loại (phân lớp). Đây là một bước quan trọng quyết định sự thành công hay thất bại của mã thám nên mất rất nhiều thời gian. Nếu việc phân loại chính xác thì sẽ thuận lợi cho các bước tiến hành tiếp theo; Ngược lại, nếu phân loại thiếu chính xác thì sẽ gây khó khăn cho các bước sau đó.

Người ta có nhiều phương pháp thực thi giai đoạn này, một trong số đó là áp dụng kỹ thuật phân lớp các đối tượng. Ý tưởng của bài toán phân lớp như sau:

Giả sử ta nhận được m bản mã $M_1, M_2, ..., M_m$ với $m > 2$. Mỗi bản mã ta gọi là một đối tượng. Tập hợp m bản mã (các đối tượng) ta ký hiệu là G.
Vậy $G = \{M_1, M_2, ..., M_m\}$, ứng với mỗi đối tượng ta cần tìm ra các đặc trưng tham số. Giả sử đối tượng M_i, có P_i đặc trưng, ở đây ta giả thiết $p_1 = p_2 = ... = p_m = p$. Vấn đề đặt ra là hãy phân tập hợp G thành k lớp không giao nhau mà ta ký hiệu là $G_1, G_2, ..., G_k$ sao cho:

i. $G_i \neq 0 \;\; i = \overline{1,k}$

ii. $G_i \cap G_j \;\; i \neq j$

iii. $G_1 \cup G_2 \cup ... \cup G_k = \bigcup_{i=m}^{k} G_i = G$

và sao cho sai sót trong phân lớp là bé nhất có thể được. Để thực hiện việc phân lớp các đối tượng ta cần đưa ra một độ đo "khoảng cách" giữa các đối tượng. Các đổi tượng càng "gần gũi" nhau sẽ được gán cho cùng một lớp.

Bước 2: Xác định mã pháp.

Sau khi hoàn thành việc phân lớp (phân loại mã pháp) ở bước 1, chúng ta tiến hành xác định phương pháp mã dịch ứng với từng lớp cụ thể (cần chú ý rằng, thường thì chúng ta tiến hành xác định mã pháp đối với các bản mã có nhiều đặc điểm nhất theo quan điểm của các nhà thám mã). Đây là một khâu rất quan trọng của công tác thám mã truyền thống. Tuy nhiên đối với một số hệ mật đối xứng hiện đại như mã DES, 3DES, AES, IDEA, PGP... thì bước này coi như được bỏ qua bởi ngay từ đầu bản mã, người ta đã chỉ ra rằng bản mã đó thuộc loại bản mã pháp nào. Ở đây chúng ta chỉ trình bày cách thức xác định mã pháp đối với các luật mã truyền thống (bước này được bỏ qua đối với những hệ mật mà thuật toán mã hoá - phương pháp mã - được công khai hoàn toàn). Bước này bao gồm các công việc sau đây:

a, Tính tần số.

Mục đích của việc tính tần số là để phát hiện tính quy luật không ngẫu nhiên tồn tại trong bản mã. Có rất nhiều loại tần số khác nhau cần tính, mà mỗi mã pháp có thể tồn tại tính không ngẫu nhiên (có quy luật) đối với các loại mã pháp khác nhau. Tùy thuộc vào kinh nghiệm của từng nhà phân tích, người ta sẽ tiến hành tính tần số loại phù hợp nhất, thông qua đó có thể bộc lộ rõ nhất tính quy luật (không ngẫu nhiên) trong bản mã. Việc tính tần số thường bao gồm:

- Tần số đơn:

Tần số đơn là tần số từng kí tự một trong bản mã. Sau khi có được kết quả tính tần số đơn, ta tiến hành sắp xếp lại thứ tự các ký tự theo tần số từ cao đến thấp. Cũng có thể lập bảng tần suất bằng cách chia tần số từng ký tự cho độ dài bản mã cần tính để xem tần số tương đối của chúng.

- Tần số bộ đôi móc xích (concatenate frequency of pairs):

Tần số bộ đôi móc xích là tần số bộ đôi nhưng các cặp kề đè lên nhau một ký tự. Mục đích của việc tính tần số bộ đôi móc xích là để xem quan hệ phụ thuộc giữa ký tự sau với ký tự kề ngay trước đó như thế nào (ta thường gọi là quan hệ xích Makov cấp 1). Từ đó có thể ước lượng được xác suất xuất hiện một ký tự nào đó khi biết trước ký tự đứng ngay trước nó.

- Tần số bộ đôi thường:

Tần số bộ đôi thường là tần số bộ đôi rời nhau, thí dụ: cho đoạn văn: Vi ee tj na m thì tần số bộ đôi thường gồm:

Vi: xuất hiện 1 lần.

ee: xuất hiện 1 lần,

tj: xuất hiện 1 lần.

na: xuất hiện 1 lần.

Ký tự cuối cùng được bỏ qua (chỉ gồm có 4 bộ đôi). Trong khi đó, tần số bộ đôi móc xích sẽ được thể hiện là: *Vi, ie, ee, et, tj, jn, na, am* gồm 8 bộ đôi

Lưu ý: Số tất cả các bộ đôi móc xích trong văn bản độ dài *n* là *n* - 1. Còn số tất cả các "bộ đôi thường" là $\left[\frac{n}{2}\right]$:

Trong đó ký hiệu [x] là số nguyên lớn nhất nhưng bé hơn hoặc bằng X.

- Tần số bộ 3, 4, 5...

Tùy theo từng trường hợp cụ thể đôi khi chúng ta phải tính tần sô bộ 3, bộ 4, bộ 5...

b, Tính mã trùng lặp (trùng mã).

Tính trùng mã tức là tính tần số trùng lặp của các dãy ký tự liền nhau trong bản mã. Thường là tính trùng lặp 3 ký tự (bộ 3), bốn ký tự (bộ 4), năm ký tự (bộ 5)... có thể xuất hiện trong bản mã và vị trí của chúng trong bản mã đó.

1. Khi tính trùng mã (các bộ) ta phải quan tâm các tham số sau đây:
2. Tần số trùng mã (trùng lặp)
3. Độ dài trùng lặp

4. Vị trí các trùng lặp
5. Khoảng cách giữa các trùng lặp
6. Trùng mã trong một bản mã và trong các bản mã khác nhau.

Những tham số trên đây rất có ích trong việc xác định mã pháp

c, Tần số định kỳ.

Ngoài việc tính tần số đơn, bộ đôi móc xích, bộ đôi thường... và trùng mã (sự trùng lặp) trong bản mã hoặc các bản mã, trong nhiều trường hợp ta phải tính tần số định kỳ. Giả sử ta có bản mã M độ dài n nào đó. Thường n khá lớn và càng lớn càng tốt. Ta lập bảng K cột ($K \geq 2$ và thường thì $K \geq 3$) và n/K hàng. Sau đó, ta viết bản mã lần lượt trái qua phải và viết từ trên xuống dưới cho đến hết thì dừng. Bây giờ ta tiến hành tính tần số đơn theo cột từ cột 1 đên cột K. Như vậy ta thường phải tính toán tần số các "định kỳ" khác nhau lần lượt k = 3, 4, 5, ... , 10... Tần số như vậy được gọi là tần số định kỳ cấp K.

d, Tần số bộ đôi dọc và bộ đôi dọc đồng tự.

Nếu ta viết 2 bản mã lần lượt bản mã này dưới bản mã kia. Thí dụ 2 bản mã$M_1 = m_{11}m_{12}... m_{1n1}$ và$M_2 = m_{21}m_{22}... m_{2n2}$.

Ta có: $M_1 = m_{11}m_{12}m_{13}... m_{1n1}$

$M_2 = m_{21}m_{22}m_{23}... m_{2n1...} m_{2n2}$.

Ta cắt phần thừa là $m_{2n1+1,...}m_{2n2}$ (giả sử $n_1 = n_2$), và ta ký hiệu độ dài hai bản mã trùng nhau là n. Ta tiến hành tính tần số từng cặp (m_{1k} ... m_{2k}), với k = 1, 2, ... , n. Ta sẽ có tần số bộ đôi và bảng này được gọi là bảng tần số bộ đôi dọc. Các phần tử trên đường chéo chính là tần số của các bộ đôi dọc đồng tự.

e, Phân tích kết quả tính các tần số và trùng mã

Bước này dựa vào các kết quả tính các loại tần số, trùng mã để kết luận bản mã (các bản mã), đó thuộc loại mã pháp nào. Để đánh giá độ chênh lệch tần số hoặc tính độc lập của các ký tự trong bản mã, người ta thường dùng các tiêu chuẩn thống kê toán, chẳng hạn tiêu chuẩn 3σ, tiêu chuẩn x^2 hoặc tiêu chuẩn

MLR (The most likelihood Ratio). Nói chung việc xác định mã pháp là công việc rất phức tạp, nó phụ thuộc một phân vào trình độ và kinh nghiệm của các mã thám viên. Có nhiều trường hợp thoáng nhìn bản mã người ta đã dự đoán được phương pháp mã nhưng cũng có rất nhiều trường hợp phải nghiên cứu rất công phu mà độ rủi ro không phải là không có.

Bước 3. Thám mã.

Giả sử, đã xác định được mã pháp tại bước thứ 2, nay chuyển sang nghiên cứu, phân tích bản mã (thám mã). Bước này cũng có hai công đoạn:

a, Thám mã trực tiếp.

Nếu mã pháp thuộc loại truyền thông đã biết như các mã pháp thủ công hoặc được mã bằng một máy mã cụ thể nào đó mà ta đã có thuật toán thám mã thì có thể tiến hành thám mã trực tiếp (thực hiện thủ công và sau đó có thể tự động hoá bằng lập trình trên máy tính).

b, Xây dựng phương pháp mã hóa.

Nếu mã pháp thuộc loại mới, công việc yêu cầu phức tạp hơn. Đó là phải xây dựng phương pháp thám mã. Nhìn chung có hai phương pháp thám mã:

- Phương pháp phân tích.
- Phương pháp dự đoán "Từ phỏng chừng".

Phương pháp phân tích được sử dụng trong trường hợp nhà mã thám đã biết được cấu trúc khoá mã đã được sử dụng làm 'mầm khoá' (key seed) để mã hoá bản mã này. Khi đó có nhiều kiểu để xác định khoá có thể, thí dụ: phương pháp 'thử - sai', phương pháp 'lượng sai', phương pháp 'những phần tử tách biệt', phương pháp 'tuyến tính'... Tóm lại tùy theo thuật toán mã hoá của bản mã như thế nào mà chọn phương pháp phân tích nào cho hợp lý.

Phương pháp 'từ phỏng chừng'

Phương pháp này chủ yếu là dựa vào thông tin tiên nghiệm về khoá và thông tin về bản mã (quy luật ngôn ngữ) để dự đoán khoá được sử dụng.

Nội dung của phương pháp này là dự đoán cụm từ có thể xuất hiện trong bản rõ gốc ứng với bản mã, sau đó tìm cách xác định khoá đúng. Nếu khoá là đúng thì có thể dịch bản mã để cho ra bản rõ...

Ngoài một số phương pháp truyền thống trên, ngày nay nhờ tốc độ máy tính đã được cải thiện đáng kể, trong những bài toán mà không gian khoá không quá lớn người ta còn có thể áp dụng một phương pháp nữa đó là 'vét cạn'. Đối với không gian khoá lớn, đây thật sự là phương pháp tồi nên chúng ta chỉ thực hiện 'vét cạn' một cách thông thường. Tuy nhiên nếu áp dụng đồng thời các kỹ thuật bổ trợ thì nó vẫn phát huy được hiệu quả tốt. Các kỹ thuật hỗ trợ được nói tới ở đây là xây dựng một thư viện phục vụ việc 'vét cạn' bao gồm cơ sở dữ liệu về khoá và các tiêu chuẩn bản rõ tốt. Trên cơ sở đó tìm cách phân hoạch không gian khoá S thành hai tập con rời nhau là S_1 vàS_2 sao cho khoá đúng sẽ 'chắc chắn' thuộc một trong hai tập con đó. Từ đó tiến hành sử dụng thuật toán vét cạn trên tập con có chứa khoá đúng, khi đó việc vét cạn trong tập con nhanh chóng thể hiện tính hiệu lực của nó. Việc này cũng có thể thực hiện ngay đối với một số phương pháp truyền thông đã có được những kết quả đáng ngạc nhiên. Khi thám mã ra bản rõ ta chỉ cần đọc được lỗ chỗ đã là thành công vì lúc đó bằng quy luật ngôn ngữ ta sẽ khôi phục được bản rõ gốc như mong muốn.

Chú ý: Ngày nay, người ta đã có những công cụ tính toán cực nhanh nhờ công nghệ cluster. Từ đó người ta có thể xây dựng mạng tính toán song song với tốc độ tính toán đạt tới gần 100GF (một trăm tỉ phép tính dấu phảy động trên một giây). Như vậy người ta có phân rã bài toán để thực hiện việc tính toán song song cực kỳ có hiệu quả, đặc biệt đối với những bài toán có độ phức tạp tính toán lớn.

2.2. Mã thay thế đơn và phương pháp thám mã.

2.2.1 Mã thay thế đơn.

Thay thế đơn là luật mã tương đối đơn giản. Nó ra đời từ trước chiến tranh thế giới lần thứ nhất, được duy trì và phát triển mãi cho đến khoảng những năm 70 của thế kỉ XX. Tuy nhiên loại mật mã này hiện nay vẫn được sử dụng lác đác đâu đó chủ yếu là vì mật mã này đơn giản, việc mã/giải không cần đến máy tính, rất phù hợp cho những người không hiểu biết nhiều về mật mã thường đi công tác lẻ nơi không có điện.

Ta hiểu "thay thế đơn" nghĩa là cứ một chữ cái trong bản thông báo được thay thế (substitution) bởi duy nhất một ký hiệu nào đó. Ký hiệu này có thể là chữ cái, chữ số, hoặc một dấu hiệu nào đó được người gửi và người nhận đích thực qui ước với nhau trước.

Thí dụ: Hanoi ↔ $\pi O \wedge \Omega \Sigma$

Về phạm trù toán học, chúng ta có thể định nghĩa mật mã thay thê đơn là một ánh xạ lên 1-1 (tức f là một song ánh): $f: \mathcal{A} \rightarrow \mathcal{B}$ trongđó, $\mathcal{A}$ là không gian bản rõ (thường là bảng chữ cái La-tinh),$\mathcal{B}$ là không gian bản mã tương ứng. Điều này có nghĩa là cứ mỗi $a \in \mathcal{A}$ tồn tại duy nhất b ∈ $\mathcal{B}$ sao cho $f(a) = b$. Do đó, xét về tập hợp thì lực lượng của 2 tập hợp $\mathcal{A}$ và $\mathcal{B}$ là như nhau. Vì vậy $\mathcal{B}$ là tập hợp các ký hiệu gì không quan trọng. Trong phạm vi nghiên cứu này, chúng ta giả thiết tập $\mathcal{B} = \mathcal{A} = \{a, b, c, \dots, z\}$

Thí dụ bản rõ Vieetjnam -» MNZZU KJCO

(Các ký tự mã thường được viết hoa, còn bản rõ là chữ thường). Trong đó chữ V (của bản rõ) được thay bởi chữ M, chữ i trong bản rõ được thay thế bởi chữ N,v.v.

Ở phần này chúng ta không phân tích chi tiết kỹ thuật mã hóa và giải mã (khi đã cho trước khóa) mà sẽ nghiên cứu các phương pháp thám mã (giải bản

mã khi không cho trước khóa mã). Tức là xây dựng các phương pháp chuyển bản mã (không đọc được) thành bản rõ gốc khi không có khoá mã trong tay.

2.2.2. Phương pháp thám mã.

2.2.2.1. Thám mã thủ công.

Giả thiết trên cơ sở nào đó, ta nhận được một bản mã sau đây (từ một đối tượng nói tiếng Anh)

XDWXA ZRSOP LSNIW KRVSD DPAX

RPDSS BUSTP UKDXS AKTKD ƯSCLS

KDXRQ SCCSF XAVBR PEPUI TCKMP

FPVS./.

Độ dài bản mã là *n* - 79 (ký tự)

Phương pháp thám mã như sau:

Bước 1: Tính tần số đơn của bản mã trên ta có bảng tần số đơn sau đây:

A:4	J:0	S:12
B:2	K:6	T:3
C:4	L:2	U:4
D:7	M:1	V:3
E:1	N:1	W:2
F:3	0:1	X:6
G:0	P:8	Y:0
H:0	Q:1	Z:1
\:2	R:5	

Qua bảng tần số trên ta thấy có 21 chữ cái được dùng để mã hóa.

Vì bản mã ngắn (79 ký tự) nên nếu đối tượng dùng tiếng Anh thì như vậy cũng hợp lý.

Ta thấy có 9 ký tự cao tần nhất. Đó là:

S:12

P: 18

D: 7

K: 6

X: 6

R: 5

A: 4

C: 4

U: 4

Đối chiếu với tần số đặc trưng ngôn ngữ tiếng Anh tự nhiên, 9 ký tự cao tần nhất trong ngôn ngữ chuẩn tiếng Anh là:

ETAOINRSH

Trong tiếng Anh tần số đặc trưng cao tần nhất là chữ cái e, vì vậy ta thay thế chữ s trong bản mã bởi chữ E của bản rõ, chữ t cho chữ p, chữ d cho chữ A, i thay cho X, n thay cho R và viết vào dòng dưới của các ký tự mã tương ứng trong bản mã, ta có:

XDWXA i a i	ZRSOP n e. t	LSNIVV e	KRVSD o n. e a	DPAFX a t i
RPDSS n t a ee	BUSTP . . e . t	UKDXS o a i e	AKTKD ooa	USCLS e e
KDXRQ o a i n	SCCSF e	XAVBR I n	PEPUI t .t	TCKMP ot
FPVS t. e				

Nhìn vào các ký tự đã được thay thế ở trên, ta thấy có điều gì đó chưa hợp lý. Ví dụ, trong tiếng Anh, 3 nguyên âm đi liền nhau như oai là rất hiếm. Ở đây, thậm chí 4 nguyên âm đi liền nhau là oaie...

Do đó giả thiết của ta có thể có ký tự ta thay chưa đúng. Ký tự mã X có thể là ký tự I là hợp lý. Vậy D cao tần trong bản mã có thể là t trong bản rõ (chứ

không phải là P) và chữ p có thể là chữ e và s có thể là chữ o trong bản rõ. Ta có:

XDWXA it i	ZRSOP n oe	LSNIW o	KRVSD an o t	DPAFX t e i
RPDSS n e t oo	BUSTP o e	UKDXS a t i o	AKTKD a a t	USCLS e o
KDXRQ a t i n	SCCSF oo	XAVBR i n	PEPUI e e	TCKMP e
FPVS e				

Qua đây ta thấy A = n chứ không phải R = n. Do đó ta xóa ký tự n vừa thay và ta thay A bởi n rõ và cứ tiếp tục thay, đọc xem có gì mâu thuẫn không cho đến khi ta khôi phục lại được bản rõ và sau đó thiết lập khóa mã là xong.

Cuối cùng ta có khóa mã như sau (dòng trên ứng với ký tự mã, dòng dưới ứng với ký tự rõ (viết thường).

A	B	C	D	E	F	G	F	G	F	G	L	M	N	O
n	u	1	t	v	w		w		w		b	c	d	m
P	Q	R	S	T	U	V	U	V	U	V				
e	f	s	o	p	r	g	r	g	r	g				

Nếu sắp xếp theo thứ tự dòng trên là biểu thị ký tự rõ, dòng dưới là ký tự mã (khóa) tương ứng, ta có:

a	b	c	d	e	f	g	h	i	j	k	i	m	n	o
K	L	M	N	P	Q	V	W	X	Y	Z	C	O	A	S
p	q	r	s	t	u	v	w	x	y	j				
T	.	U	R	D	B	E	F	.	I	J				

Vậy từ khóa thay thế là COAST.

Do bản mã ngắn quá nên chúng ta không thể khôi phục lại đầy đủ khóa mã, còn chữ q và X trong bản rõ không biết được mã bởi ký tự gì (lần lượt có thể là G, H nhưng cũng có thể là H, G).

2.2.2.2. Thám mã với sự trợ giúp của máy tính PC.

Có một số phương pháp thám mã thay thế đơn chữ cái. Mỗi phương pháp có những ưu và nhược điểm của nó, và hầu như phương pháp nào cũng không thể tự động hóa thám mã 100% mà phải có sự trợ giúp của con người. Một trong những phương pháp đáng lưu ý là phương pháp dựa trên giải thuật di truyền, chúng ta không trình bày nội dung của giải thuật di truyền (Genetic Algorithms - GA).Ở đây, tôi chỉ trình bày ứng dụng của nó vào thám mã thay thế đơn chữ cái. Nội dung của phương pháp này được tiến hành theo các bước lớn sau đây:

a. Biểu diễn khóa.

Ta biết rằng: một khóa là một dãy gồm 26 chữ cái. Để phục vụ việc thám mã các khóa được sắp xếp theo thứ tự tần suất giảm dần.

Thí dụ: một khóa có dạng:

CDEFGHIJKLMNOPQRSTUVWXYZAB chỉ ra rằng chữ cái có tần số xuất hiện cao nhất trong bản mã (ứng với khóa đó) là chữ c, chữ cái có tần số xuất hiện cao thứ 2 là chữ D, cao thứ ba là chữ E... và cuối cùng chữ cái có tần số xuất hiện thấp nhất trong bản mã ứng với khóa đó là chữ B.

Biểu diễn như vậy cũng ngầm định là thay thế tương ứng với các chữ cái theo trật tự định trong bản rõ là:

TAOINHSRDLUMWXCGFYPBKVJXQZ.

Tức là chữ E trong bản rõ sẽ được thay cho chữ cái c trong bản mã. Với sự biểu diễn này thì không gian khóa sẽ là: $26! \approx \left\{\frac{26}{e}\right\}^{26} \sqrt{2 \cdot \pi \cdot 26}$, Trong đó $e \approx 2{,}71828\ldots$ Hay $26! \approx (9{,}6)^{26} \cdot 13$và do đó công việc tìm kiếm khóa ngẫu nhiên(vét cạn) là không khả thi. Tuy nhiên thuật toán di truyền lại thực

hiện thử và sai (Trial and error) theo một cách ngẫu nhiên có lựa chọn. Để thực hiện điều này trước hết người ta đưa vào một hàm đo sự phù hợp và được định nghĩa là:

$fitness = \{1 - \sum_{i-1}^{26}[|SF[i] - DF[i]| + \sum_{j-1}^{26}|SDF[i,j] - DDF[i,j]|] / 4\}^8$ (2.1)

Công thức này do R.spillman đưa ra

Trong đó:

SF = (*SF*[1], ..., *SF*[26]) là tần suất đơn tiếng Anh chuẩn (tức tần số đơn đã tính ra tỉ số phần trăm).

$SDF = (SDF[i,j])_{i,j=\overline{1,26}}$ là bảng tần suất bộ đôi móc xích tiếng Anh chuẩn đã được tính như sau:

$$SDF[i,j] = \frac{m_{ij}}{M_i}$$

Với m_{ij} là tần bộ đôi móc xích bộ (i, j); i, j = 1,2, …,26

Còn $M_i = \sum_1^{26} m_{i,j}; i = 1,2,\dots 26$

DF = (*DF*[1], ..., *DF*[26])

DDF = $(DDF[i,j])_{i,\,j=1,2,\dots,26}$

cũng được ký hiệu và tính hoàn toàn tương tự như *SF* và *SDF* ở trên, chỉ có khác là chúng được tính trên bản mã cần tính (tức là bản rõ sau khi đã được giải bởi một "khóa" nào đó).

Quá trình xử lý như sau:

1. Dùng một khóa nào đó để giải mã bản mã gốc (tức là bản mã mà mã thám viên nhận được từ đầu)
2. Kết quả giải mã ở bước 1, được đem vào phân tích tần số đơn và tần số bộ đôi móc xích (chuyển sang tần suất)
3. Kết quả phân tích tần số (tần suất) được so sánh với tần số tiếng Anh chuẩn để xác định độ sai lệch bằng công thức (2.1).

4. khóa đúng sẽ là khóa ứng với "bản rõ" có độ sai lệch so với tần suất chuẩn tiếng Anh là bé nhất, tức là có độ phù hợp *fitness* lớn nhất.

b. Trao đổi chéo.

Một thế hệ là một tập các khóa, mỗi khóa có tương ứng với một độ phù hợp. Giải thuật được tiến hành bằng cách chọn ngẫu nhiên hai khóa để làm trao đổi chéo. Hai khóa này được gọi là khóa bố và khoá mẹ, hay gọi là cặp khóa bố mẹ.

Cặp khóa bố mẹ này sẽ sinh ra hai khóa con nhò việc trao đổi chéo. Việc trao đổi chéo này liên quan đến việc rà quét các thành phần của hai khóa từ hai đầu và chọn ra chữ cái có tần số xuất hiện cao hơn sẽ được đưa vào "con".

Thí dụ: lấy 2 khóa bố mẹ là:

bố: CDEFGHIJKLMNOPQRSTUVWXYZAß

mẹ: FGHIJKLMNOPQKSTUVWXYZABCDE

bố 1: CDEFGHIJKLMNOPQRSTUVWWXYZAB

mẹ1: FGHIJKLMNOPQRSTUVWWXYZABCD E

Chữ F xuất hiện nhiều hơn trong "bản mã" so vối chữ c nên con 1 sẽ có chữ cái đầu tiên là F.

con 1 FD……E

Ký tự thứ 2 của con 1 bằng cách so sánh 2 chữ cái tiếp theo của bô, mẹ là D và G. Giả sử chữ D cao tần hơn, lúc đó ký tự thứ 2 của con 1 là chữ D... Tiếp tục như vậy cho đên 2 chữ cuối cùng là B và E. Giả sử đó là chữ E và giả sử ta có đứa con 1 là:

con l: FDEIJHLMKOPQRSTUVWWXYZABCDE

Quá trình này được lặp lại để tạo ra khóa thứ 2 (con 2) bằng cách quét ngược lại từ phải qua trái. Trong trường hợp này ta xét cặp B, E (bố mẹ) trước và cuối cùng xét cặp C, F. Giả sử ta có kết quả tạo ra đứa con 2 như sau:

con2: CDVFGHIJKLMNOPQRSTUVWXBZAE

c. Quá trình đột biến.

Sau khi thế hệ mới được tạo ra, các khóa phụ thuộc vào quá trình biến đổi (đột biến) với tần số thấp, ở đây khi một ký tự (chữ cái) trong khóa được chọn để biến đổi thì ký tự đó được trao đổi với một ký tự nào đó được chọn ngẫu nhiên trong dãy khóa nếu giá trị fitness của dãy khóa thuộc nửa dưới của gene pool (của họ). Nếu giá trị fitness của dãy khóa thuộc vào nửa trên của họ thì ký tự đó được trao đổi với ký tự đứng ngay bên phải nó.

Thí dụ: Trong hình dưới mô tả lý luận vừa nêu: Ký tự D đứng ở nửa trên của dãy, nên nó được đột biến (biến đổi) bằng cách trao đổi nó với ký tự đứng ngay bên phải nó (ở đây là chữ cái "E") để sinh ra dãy dưới:

Nửa trên của họ	Nửa dưới của họ
QAZXSWEDCRFVT	GBYHNUJMIKOLP
QAZXSWDECRFVT	GBYHNUJMIKOLP

d, Giải thuật được hoàn thành

Quá trình trên được kết hợp với nhau để tạo nên một giải thuật hoàn chỉnh. Các bước của giải thuật là:

1. Một họ ngẫu nhiên của các dãy khóa được sinh ra.
2. Một giá trị fitness đối vối mỗi khóa trong họ được xác định.
3. Việc lựa chọn ngẫu nhiên dựa trên giá trị fitness của bố mẹ được dẫn ra.
4. Thao tác trao đổi chéo được áp dụng đôi với bố mẹ đã được lựa chọn.
5. Quá trình đột biến được áp dụng đối với các con.
6. Một họ mới gồm các dãy khóa được quét và được sử dụng để cập nhật thành một danh sách khoảng 10 khóa tốt nhất qua các thế hệ.

Quá trình này sẽ kết thúc sau một số lượng nhất định các thế hệ và các khóa tốt nhất sẽ được sử dụng để giải bản mã.

2.3. Luật mã CAESAR và phương pháp thám.

2.3.1. Khái quát.

Mã Caesar là hệ mã truyền thống, được cho dưới dạng sau đây:

*Mã hóa: Giả sử R là bản thông báo cần mã (gọi là bản rõ) còn M là văn bản đã được mã hóa, khi đó ta có công thức mã hóa như sau: $M = R®K$, trong đó: phép cộng ® được thực hiện theo mô-đun26. Cụ thể, giả sử: $R = r_1 r_2 \dots r_n$ với: ki $r_i \in \{a, b, ..., z\}$, K là một khóa mã cho trước: $K \in \{1, 2, ..., 26\}$. Còn $M = m_1 m_2 \dots m_n$. Lúc đó ta có hệ thức mã hóa: $m_i = r_i + K (\text{mod } 26)$, $i = \overline{1,n}$

Như vậy để thực hiện phép cộng thông thường theo mô đun 26, người ta cần chuyển ký tự rõ r_i sang chữ số từ 0÷25theo qui tắc sau đây:

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

Hoặc

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
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	26

**Phương pháp mã dịch vòng:*

Phương pháp giải mã được thực hiện theo công thức:

$r_i = m_i - k (\text{mod } 26)$; $i = 1, 2, \dots, n$

Thí dụ: người ta muốn mã bản rõ

R = coongj hoafxaxhooijchurnghiaxvieetjnam

Có khóa $k = 7$. Người ta tiến hành như sau

Chuyển bản rõ R sang chữ số, sau đó viết khóa xuống dưới các kí tự tương ứng ta có:

R:	2	14	14	13	6	9	7	14	0	5	23	0	23	7
K:	7	7	7	7	7	7	7	7	7	7	7	7	7	7

R:	14	14	8	9	2	7	20	17	13	6	7	8	0	23
K:	7	7	7	7	7	7	7	7	7	7	7	7	7	7

R:	21	8	4	4	20	9	13	0	12					
K:	7	7	7	7	7	7	7	7	7					

Tiến hành mã hóa bằng cách cộng theo mô đun 26 ký tự ứng với từng cột với nhau, sau đó chuyển về chữ cái, và có:

R:	2	14	14	13	6	9	7	14	0	5	23	0	23	7
K:	7	7	7	7	7	7	7	7	7	7	7	7	7	7
M	J	V	V	U	N	Q	O	V	H	M	E	H	E	C

R:	14	14	8	9	2	7	20	17	13	6	7	8	0	23
K:	7	7	7	7	7	7	7	7	7	7	7	7	7	7
M	V	V	P	Q	J	O	B	Y	U	N	O	P	H	E

R:	21	8	4	4	20	9	13	0	12					
K:	7	7	7	7	7	7	7	7	7					
M	C	P	L	L	B	Q	U	H	T					

Vậy ta có bảng mã:

M = JWUN QOVHM EHEOV VPQJO BYUNOPHECP LLBQƯ HT./.

*Giải mã (*thường gọi là dịch mã*)

Muốn giải bản mã này sang bản rõ khi có khóa tương ứng là rất đơn giản: ta chỉ việc lấy bản mã sau khi đãđược chuyển sang chữ số tương ứng đem trừ đi khóa theo mô đun 26.

M	J	V	V	U	N	Q……
K:	7	7	7	7	7	7……
R:	2	14	14	13	6	9…….

Chuyển R ở dạng số về chữ cái tương ứng, ta có:

2 → C

14 → O

13 → N

6 → G

9 → J

và kết quả là:coongjhoafxaxhooijchurnghiaxvieetjnam

2.3.2. Phương pháp thám mã

2.3.2.1. Thám mã thủ công

*Nhận xét: Mã Caesar thực ra là mã thay thế đặc biệt, nói cách khác nó là phương pháp chuyển dịch các ký tự của bản rõ tăng lên một hằng số không đổi theo mô đun 26. Nếu ta tìm được hằng số đó (tức là khóa) thì coi như bản mã đó đã bị phá. Nhìn vào bản mã ta thấy rằng nếu 2 ký tự của bản rõ bằng nhau mà đứng cạnh nhau thì 2 ký tự mã tương ứng cũng bằng nhau.

Thí dụ: oo → vv (nếu khóa $k = 7$)

Như vậy 2 ký tự đồng tự trong bản rõ sẽ cho ta 2 ký tự đồng tự tương ứng trong bản mã. Căn cứ vào tần số bộ đôi đồng tự trong bản mã để dự đoán bộ đôi đồng tự tương ứng trong bản rõ và từ đó dự đoán được hằng số. Lấy hằng số (khóa) này đem giải mã cho các ký tự khác, nếu kết quả thành đoạn văn có nghĩa thì coi như cứ giải tiếp, nếu trái lại thì ta thay đổi giả thiết. Cứ như vậy sau một lúc nào đó chúng ta sẽ xác định được khóa mã.

Ngoài ra ta cũng nên đưa vào các từ giả định để dự đoán và tìm ra khóa *K*. Hơn nữa, ký tự cao tần nhất trong bản rõ sẽ cho ta ký tự cao tần trong bản mã tương ứng. Khi dự đoán được khóa *K* thì ta tiến hành giải thử các ký tự khác của bản mã xem có cho ta đoạn văn có ý nghĩa hay không.

Về nguyên lý, khóa K chỉ có giá trị từ 1 đến 26 nên với thời gian chấp nhận được, ta sẽ tìm ra khóa mã một cách không khó khăn lắm.

Từ nhận xét trên đây, chúng ta có phương pháp thám mã như sau:

1. Giả sử trên cơ sở nào đó chúng ta nhận được bản mã:

M = m1m2 ..... mn, Trong đó:

$$m_i \in \{a, b, \dots, z\}, i = 1, 2, 3, \dots, nn \geq 50$$

2. Tiến hành tính tần số đơn bản mã, giả sử sau khi tính tần số đơn bản mã, ta có kết quả: $f = (f_{13}\ f_{2?} \dots..,\ f_{26})$, $f_j \geq 0$ là tần số ký tự thứ i của bản mã (i = 1, 2 ,..., 26).

3. Đánh dấu những cặp (bộ đôi) đồng tự xuất hiện trong bản mã.

4. Dự đoán ngôn ngũ bản rõ tương ứng với bản mã.

Thông thường nếu tổng số bộ đôi đồng tự chiếm khoảng 0,067 ÷ 0,072 thì bản rõ tương ứng có nhiều khả năng là tiếng Anh. Nếu tỷ lệ đó chiếm trên 0,08 thì rất có khả năng bản rõ là tiếng Việt có dấu được viết theo lối điện báo (telex).

5. Thám đột phá.

Căn cứ qui luật ngôn ngữ với biểu hình cao tần đồng tự trong bản mã để dự đoán.

6. Sau khi xác định được khóa ta dịch (deciphering) thử với các ký tự khác xem có hợp lý không.

7. Giải bản mã và đọc

Thí dụ với bản mã trên:

M = JWUN QOVHM EEHOV VPQJO BYUNO PHECP LLBQU HT./.

Giải: ta thấy rằng bản mã này có độ dài 37 (37 kí tự). Do đó số lượng các bộ đôi có thể có là 36.

Ta cũng thấy ngay rằng có 3 bộ đôi đồng tự w (xuất hiện 2 lần, LL xuất hiện một lần)

Ta có:

3/36 = 0,0833 > 0,08 => bản mã rất có thể là mã Caesar với bản rõ tiếng Việt được viết theo kiểu Telex. Trong tiếng Việt bộ đôi đồng tự đi liền nhau

phần lớn là nguyên âm a, e,o. Ở đây có thể là aa, ee hoặc oo. Nhưng trong bảng tần số bộ đôi thì ee, oo xuất hiện nhiều hơn là aa. Vậy ta giả thiết coi vv = ee.

Ta lấy v - e(mod 26) = 21 - 4(mod 26) = 17

Bây giờ lấy J -17 (J là ký tự đứng ngay trước w ở nhóm đầu tiên của bản mã). Ta có:

J - 17 (mod 26) = (9 + 26 - 17)(mod 26) = 18 = S

Ta tiếp tục xét ký tự U đứng ngay sau VV, ta có:

U - 17(mod 26) = 20 - 17(mod 26) = 3 = D

Như vậy là nếu khóa k = 17 thì 3 ký tự đầu tiên được giải ra là SED. Trong tiếng Việt chúng không có nghĩa. Vậy giả sử ta coi VV = OO. Ta có V - O(mod 26) = 21 - 14(mod 26) = 7. Tiếp tục thử như trên với khóa được giả định là 7 ta có:

J - 7(mod 26) = 2 = C và

V - 7(mod 26) = 21 - 7 = 14 = O tiếp theo

V - 7(mod 26) = 21 - 7 = 14 = O chúng ta thấy rất hợp lý với tiếng Việt. Tiếp tục dịch (giải mã) bởi khóa K = 7 cho toàn bộ bản mã, ta nhận được bản rõ là:

"coongjhoafxaxhooijchurnghiaxvieetjnam".

2.3.2.2. Thám mã với sự trợ giúp của PC.

Giả sử trên cơ sở nào đó ta nhận được một bản mã được mã bởi mã pháp Caesar: $M = m_1m_2m_3 \ldots m_n$

Giả sử ngôn ngữ được dự đoán là tiếng Anh chẳng hạn (ngôn ngữ nào không quan trọng nhưng phải biết cụ thể nó là loại ngôn ngữ gì).

Ta tiến hành các bước thám sau đây:

Bước 1. Tính tần số đơn bản mã M. Giả sử ta nhận được kết quả tần số đơn của bản mã M là:

$$f = (f_1, f_2, \ldots, f_{26}); \; f_i \geq 0, \; i = 1, 2, 3, \ldots, 26$$

Bước 2: Chọn 10 chữ cái ứng với tần số cao nhất của bản mã. Ta ký hiệu 10 chữ cái cao tần nhất trong bản mã được xếp theo thứ tự giảm dần là:

$$\{r_1, r_2, r_3, ..., r_{10}\} = G_M$$

Ta biết 10 chữ cái cao tần nhất của ngôn ngữ chuẩn tiếng Anh được sắp xếp theo thứ tự giảm dần là:

$$\{E, T, A, I, N, O, R, S, H, D\} = G_0$$

Bước 3: Tìm J_0 trong khoảng $0 \div 25$ sao cho:

$$card\{(G_M - J_0) \cap G_0\} \max_{0 \le j \le 25} \{(G_M - J) \cap G_0\}$$

Trong đó: $\big((G_M - J_0) = (r_1 - J, r_2 - J, \dots, r_{10} - J)(mod\ 26)\big)$

Bước 4: Sau khi xác định được J_0, ta chỉ việc giải bản mã bình thường:

$$R = M - J_0 (\text{mod } 26)$$

Trong chương này chúng ta đã đi tìm hiểu về các cách tấn công hệ mã truyền thống, cụ thể đã tìm hiểu nguyên lý và cách thám hệ mã CAESAR. Nhìn chung ta thấy các hệ mã bảo mật đều có chức năng bảo vệ cho thông tin không bị khai thác bất hợp pháp, chống lại các tấn công sau:

- Thám mã thụ động: bao gồm các hoạt động:
  + Thu chặn.
  + Dò tìm.
  + So sánh tương quan.
  + Suy diễn.
- Thám mã tích cực: bao gồm các hoạt động:
  + Giả mạo.
  + Ngụy trang.
  + Sử dụng lại.

+ Sửa đổi.

Phân tích mã là khoa học nghiên cứu cách phá các hệ mật nhằm phục hồi bản rõ ban đầu từ bản mã. Việc tìm hiểu các thông tin về khóa và các phương pháp biến đổi thông tin cũng là một nhiệm vụ quan trọng của phân tích mật mã. Có ba phương pháp tấn công cơ bản của thám mã:

- Tìm khóa vét cạn.
- Phân tích thống kê.
- Phân tích toán học.

Việc tấn công của thám mã có thể được thực hiện với các giả định:

- Tấn công chỉ với bản mã.
- Tấn công với bản rõ đã biết.
- Tấn công với các bản rõ được chọn.
- Tấn công với các bản mã được chọn

CHƯƠNG 3: ĐỀ XUẤT THUẬT TOÁN CẢI TIẾN NHẰM NÂNG CAO ĐỘ AN TOÀN CHO HỆ MẬT MÃ TRUYỀN THỐNG

3.1. Mục đích ý nghĩa

Hiện tại mã chuyển vị một lần đã có phương pháp thám bằng phương pháp thủ công hoặc tự động trên máy tính. Do đó, sau khi tìm hiểu ta thấy rằng trong thám mã, không gian khóa và độ phức tạp của thuật toán đóng vai trò quan trọng: Vì vậy bài toán đặt ra là cần tìm cách tăng lượng không gian khóa và tăng độ phức tạp của thuật toán. Đối với mật mã chuyển vị đơn, người ta thám mã bằng cách dự đoán độ dài khóa chuyển vị và tìm cách ghép các cột nhờ qui luật ngôn ngữ tự nhiên. Vậy nếu ta tìm cách xóa được qui luật ngôn ngữ thì coi như chúng ta đã thành công nhằm đối phó với khả năng "ghép cột" của loại mật mã này. Để giải quyết được vấn đề này, một phương pháp đơn giản là mã chuyển vị hai lần (chuyển vị kép).

3.2. Đề xuất thuật toán.

Sơ đồ các hệ mật mã được định nghĩa như sau:

$$S_1 = (P , C_i , K_i , E_i , D_i)$$

Trong đó, P là tập các ký tự bản rõ,C_1 là tập các bản mã, K_i tập tất cả các hoán vị của các số nguyên { 1, 2, ... , l_i }, E_i là thuật toán mã hóa, D_i thuật toán giải mã.

sao cho $C_1 = E_{(k1)}(P)$ với $k_1 \in K_1$

Bây giờ ta xây dựng Hệ mật mã thứ hai là :

$S_2 = (C_1 , C_2 , K_2 , E_2 , D_2)$, trong đó,

$C_2 = E_{k2}(C_1)$ với $k_2 \in K$ = { hoán vị {1, 2, ..., l_2}}, nói chung $l_2 \neq l_1$.

Quá trình mã hóa như sau:

Cho bản rõ $P = P_1P_2 ... P_n$,với độ dài là n

Mã hóa lần 1:

Bước 1. Chọn l_1 lớn ta có khóa k_1 là dãy số hoán vị của dãy số có thứ tự {1, 2, 3.. l_1}

$k_1 = \{k_{11}, k_{12}, ..., k_{1l1}\}$

Bước 2. Lập bảng với kích thước $l_1 * n_1/l_1$ nếu n chia hết cho l_1 và bằng $l_1 * (n_1/l_1 + 1)$ nếu n không chia hết cho l_1.

Bước 3. Nhập các ký tự $P_1, P_2, ..., P_n$ tương ứng các ô theo thứ tự tự nhiên cho đến hết bản thông báo P.

Bước 4. nhặt các ký tự của bảng theo thứ tự từ trên cột từ trên xuống dưới và từ cột có chỉ số thấp nhất đến cột cao nhất cuối cùng, ta được:

$C_1 = c_{11} c_{12} c_{13} .. c_{1i}$

Mã hóa lần 2:

Coi C_1 là bản rõ mã hóa tương tự như trên với l_2

Bước 1. Chọn l_2 là số tự nhiên tùy ý $l_2 \neq l_1$, ta có khóa k_2 là một hoán vị của dãy số $\{1, 2, 3, ..., l_2\} = K_2$, $k_2 = \{k_{21}, k_{22}, ..., k_{2l2}\}$.

Bước 2. Lập bảng với kích thước $l_2 * n/l_2$ nếu n chia hết cho l_2 và bằng $l_2 * (n/l_2 + 1)$ nếu n không chia hết cho l_2.

Bước 3. Nhập các ký tự $C_1, C_2, ..., C_n$ theo thứ tự tự nhiên vào các ô của bảng cho ở bước 2, Bước 4. Nhặt các ký tự của bảng cho ở bước 3 theo thứ tự từ trên xuống và từ cột có chỉ số thấp nhất đến các cột có chỉ số cáo nhất,ta được bản mã cuối cùng là:

$C_2 = c_1 c_2 c_3, ..., c_n$

Thí dụ: Có bản rõ

P = coongj hoaf xax hooij chur nghiax Vieetj Nam (ở đây độ dài bản thông báo là n = 37)

Quá trình mã hóa như sau:

Ban đầu chọn $l_1 = 8$, ta có khóa k_1 chẳng hạn là:

$k_1 = \{3, 1, 2, 6, 5, 4, 7, 8\}$.

Lập bảng: Do số ký tự của chuỗi là 37 ký tự mà ta có tất cả là l_1* (4+1) = 8*5 = 40. Vậy bảng này có 5 hàng thừa 3 ô cuối được bỏ để trống và 8 cột:

	3	**1**	**2**	**6**	**5**	**4**	**7**	**8**
1	c	o	o	n	g	j	h	o
2	a	f	x	a	x	h	o	o
3	i	j	c	h	u	r	n	g
4	h	i	a	x	v	i	e	e
5	t	j	n	a	m			

Bản mã số 1:

C_1 = OFJIJ OXCAN CAIHT JHRIG XUVMN AHXAH ONEOO GE

Bây giờ coi bản mã số 1 là bản rõ

P_1 = OFJIJ OXCAN CAIHT JHRIG XUVMN AHXAH ONEOO GE.

Để mã hóa lần hai, ta chọn l_2 = 9, ta chọn khóa $k_2$là :

k_2 = {1, 5, 3, 6, 7, 9, 2, 4, 8}.

Tiếp tục lập bảng với k_2 = { 1, 5, 3, 6, 7, 9, 2, 4, 8} cột và 5 hàng :

Do số ký tự của chuỗi là 37 mà ta có tất cả là l_2* (37/9 +1)= 9*5 = 45 (8 ô cuối cùng của bảng được bỏ trống). Nhập các ký tự của P_1 vào bảng, ta nhận được kết quả:

	1	**5**	**3**	**6**	**7**	**9**	**2**	**4**	**8**
1	O	F	J	I	J	O	X	C	A
2	N	C	A	I	H	T	J	H	R
3	I	G	X	U	V	M	N	A	H
4	X	A	H	O	N	E	O	O	G
5	E								

Bản mã số 2 và là bản mã cuối cùng là:

C = ONIXE XJNOJ AXHCH AOFCG AIIUO JHVNA RHGOT ME

Quá trình giải mã:

Với bản mã C = ONIXE XJNOJ AXHCH AOFCG AIIUO JHVNA RHGOT ME.

Với các khóa: k_1= {3, 1, 2, 6, 5, 4, 7, 8} và k_2 = {1, 5, 3, 6, 7, 9, 2, 4, 8},

Bước 1: Lập bảng với khóa k_2 và n_2 = 5 (n_2 = 37/9 + 1)

Sắp xếp các ký tự theo cột với thứ tự của khóa

	1	**5**	**3**	**6**	**7**	**9**	**2**	**4**	**8**
1	O	F	J	I	J	O	X	C	A
2	N	C	A	I	H	T	J	H	R
3	I	G	X	U	V	M	N	A	H
4	X	A	H	O	N	E	O	O	G
5	E								

Từ bảng ta lấy chuỗi ký tự theo hàng theo thứ tự của hàng có được bản rõ P_1 := OFJIJ OXCAN CAIHT JHRIG XUVMN AHXAH ONEOO GE. Coi bản rõ P_1 là bản mã để giải với khóa k_1 ta lập bảng theo thứ tự của khóa:

	3	**1**	**2**	**6**	**5**	**4**	**7**	**8**
1	C	O	O	N	G	J	H	O
2	A	F	X	A	X	H	O	O
3	I	J	C	H	U	R	N	G
4	H	I	A	X	V	I	E	E
5	T	J	N	A	M			

Từ bảng ta nhặt theo thứ tự hàng thì nhận được bản rõ P, cụ thể là :

P = coongj hoaf xax hooij chur nghiax Vieetj Nam

3.3. Đánh giá độ an toàn của hệ mật mã được đề xuất

Như đã đề xuất ở trên thuật toán được xây dựng dựa trên hệ mã chuyển vị, nhưng ở đây đã được cải tiến thành hai lần mã hóa tương ứng với hai lần chuyển vị (chuyển vị kép) vậy về mặt lý thuyết đã tăng độ an toàn so với hệ mã cũ lên gấp hai lần, khắc phục được sự phát hiện quy luật của bản rõ. Không những thế khóa của hệ mật mã mới là một cặp mã hoàn toàn khác nhau, mỗi khóa lại là một sự hoán vị bất kỳ của một dãy số (đủ lớn). Như vậy không gian khóa đã tăng lên rất nhiều. Độ dài mã một lần có số lượng khóa có thể là $L_1!$, thuật toán cải tiến có độ dài khóa là $L_1! \cdot L_2!$. Trong đó L! = 1. 2. 3 ... L. Để thám mã sử dụng phương pháp vét cạn với các độ dài khóa L_1 và L_2 đủ lớn thì khó có thể thực hiện được, kể cả sử dụng công cụ tính toán hiện đại.

3.4. Cài đặt kiểm thử

3.4.1 Giới thiệu thuật toán

a, Thuật toán mã hóa

Đầu vào:

- File văn bản P theo định dạng txt
- l1, l2 là độ dài của khóa k1, k2

Đầu ra:

- File văn bản C đã được mã hóa
- File Key.dat chứa hai khóa k1, k2, l1, l2, n1, n2

Các bước thực hiện:

Bước 1:

- Chọn file văn bản cần mã hóa
- Chọn l1, l2, sinh ra khóa k1, k2, n1, n2

Bước 2: mã hóa

- *Mã hóa lần 1:*

+ Chuyển đổi văn bản thành mảng hai chiều có kích thước [l1, n1] theo thuật toán lập bảng trình bày ở trên

+ Chuyển đổi mảng thành chuỗi theo khóa k1

- *Mã hóa lần 2:*

+ Chuyển đổi chuỗi ở kết quả mã hóa lần 1 thày mảng hai chiều có kích thước [l2, n2]

+ Chuyển đổi mảng thành chuỗi theo khóa k2

Bước 3: lưu kết quả

- Lưu chuỗi mã hóa lần 2 thành file có phần mở rộng txt

- Lưu Key.dat chứa khóa k1, k2, l1, l2, n1, n2

b, Thuật toán giải mã

Đầu vào:

- File văn bản C đã được mã hóa

- File Key.dat chứa hai khóa k1, k2, l1, l2, n1, n2

Đầu ra:

- File văn bản P theo định dạng txt

Các bước thực hiện:

Bước 1:

- Chọn file văn bản cần giải mã

- Chọn file Key.datchứa khóa k1, k2, l1, l2, n1, n2

Bước 2: Giải mã

- *Giải mã lần 1:*

+ Lập bảng theo cột theo thứ tự của khóa k1

+ Lấy chuỗi theo thứ tự của hàng n1

- *Giải mã lần 2:*

+ Lập bảng từ chuỗi đã giải ở trên theo thứ tự của khóa k2

+ Lấy chuỗi theo thứ tự của hàng n2

Bước 3:

- Lưu file đã được giải mã ở lần 2 với phần mở rộng txt

3.4.2 Giới thiệu thuật toán

- Giao diện chính: gồm có form sử dụng ToolStripMenu

Các sự kiện phương thức trong From chính:

```
private void encryptionToolStripMenuItem_Click(object sender, EventArgs e)
    {
        frmEncryption frmEn = new frmEncryption();
        frmEn.MdiParent = this;
        frmEn.WindowState = FormWindowState.Maximized;
        frmEn.Show();
    }
    private void decryptionToolStripMenuItem_Click(object sender, EventArgs e)
    {
        frmDeCryption frmDe = new frmDeCryption();
        frmDe.MdiParent = this;
```

```
            frmDe.WindowState = FormWindowState.Maximized;
            frmDe.Show();
        }
```

Lớp Key chứa thuộc tích để sử dụng khi ghi file Key.dat:

```
[Serializable()]
    class Key
    {
        private int n1, n2, l1, l2;
        private List<int> k1, k2;
        public int L1
        {
            get { return l1; }
            set { l1 = value; }
        }
        public int N2
        {
            get { return n2; }
            set { n2 = value; }
        }
        public int N1
        {
            get { return n1; }
            set { n1 = value; }
        }
        public int L2
        {
            get { return l2; }
```

```
        set { l2 = value; }
    }
     public List<int> K2
    {
        get { return k2; }
        set { k2 = value; }
    }
    public List<int> K1
    {
        get { return k1; }
        set { k1 = value; }
    }
    public Key(int l1, int l2)
    {
        this.L1 = l1;
        this.L2 = l2;

    }
    public Key()
    {
     }
  }
```

- Lớp Utlities chứa các phương thức đọc ghi key, tạo key, mã hóa, giải mã

```
class Utlities
  {
    private const string KEY_FILE = "KEY.DAT";
    public static Key objKey = new Key();
```

```
//Phương thức để ghi key
 public static bool SaveKey(Key objKey, string sPath)
 {
    string sFileName;
    FileStream fs = null;
    try
    {
       sFileName = sPath + "\\" + KEY_FILE;
       fs = new FileStream(sFileName, FileMode.OpenOrCreate);
       BinaryFormatter bf = new BinaryFormatter();
       bf.Serialize(fs, objKey);
       fs.Close();
    }
    catch
    {

       fs.Close();
       return false;
    }
    return true;
 }
 //Phương thức đểđọc key
 public static bool loadKey(string sPath)
 {
    string sFileName;
    FileStream fs = null;
    try
```

```
    {
        sFileName = sPath + "\\" + KEY_FILE;
        fs = new FileStream(sFileName, FileMode.OpenOrCreate);
        BinaryFormatter bf = new BinaryFormatter();
        objKey = (Key)bf.Deserialize(fs);
        fs.Close();
    }
    catch
    {
        fs.Close();
        return false;
    }
    return true;
 }
 //Tạo key random
public static List<int> GenneraKey(int l)
 {
    //MessageBox.Show("call1");
    List<int> k = new List<int>(l);
    Random rd = new Random();
    for (int i = 1; i <= l; i++)
    {
        int a = rd.Next(l);
        while (k.IndexOf(a) >= 0)
        {
            a = rd.Next(l);
        }
```

```
            k.Add(a);
        }
        return k;
    }
    //Chuyển key thành chuỗi
public static string StringKey(List<int> k)
{
    string strkey = "";
    foreach (int i in k)
    {
        strkey = strkey + i.ToString() + "; ";
    }
    return strkey;
}
public static string encryption(string P, List<int> key, int n)
{
    string str = "";
    int l = key.Count;
    char[] PChar = new char[P.Length];
    PChar = P.ToCharArray();
    char[,] strArr = new char[n, l];//khai báo bảng
    //Lập bảng
    int count = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < l; j++)
        {
```

```
            if (count < P.Length)
            {
                //MessageBox.Show("i="+i.ToString()  +  ";J="+j.ToString()+
PChar[count].ToString());
                strArr[i, j] = PChar[count];
            }
            count += 1;
        }
    }
    //Lấy chuỗi theo khóa k
    for (int i = 0; i < l; i++)
    {
        int j = key.IndexOf(i);
        //MessageBox.Show(j.ToString());
        for (int x = 0; x < n; x++)
        {
            //MessageBox.Show("x=" + x.ToString() + ";J=" + j.ToString() +
strArr[x, j].ToString());
            str += strArr[x, j];
        }
    }
            return str;
}
public static string Dencryption(string P, List<int> key, int n)
{
    string str = "";
    int l = key.Count;
```

```
char[] PChar = new char[P.Length];
PChar = P.ToCharArray();
char[,] strArr = new char[n, l];//khai báo bảng
//Lập bảng theo cột là khóa k
int count = 0;
for (int i = 0; i < l; i++)
{
   int j = key.IndexOf(i);
   //MessageBox.Show(j.ToString());
   for (int x = 0; x < n; x++)
   {
      if (count < P.Length)
      {
         // MessageBox.Show("i="+i.ToString() + ";J="+j.ToString()+
PChar[count].ToString());
         strArr[x, j] = PChar[count];
      }
      count += 1;
   }
}
 //Lấy chuỗi theo hàng
MessageBox.Show(str);
for (int i = 0; i < n; i++)
{
   for (int j = 0; j < l; j++)
   {
      str += strArr[i, j];
```

```
            }
        }
        return str;
    }
}
```

- Giao diện phần mã hóa: gồm có form và các đối tượng Button, Panel, Textbox, OpentFileDailog, SaveFileDailog, RichTextBox, Lable.

Các phương thức trong Form:

Sự kiện khi nhấn vào nút để mở file cần mã hóa:

```
private void btnOpent_Click(object sender, EventArgs e)
    {
        if (opfile.ShowDialog() == DialogResult.OK)
        {
            try {
                    FileStream fs = new FileStream(opfile.FileName, FileMode.Open);
                StreamReader str = new StreamReader(fs);
```

```
            rtfFile.Text = str.ReadToEnd();
            l = rtfFile.Text.Length;
            rtfFile1.Text = "";
        }
        catch(IOException io)
        {
            MessageBox.Show(io.Message);
        }
                }
    }
```

Sự kiện tạo key 1:

```
private void btnK1_Click(object sender, EventArgs e)
    {
        try
        {
            //MessageBox.Show("Click me");

            l1 = Convert.ToInt32(txtL1.Text.Trim());
            k1 = new List<int>(l1);
            k1= Utlities.GenneraKey(l1);
            lblK1.Text = "Key1 = " + Utlities.StringKey(k1);

            if (rtfFile.Text.Length > 0)
            {
                if (rtfFile.Text.Length % l1 == 0)
                {
                    n1 = rtfFile.Text.Length / l1;
```

```
            }
            else
            {
                n1 = rtfFile.Text.Length / l1 + 1;
            }

            //MessageBox.Show(n1.ToString());
            lblN1.Text = "n1 = " + n1.ToString();
        }
      }
      catch(Exception ex)
      {
        MessageBox.Show(ex.Message);
      }
    }
```

Sự kiện tạo key2:

```
private void btnK2_Click(object sender, EventArgs e)
    {
      try
      {
        l2 = Convert.ToInt32(txtL2.Text.Trim());
        k2 = new List<int>(l2);
        k2 = Utlities.GenneraKey(l2);
        lblK2.Text = "Key2 = " +  Utlities.StringKey(k2);
        if (rtfFile.Text.Length > 0)
        {
          if (rtfFile.Text.Length % l2 == 0)
```

```
            {
                n2 = rtfFile.Text.Length / l2;
            }
            else
            {
                n2 = rtfFile.Text.Length / l2 + 1;
            }
            lblN2.Text = "n2 = " + n2.ToString();
           // MessageBox.Show(n2.ToString());
         }

      }
      catch (Exception ex)
      {
         MessageBox.Show(ex.Message);
      }
   }
```

Sự kiện mã hóa lần 1:

```
private void btnEncryption1_Click(object sender, EventArgs e)
   {
     rtfFile1.Text = Utlities.encryption(rtfFile.Text, k1, n1);
   }
```

Sự kiện mã hóa lần 2:

```
private void btnEncryption2_Click(object sender, EventArgs e)
   {
      rtfFile.Text = rtfFile1.Text;
      rtfFile1.Text = Utlities.encryption(rtfFile.Text, k2, n2);
```

```
    }
```

Sự kiện ghi file đã được mã hóa với key:

```
private void btnSave_Click(object sender, EventArgs e)
    {
        if (savefile.ShowDialog() == DialogResult.OK)
        {
            Key objKey = new Key(l1, l2);
            objKey.K1 = k1;
            objKey.K2 = k2;
            objKey.N1 = n1;
            objKey.N2 = n2;
            try
            {
                FileStream fs = new FileStream(savefile.FileName + ".txt",
FileMode.OpenOrCreate);
                StreamWriter sw = new StreamWriter(fs, Encoding.Unicode);
                foreach (string line in rtfFile1.Lines)
                    sw.WriteLine(line);
                sw.Close();
                string local = savefile.FileName;
                local = local.Remove(local.LastIndexOf("\\"));
                Utlities.SaveKey(objKey, local);
                //MessageBox.Show(savefile.FileName);
            }
            catch (IOException io)
            {
                MessageBox.Show(io.Message);
```

```
            }

        }
    }
```

- Giao diện phần giải mã: gồm có form và các đối tượng Button, Panel, Textbox, OpentFileDailog, SaveFileDailog,RichTextBox, Lable.

Các phương thức trong Form:

Đọc file cần giải mã:

```
private void btnOpent_Click(object sender, EventArgs e)
    {
        if (opfile.ShowDialog() == DialogResult.OK)
        {
            try {

                FileStream fs = new FileStream(opfile.FileName, FileMode.Open);
                StreamReader str = new StreamReader(fs);
```

```
            rtfFile.Text = str.ReadToEnd();
            l = rtfFile.Text.Length;
        }
        catch(IOException io)
        {
            MessageBox.Show(io.Message);
        }
                }
    }
```

Đọc file chứa key:

```
private void btnLoadKey_Click(object sender, EventArgs e)
    {
        if (opfile.ShowDialog() == DialogResult.OK)
        {
            try
            {
                if (Utlities.loadKey(opfile.FileName))
                    objKey = Utlities.objKey;
                txtL1.Text = objKey.L1.ToString();
                lblK1.Text = "k1: " + Utlities.StringKey(objKey.K1);
                lblK2.Text = "k2: " + Utlities.StringKey(objKey.K2);
                lblN1.Text = objKey.N1.ToString();
                lblN2.Text = objKey.N2.ToString();
                        }
            catch (IOException io)
            {
                MessageBox.Show(io.Message);
```

```
            }

        }
    }
```

Giải mã lần 1:

```
private void btnDecryption1_Click(object sender, EventArgs e)
    {
        rtfFile1.Text = Utlities.Dencryption(rtfFile.Text, objKey.K2, objKey.N2);
    }
```

Giải mã lần 2:

```
private void btnDecryption2_Click(object sender, EventArgs e)
    {
        rtfFile.Text = rtfFile1.Text;
        rtfFile1.Text = Utlities.Dencryption(rtfFile.Text, objKey.K1, objKey.N1);
    }
```

Ghi File đã được giải mã:

```
private void btnSave_Click(object sender, EventArgs e)
    {
        if (savefile.ShowDialog() == DialogResult.OK)
        {
                    try
            {
                FileStream fs = new FileStream(savefile.FileName + ".txt", FileMode.OpenOrCreate);
                StreamWriter sw = new StreamWriter(fs, Encoding.Unicode);
```

```
            foreach (string line in rtfFile1.Lines)
                sw.WriteLine(line);
            sw.Close();

            //MessageBox.Show(savefile.FileName);
        }
        catch (IOException io)
        {
            MessageBox.Show(io.Message);
        }

    }
}
```

KẾT LUẬN

1. Phần làm đã làm được: Sau quá trình nghiên cứu, thực hiện luận văn, em đã tìm hiểu và nắm được một số vấn đề như:

- Tìm hiểu về một số loại mã hóa cổ điển cũng như một số loại mã hóa cao cấp được sử dụng hiện nay.
- Tìm hiểu về .NET Framework cũng như .NET Framework trong bảo mật thông tin.
- Xây dựng được chương trình demo về mã hóa file với thuật toán mã hóa và giải mã cải tiến mã cổ điển.

2. Phần chưa làm được: Bên cạnh những phần đã thực hiện được, luận văn vẫn còn tồn tại một số hạn chế như:

- Chưa đánh giá được thời gian mã hóa và giải mã.
- Chương trình ứng dụng còn đơn giản, giao diện chưa thân thiện.
- Chưa giải quyết được vấn đề trao đổi khóa mã k_1 và khóa k_2.
- Chưa so sánh, đánh giá được mức độ an toàn của thuật toán tốt hơn so với các thuật toán cổ điển khác.

3. Hướng phát triển: Tôi sẽ tập trung giải quyết vấn đề trao đổi các khóa mã và hoàn thiện các vấn đề còn tồn tại như: độ dài các khóa mã k_1 và k_2 bao nhiêu là đủ mức độ an toàn nhằm đưa những kết quả nghiên cứu của Luận văn vào ứng dụng thử trong thực tế. Vì thời gian hạn chế và mức độ nghiên cứu chưa sâu rộng. Em rất mong luận văn sẽ là tài liệu tham khảo cho các bạn sinh viên khóa sau khắc phục một số khuyết điểm để chương trình được hoàn thiện hơn như:

- Xây dựng ứng dụng có giao diện thân thiện dễ sử dụng.
- Bắt lỗi chi tiết từng lỗi và nêu ra lỗi ở đâu.
- Xây dựng hệ thống trợ giúp.

TÀI LIỆU THAM KHẢO

[1.] Nhập môn phân tích thông tin có bảo mật. Nhà xuất bản Thông tin và Truyền thông - TS. Hồ Văn Canh, TS. Nguyễn Viết Thế - 2010.

[2.] Giáo trình Mật mã học và an toàn thông tin - Nhà xuất bản Thông tin và Truyền thông - TS Thái Thanh Tùng – 2011.

[3.] Lý thuyết mật mã & An toàn thông tin. Nhà xuất bản Đại học Quốc Gia Hà Nội – Phan Đình Diệu – 2002.

[4.] Ngô Phương Nam, (Luận văn Thạc sĩ- 2012) " Nghiên cứu các phương pháp Thám mã một số luật mã thuộc Hệ mật mã Cổ điển trên văn bản tiếng Việt". Đà Nẵng, 3/ 2012.

[5.] Alfred J. Menezes, Paul C. Van Oorschot, Scott A. Vanstone, "Handbook of Applied Cryptography", CRC Press : Boca Raton, NewYork, London, and Tokyo, 2000.

[6.] Cryptography and Network Security Principles and Practices, 4 th Edition – William Stallings – Prentice Hall – 2005.

[7.] Caxton c. foster:“Cryptanalysis for Microcomputers” hayden book company, INC, New Jersey 1996.

[8.] D. Bonch, “Twenty Years of Attacks on the RSA Cryptosystem”, Notices of the AMS, Febryary 1994.

[9.] H. Delfs and H. KNEBL: “Introduction to Cryptography Principles and Applications”, Springer – Verlag, 2012.

[10.] Mark Stamp, Richard M. Low: “Applied Cryptanalysis: Breaking Ciphers in the Real World”, San Jose State University, 2016.

Hải Phòng, ngày 10 tháng 11 năm 2017