LỜI CẢM ƠN

Em xin được bày tỏ lòng biết ơn sâu sắc tới PGS.TS.Trịnh Nhật Tiến, người đã trực tiếp hướng dẫn, tận tình chỉ bảo em trong suốt quá trình làm khóa luận.

Em xin chân thành cảm ơn tất cả các thầy cô giáo trong khoa Công nghệ thông tin - Trường ĐHDL Hải Phòng, những người đã nhiệt tình giảng dạy và truyền đạt những kiến thức cần thiết trong suốt thời gian em học tập tại trường, để em hoàn thành tốt khóa luận.

Cuối cùng em xin cảm ơn tất cả các bạn đã góp ý, trao đổi hỗ trợ cho em trong suốt thời gian vừa qua.

Em xin chân thành cảm ơn!

Hải Phòng, ngày ... tháng 07 năm 2009

Sinh viên

Vũ Thị Ngân

.

MỤC LỤC

LỜI CẢM ƠN……1

MỤC LỤC……..2

GIỚI THIỆU ĐỀ TÀI……...5

***Chương 1*: MỘT SỐ KHÁI NIỆM CƠ BẢN……..6**

1.1. CÁC KHÁI NIỆM TOÁN HỌC……6

1.1.1. Một số khái niệm trong số học……...6

1.1.1.1. Khái niệm số nguyên tố……...6

1.1.1.2. Định lý về số nguyên tố……...6

1.1.1.3. Khái niệm số nguyên tố cùng nhau……....7

1.1.1.4. Khái niệm đồng dư……..7

1.1.2. Một số khái niệm trong đại số……8

1.1.2.1. Khái niệm Nhóm……8

1.1.2.2. Khái niệm Nhóm con của nhóm (G, *)……..9

1.1.2.3. Khái niệm Nhóm Cyclic…….....9

1.1.2.4. Khái niệm Tập thặng dư thu gọn theo modulo…….....9

1.1.2.5. Phần tử nghịch đảo……...10

1.1.2.6. Cấp của một phần tử……10

1.1.2.7. Phần tử nguyên thủy……11

1.1.3. Khái niệm Độ phức tạp của thuật toán……...12

1.1.3.1. Khái niệm bài toán……....12

1.1.3.2. Khái niệm Thuật toán……...12

1.1.3.3. Khái niệm Độ phức tạp của thuật toán……....13

1.1.3.4. Khái niệm "dẫn về được"……14

1.1.3.5. Khái niệm "khó tương đương"……....14

1.1.3.6. Khái niệm lớp bài toán P, NP……..14

1.1.3.7. Khái niệm lớp bài toán NP – Hard……...15

1.1.3.8. Khái niệm lớp bài toán NP – Complete……....15

1.1.3.9. Khái niệm hàm một phía và hàm cửa sập một phía……15

1.2. VẤN ĐỀ MÃ HÓA……16
1.2.1. Giới thiệu về mã hóa……..16
1.2.1.1. Khái niệm mật mã……16
1.2.1.2.Khái niệm mã hóa (Encryption)……...17
1.2.1.3. Khái niệm hệ mã hóa……....17
1.2.1.4. Những tính năng của hệ mã hóa……..18
1.2.2. Các phương pháp mã hóa……19
1.2.2.1. Hệ mã hóa khóa đối xứng……....19
1.2.2.2. Hệ mã hóa khóa phi đối xứng (hệ mã hóa khóa công khai)……...21
1.3. Một số bài toán trong mật mã……...23
1.3.1. Bài toán kiểm tra số nguyên tố lớn……23
1.3.2. Bài toán phân tích thành thừa số nguyên tố……..27
1.3.3. Bài toán tính logarit rời rạc theo modulo……...30
1.4. VẤN ĐỀ AN TOÀN CỦA HỆ MÃ HÓA……32
1.4.1. Các phương pháp thám mã……..32
1.4.1.1.Thám mã chỉ biết bản mã……...33
1.4.1.2. Thám mã biết bản rõ……34
1.4.1.3. Thám mã với bản rõ được chọn……...35
1.4.1.4. Thám mã với bản mã được chọn. ……....37
1.4.2. Tính an toàn của một hệ mật mã……42
1.4.2.1. An toàn một chiều (One - Wayness)……....42
1.4.2.2. An toàn ngữ nghĩa (Semantic Security)……...43
1.4.2.3. Tính không phân biệt được (Indistinguishability : IND)……....45
1.4.2.4. An toàn ngữ nghĩa tương đương với IND……....47
1.4.2.5. Khái niệm an toàn mạnh nhất IND-CCA……....48
***Chương 2*: TẤN CÔNG BẢN MÃ……50**
2.1. TẤN CÔNG HỆ MÃ HÓA RSA……...50
2.1.1. Hệ mã hóa RSA……..50
2.1.2. Các loại tấn công vào mã hóa RSA.....51
2.1.2.1. Tấn công loại 1: Tìm cách xác định khóa bí mật.....51

2.1.2.2. Tấn công dạng 2: Tìm cách xác định bản rõ.....53
2.2. TẤN CÔNG HỆ MÃ HÓA ELGAMAL.....55
2.2.1. Hệ mã hóa ELGAMAL.....55
2.2.2. Các dạng tấn công vào mã hóa ELGAMAL.....56
2.2.2.1. Tấn công dạng 1: Tìm cách xác định khóa bí mật.....56
2.2.2.2. Tấn công dạng 2: Tìm cách xác định bản rõ.....56
2.3. TẤN CÔNG HỆ MÃ HÓA: DỊCH CHUYỂN.....57
2.3.1. Mã dịch chuyển.....57
2.3.2. Dạng tấn công vào mã dịch chuyển: Tìm cách xác định khóa k.....57
2.4. TẤN CÔNG MÃ THAY THẾ.....58
2.4.1. Mã thay thế.....58
2.4.2. Dạng tấn công vào mã thay thế: Tìm cách xác định bản rõ.....58
2.5. TẤN CÔNG HỆ MÃ HÓA: AFFINE.....62
2.5.1. Mã Affine.....62
2.5.2. Dạng tấn công vào mã Affine: Tìm cách xác định khóa.....62
KẾT LUẬN.....65
BẢNG CHỮ CÁI VIẾT TẮT.....66
TÀI LIỆU THAM KHẢO.....67

GIỚI THIỆU ĐỀ TÀI

Khoa học mật mã từ khi ra đời tới nay đã trải qua nhiều giai đoạn phát triển, từ một môn khoa học thực nghiệm đã nhanh chóng trở thành môn khoa học logic đỉnh cao và ngày càng hội tụ những kiến thức tinh túy của loài người. Sự phát triển của khoa học mật mã đã góp phần thúc đẩy xã hội loài người ngày càng tiến lên. Đặc biệt trong thời đại ngày nay dưới tác động của cuộc cách mạng tin học hóa toàn cầu, khi các hoạt động kinh tế - xã hội trong mô hình kinh tế mở và biến động không ngừng, đặc biệt là với các dự án xây dựng chính phủ điện tử thì khoa học mật mã chiếm vị trí ngày càng quan trọng, và có những đóng góp không nhỏ trong việc bảo đảm an ninh cho các quốc gia, an toàn cho thông tin kinh tế - xã hội.

Như chúng ta đã biết, năm 1949 C.Shannon đã đưa ra mô hình hệ mật mã khóa đối xứng an toàn vô điều kiện dựa trên cơ sở lý thuyết thông tin. Trong thời đại ngày nay nhiều bài toán mật mã trong thực tế được đặt ra là " Chỉ cần giữ bí mật trong một thời gian nào đó cho một thông tin nào đó mà thôi".

Với mục đích giải quyết vấn đề trên, vào năm 1976 W.Diffie_M.E.Hellmam đã đề xuất mô hình hệ mật mã khóa phi đối xứng hay còn gọi là hệ mật mã khóa công khai, an toàn về mặt tính toán dựa trên cơ sở lý thuyết độ phức tạp tính toán.

Song song với việc chúng ta luôn tìm ra các giải pháp mã hóa tốt nhất để đảm bảo an toàn cho các thông tin được truyền đi, thì các kẻ thám mã cũng không ngừng nỗ lực tìm ra các sơ hở, các điểm yếu của những hệ mã hóa đó để phá được bản mã khi chúng "bắt" được một bản mã nào đó.

Với lý do trên em chọn đề tài: " Nghiên cứu một số loại tấn công bản mã", để biết được những điểm yếu cũng như những sơ hở của một số hệ mã hóa chúng ta sử dụng, mà theo đó kẻ thám mã có thể lợi dụng để "tấn công" vào các hệ mã hóa, biết được các thông tin bí mật. Từ đó giúp ta tìm cách phòng tránh, đưa ra các giải pháp tối ưu nhất, để đảm bảo an toàn cao nhất khi sử dụng các hệ mã hóa.

Chương 1: CÁC KHÁI NIỆM CƠ BẢN

1.1. CÁC KHÁI NIỆM TOÁN HỌC

1.1.1. Một số khái niệm trong số học

1.1.1.1. Khái niệm số nguyên tố

1□. Khái niệm

Số nguyên tố là số tự nhiên lớn hơn 1 và chỉ có hai ước là 1 và chính nó.

2□. Ví dụ:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37.

1.1.1.2. Định lý về số nguyên tố

1/. Định lý: về số nguyên dương > 1.

Một số nguyên dương n > 1 đều có thể biểu diễn được *duy nhất* dưới dạng:

$$n = P_1^{n_1}.P_2^{n_2}...P_k^{n_k}$$, trong đó:

k, n_i (i=1, 2,…, k) là các số tự nhiên, P_i là các số nguyên tố, từng đôi một khác nhau.

2./ Định lý: Mersenne.

Cho $p = 2^k - 1$, nếu p là số nguyên tố, thì k phải là số nguyên tố.

Chứng minh

Bằng phản chứng, giả sử k không là số nguyên tố. Khi đó k = a.b với

$1 < a, b < k$. Như vậy $p = 2^k - 1 = 2^{ab} - 1 = (2^a)^b - 1 = (2^a - 1).E$

(Trong đó E là một biểu thức nguyên – áp dụng công thức nhị thức Niu-tơn).

Điều này mâu thuẫn giả thiết p là số nguyên tố. Vậy giả sử sai, hay k là số nguyên tố

3/. Hàm Euler

Cho số nguyên dương n, *số lượng* các số nguyên dương bé hơn n và *nguyên tố cùng nhau* với n được ký hiệu $\phi(n)$ và gọi là hàm *Euler*.

Nhận xét: Nếu p là số nguyên tố, thì $\phi(p) = p - 1$.

Ví dụ:

Tập các số nguyên không âm nhỏ hơn 7 là $Z_7 = \{0, 1, 2, 3, 4, 5, 6\}$.

Do **7** là *số nguyên tố*, nên tập các số nguyên dương nhỏ hơn 7 và *nguyên tố cùng nhau* với 7 là $Z_7^* = \{1, 2, 3, 4, 5, 6\}$. Khi đó $/Z/ = \phi(p) = p - 1 = 7 - 1 = 6$.

Định lý: Nếu n là tích của hai số nguyên tố n = p.q, thì $\phi(n) = \phi(p).\phi(q) = (p-1)(q-1)$.

1.1.1.3. khái niệm số nguyên tố cùng nhau

1/. Khái niệm

Hai số nguyên a và b được gọi là nguyên tố cùng nhau nếu gcd(a, b) = 1.

2/. Ví dụ:

gcd(1, 3) = 1, gcd(2, 7) = 1, gcd(3, 10) = 1, gcd(5, 13) = 1…

1.1.1.4. Khái niệm đồng dư

1/. Khái niệm

Cho n là một số nguyên dương. Nếu a và b là hai số nguyên, khi đó a được gọi là đồng dư với b theo modulo n, được viết $a \equiv b \pmod{n}$ nếu $n \mid (a - b)$, và n được gọi là modulo của đồng dư.

2/. Ví dụ:

$24 \equiv 9 \pmod{5}$, $17 \equiv 5 \pmod{3}$

3/. Tính chất:

(i) $a \equiv b \pmod{n}$, nếu và chỉ nếu a và b đều trả số dư như nhau khi đem chia chúng cho n.m ⏐ (a-b).

(ii) $a \equiv a \pmod{n}$ (tính phản xạ).

(iii) Nếu $a \equiv b \pmod{n}$ thì $b \equiv a \pmod{n}$.

(iv) Nếu $a \equiv b \pmod{n}$ và $b \equiv c \pmod{n}$ thì $a \equiv c \pmod{n}$.

(v) Nếu $a \equiv a_1 \pmod{n}$ và $b \equiv b_1 \pmod{n}$ thì $a + b \equiv (a_1 + b_1) \pmod{n}$ và $a.b \equiv a_1.b_1 \pmod{n}$.

1.1.2. Một số khái niệm trong đại số

1.1.2.1. Khái niệm Nhóm

1/. Khái niệm

Nhóm là một bội (G, *), trong đó G ≠ ∅, * là ***phép toán hai ngôi*** trên G thỏa mãn ba tính chất sau:

+ Phép toán có tính kết hợp: (x*y)*z = x*(y*z) với mọi x, y, z ∈ G.

+ Có phần tử ***trung lập*** e ∈ G: x*e = e*x = x với mọi x ∈ G.

+ Với mọi x ∈ G, có phần tử nghịch đảo x' ∈ G: x*x' = x'*x = e.

Cấp của nhóm G được hiểu là số phần tử của nhóm, ký hiệu là |G|.

Cấp của nhóm có thể là ∞ nếu G có vô hạn phần tử.

Nhóm Abel là nhóm (G, *), trong đó phép toán hai ngôi * có tính giao hoán.

Tính chất: Nếu a*b = a*c, thì b = c.

Nếu a*c = b*c, thì a = b.

2/. Ví dụ:

* Tập hợp các số nguyên Z cùng với phép cộng (+) thông thường là nhóm giao hoán, có phần tử đơn vị là số 0. Gọi là ***nhóm cộng*** các số nguyên.

* Tập Q^* các số hữu tỷ khác 0 (hay tập R^* các số thực khác 0), cùng với phép nhân (*) thông thường là nhóm giao hoán. Gọi là ***nhóm nhân*** các số hữu tỷ (số thực).

* Tập các vectơ trong không gian với phép toán cộng vectơ là nhóm giao hoán.

*1.1.2.2. Khái niệm Nhóm con của nhóm (G, *)*

Nhóm con của G là tập S ⊂ G, S ≠ ϕ, và thỏa mãn các tính chất sau:

+ Phần tử trung lập e của G nằm trong S.

+ S khép kín đối với phép tính (*) trong G, tức là x*y ∈ S với mọi x, y ∈ S.

+ S khép kín đối với phép lấy nghịch đảo trong G, tức x^{-1} ∈ S với mọi x ∈ S.

1.1.2.3. Khái niệm Nhóm Cyclic

1/. Khái niệm

Nhóm (**G**, *) được gọi là ***Nhóm Cyclic*** nếu nó được sinh ra bởi một trong các phần tử của nó.

Tức là có phần tử **g** ∈ **G** mà với mỗi **a** ∈ **G**, đều tồn tại **n** ∈ **N** để g^n **=g*g*...*g = a.** (Chú ý: **g*g*...*g** là **g*g** với **n** lần).

Nói cách khác: **G** được gọi là Nhóm Cyclic nếu tồn tại **g** ∈ **G** sao cho mọi phần tử trong **G** đều là một ***lũy thừa nguyên*** nào đó của g.

2/. Ví dụ:

Nhóm ($\mathbf{Z}^+$, +) gồm các số nguyên dương là Cyclic với phần tử sinh **g** = 1.

1.1.2.4. Khái niệm Tập thặng dư thu gọn theo modulo

1/. Khái niệm

Kí hiệu $\mathbf{Z}_n$ = {0, 1, 2, ..., n-1} là tập các số nguyên không âm < n.

$\mathbf{Z}_n$ và phép cộng (+) lập thành ***nhóm Cyclic*** có phần tử sinh là **1**, pt trung lập **e** = 0.

($\mathbf{Z}_n$, +) gọi là nhóm cộng, đó là nhóm hữu hạn có cấp n.

Kí hiệu $\mathbf{Z}_n^*$ = {**x** ∈ $\mathbf{Z}_n$, **x** là nguyên tố cùng nhau với **n**}. Tức là **x** phải ≠ 0.

$\mathbf{Z}_n^*$ được gọi là ***Tập thặng dư thu gọn theo mod n***, có số phần tử là **ϕ(n).**

$\mathbf{Z}_n^*$ với phép nhân mod n lập thành một nhóm (nhóm nhân), pt trung lạp e = 1.

Tổng quát ($\mathbf{Z}_n^*$, phép nhân mod n) không phải là nhóm Cyclic.

Nhóm nhân $\mathbf{Z}_n^*$ là Cyclic chỉ khi **n** có dạng: 2, 4, p^k hay $2p^k$ với p là nguyên tố lẻ.

2/. Ví dụ:

Cho n = 21, Z_n^* = {1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20}.

1.1.2.5. Phần tử nghịch đảo

1/. Khái niệm

Cho a $\in$ Z_n, nếu tồn tại b $\in$ Z_n sao cho a.b $\equiv$ 1 (mod n), ta nói b là *phần tử nghịch đảo* của a trong Z_n và ký hiệu a^{-1}.

Một phần tử có phần tử nghịch đảo, gọi là khả nghịch.

2/. Ví dụ: Xét trong tập Z_7

Phần tử khả nghịch	1	2	3	4	5
Phần tử nghịch đảo	1	4	5	2	3

3/. Định lý:

UCLN (a, n) = 1 $\Leftrightarrow$ Phần tử a $\in$ Z_n có phần tử nghịch đảo.

4/. Hệ quả:

Mọi phần tử trong Z_n^* đều có phần tử nghịch đảo.

1.1.2.6. Cấp của một phần tử

1/. Định nghĩa

Cho a $\in$ Z_n^*, khi đó cấp của a, ký hiệu ord(a) là số nguyên dương t nhỏ nhất sao cho $a^t \equiv 1$ (mod n) trong Z_n^*.

2/. Ví dụ: Z_{21}^* = {1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20}.

a $\in Z_{21}^*$	1	2	4	5	8	10	11	13	16	17	19	20
Cấp của a	1	6	3	6	2	6	6	2	3	6	6	2

1.1.2.7. Phần tử nguyên thủy

1/. Khái niệm

Nếu n là một số nguyên tố, thì $\phi(n) = n - 1$, ta có với mọi $\alpha \in Z_n^*$

$$\alpha^{n-1} \equiv 1 \pmod{n}$$

Nếu α có cấp n – 1, tức n – 1 là số mũ bé nhất thỏa mãn công thức trên, thì các phần tử $\alpha, \alpha^2, \ldots, \alpha^{n-1}$ đều khác nhau và theo mod p, chúng lập thành Z_n^*. Khi đó ta nói Z_n^* là nhóm cyclic và α là phần tử sinh hay phần tử nguyên thủy của nhóm đó.

2/. Ví dụ:

Với số nguyên tố n = 2357, phần tử sinh của tập Z^*_{2357} là $\alpha = 2$.

3/. Tính chất:

(i) Với mọi số nguyên tố n, Z_n^* là nhóm cyclic, có $\phi(n - 1)$ phần tử nguyên thủy.

(ii) Nếu $n - 1 = n_1^{\alpha 1} . n_2^{\alpha 2} ... n_s^{\alpha s}$ là khai triển chính tắc của n – 1, và nếu:

$a^{n-1/n1} \equiv 1 \pmod{n}, \ldots, a^{n-1/ns} \equiv 1 \pmod{n}$, thì a là phần tử sinh của Z_n^* theo mod p.

(iii) Nếu g là phần tử nguyên thủy theo mod n, thì $\beta = g^i \bmod n$ với mọi i mà gcd(i, n - 1) = 1, cũng là phần tử sinh theo mod n.

1.1.3. Khái niệm Độ phức tạp của thuật toán

1.1.3.1. Khái niệm bài toán

Bài toán được diễn đạt bằng hai phần:

Input: Các dữ liệu vào của bài toán.

Ouput: Các dữ liệu ra của bài toán (kết quả).

Không mất tính chất tổng quát, giả thiết các dữ liệu đều là số nguyên dương.

1.1.3.2. Khái niệm Thuật toán

"Thuật toán" được hiểu đơn giản là cách thức để giải một bài toán. Cũng có thể được hiểu bằng hai quan niệm: Trực giác hay Hình thức như sau:

1/. Quan niệm trực giác về "Thuật toán"

Một cách trực giác, thuật toán được hiểu là một dãy hữu hạn các qui tắc (chỉ thị, mệnh lệnh) mô tả một quá trình tính toán, để từ dữ liệu đã cho (Input) ta nhận được kết quả (Output) của bài toán.

2/. Quan niệm toán học về "Thuật toán"

Một cách hình thức, người ta quan niệm thuật toán là một máy Turing.

Thuật toán được chia thành hai loại: Đơn định và không đơn định.

Thuật toán đơn định (Deterministic):

Là thuật toán mà kết quả của mọi phép toán đều được xác định duy nhất.

Thuật toán không đơn định (NoDeterministic):

Là thuật toán có ít nhất một phép toán mà kết quả của nó là không duy nhất.

1.1.3.3. Khái niệm Độ phức tạp của thuật toán

1/. Chi phí của thuật toán (Tính theo một bộ dữ liệu vào):

Chi phí phải trả cho một quá trình tính toán gồm chi phí về thời gian và bộ nhớ.

Chi phí thời gian của một quá trình tính toán là thời gian cần thiết để thực hiện một quá trình tính toán. Với thuật toán tựa Algol: Chi phí thời gian là số các phép tính cơ bản thực hiện trong quá trình tính toán.

Chi phí bộ nhớ của một quá trình tính toán là số ô nhớ cần thiết để thực hiện một quá trình tính toán.

Gọi A là một thuật toán, e là dữ liệu vào của bài toán đã được mã hóa bằng cách nào đó. Thuật toán A tính trên dữ liệu vào e phải trả một giá nhất định.

Ta ký hiệu:

$\mathbf{t}_A$(e) là giá thời gian và $\mathbf{l}_A$**(e)** là giá bộ nhớ.

2/. Độ phức tạp về bộ nhớ (trong trường hợp xấu nhất)**:**

$\mathbf{L}_A$**(n) = max{** $\mathbf{l}_A$**(e), với |e| ≤ n}, n** là "kích thước" đầu vào của thuật toán.

3/. Độ phức tạp thời gian (trong trường hợp xấu nhất)**:**

$\mathbf{T}_A$**(n) = max {** $\mathbf{t}_A$**(e), với |e| ≤ n}.**

4/. Độ phức tạp tiệm cận: Độ phức tạp PT(n) được gọi là tiệm cận tới hàm f(n), ký hiệu O(f(n)), nếu ∃ các số n_0, c mà PT(n) ≤ c.f(n), $\forall n \geq n_0$.

5/. Độ phức tạp đa thức:

Độ phức tạp PT(n) được gọi ***đa thức***, nếu nó ***tiệm cận tới đa thức p(n).***

6/. Thuật toán đa thức: Thuật toán được gọi là ***đa thức***, nếu độ phức tạp về thời gian (trong trường hợp xấu nhất) của nó là ***đa thức***.

Nói cách khác:

+ Thuật toán ***thời gian đa thức*** là thuật toán có độ phức tạp thời gian $\mathbf{O(n^t)}$, trong đó t là hằng số.

+ Thuật toán ***thời gian hàm mũ*** là thuật toán có độ phức tạp thời gian $\mathbf{O(t^{f(n)})}$, trong đó t là hằng số và f(n) là đa thức của n.

**** Thời gian chạy của các lớp thuật toán khác nhau:***

Độ phức tạp	Số phép tính ($n = 10^6$)	Thời gian (10^6 ptính/s)
O(1)	1	1 micro giây
O(n)	10^6	1 giây
$O(n^2)$	10^{12}	11,6 ngày
$O(n^3)$	10^{18}	32 000 năm
$O(2^n)$	10^{301030}	10^{301006} tuổi của vũ trụ

Chú ý:

- Có người cho rằng ngày nay máy tính với tốc độ rất lớn, không cần quan tâm nhiều tới thuật toán nhanh, chúng tôi xin dẫn một ví dụ đã được kiểm chứng.
- Bài toán xử lý n đối tượng, có ba thuật toán với 3 mức phức tạp khác nhau sẽ chịu 3 hậu quả như sau: ***Sau 1 giờ:***

Thuật toán A có độ phức tạp O(n) : xử lý được 3,6 triệu đối tượng.

Thuật toán B có độ phức tạp O(n log n) : xử lý được 0,2 triệu đối tượng.

Thuật toán C có độ phức tạp $O(2^n)$: xử lý được 21 đối tượng.

1.1.3.4. Khái niệm "dẫn về được"

Bài toán được gọi là ***"Dẫn về được"*** bài toán A một cách ***đa thức*** , ký hiệu: **B** $\propto$ **A**, nếu có thuật toán đơn định đa thức để giải bài toán **A**, thì cũng có thuật toán đơn định để giải bài toán **B**.

Nghĩa là: Bài toán A "khó hơn" bài toán **B**, hay **B** "dễ" hơn **A, B** được diễn đạt bằng ngôn ngữ của bài toán A, hay có thể hiểu B là trường hợp riêng của **A**.

Vậy nếu giải được bài toán A thì cũng sẽ giải được bài toán **B**.

Quan hệ $\propto$ có tính chất bắc cầu: Nếu C $\propto$ B và B $\propto$ A thì C $\propto$ A.

1.1.3.5. Khái niệm "khó tương đương"

Bài toán A gọi là "khó tương đương" bài toán B, ký hiệu **A ~ B,**

nếu: A $\propto$ B và B $\propto$ A.

1.1.3.6. Khái niệm lớp bài toán P, NP.

Ký hiệu:

P là lớp bài toán giải được bằng thuật toán đơn định, đa thức (Polynomial).

NP là lớp bài toán giải được bằng thuật toán không đơn định, đa thức.

Theo định nghĩa ta có p $\subset$ NP.

Hiện nay người ta chưa biết được **P** $\neq$ **NP ?**

1.1.3.7. Khái niệm lớp bài toán NP – Hard

Bài toán **A** được gọi là ***NP - Hard*** (NP - khó) nếu $\forall$**L** $\in$ **NP** đều là **L** $\propto$ **A.**

Lớp bài toán NP - Hard bao gồm tất cả những bài toán NP - Hard.

Bài toán NP - Hard có thể nằm ***trong*** hoặc ***ngoài*** lớp NP.

1.1.3.8. Khái niệm lớp bài toán NP – Complete

Bài toán A được gọi là NP - Complete (NP-đầy đủ) nếu A là ***NP - Hard*** và A $\in$ **NP.**

Bài toán NP - Complete là bài toán NP - Hard nằm trong lớp NP.

Lớp bài toán NP - Complete bao gồm tất cả những bài toán NP - Complete .

Lớp NP – Complete là có thực, vì Cook và Karp đã chỉ ra BT đầu tiên thuộc lớp này, đó là bài toán "thỏa được": SATISFYABILITY.

1.1.3.9. Khái niệm hàm một phía và hàm cửa sập một phía

1/. Hàm f(x) được gọi là ***hàm một phía*** nếu tính ***"xuôi"*** y = f(x) thì ***"dễ"***, nhưng tính ***"ngược"*** $x = f^{-1}(y)$ lại rất ***"khó".***

Ví dụ:

Hàm $f(x) = g^x \pmod p$, với p là số nguyên tố lớn, (g là phần tử nguyên thủy mod p) là hàm một phía.

2/. Hàm f(x) được gọi là ***hàm cửa sập một phía*** nếu tính y = f(x) thì ***"dễ"***,

tính $x = f^{-1}(y)$ lại rất ***"khó"*** . Tuy nhiên có cửa sổ sập **z** để tính $x = f^{-1}(y)$ là ***"dễ".***

Ví dụ:

$f(x) = x^a \pmod n$ (n là tích của hai số nguyên tố lớn, n = p*q) là hàm một phía. Nếu chỉ biết a và n thì tính $x = f^{-1}(y)$ rất ***"khó"*** , nhưng nếu biết ***cửa sập*** p và q, thì tính được $f^{-1}(y)$ là khá ***"dễ".***

1.2. VẤN ĐỀ MÃ HÓA

1.2.1. Giới thiệu về mã hóa

Mã hóa được sử dụng để bảo vệ tính bí mật của thông tin khi thông tin được truyền trên các kênh thông tin công cộng như các kênh bưu chính điện thoại, mạng internet v.v… Giả sử một người gửi A muốn gửi đến người nhận B một văn bản (chẳng hạn một bức thư) p, để bảo mật A lập cho p một bản mã c, và thay cho việc gửi p, A gửi cho B bản mã c, B nhận được c và "giải mã" c để lại được văn bản p như A định gửi. Để A biến p thành c và B biến ngược lại c thành p, A và B phải thỏa thuận trước với nhau các thuật toán lập mã và giải mã, và đặc biệt khóa mã hóa chung K để thực hiện các thuật toán đó.

Người ngoài, không biết các thông tin đó (đặc biệt không biết khóa K), cho dù có lấy trộm được c, cũng khó tìm được văn bản p mà hai người A và B muốn gửi cho nhau. Sau đây ta sẽ định nghĩa hình thức về sơ đồ mã hóa và cách thức thực hiện để lập mã và giải mã.

1.2.1.1. Khái niệm mật mã

"Mật mã" có lẽ là kỹ thuật được dùng lâu đời nhất trong việc bảo đảm ***"An toàn thông tin".*** Trước đây *"mật mã"* chỉ được dùng trong ngành an ninh quốc phòng, ngày nay việc đảm bảo ***"An toàn thông tin"*** là nhu cầu của mọi ngành, mọi người (do các thông tin chủ yếu được truyền trên mạng công khai), vì vậy kỹ thuật *"mật mã"* là công khai cho mọi người dùng. Điều bí mật nằm ở ***"khóa"*** mật mã.

Hiện nay có nhiều kỹ thuật mật mã khác nhau, mỗi kỹ thuật có ưu, nhược điểm riêng. Tùy theo yêu cầu của môi trường ứng dụng ta dùng kỹ thuật này hay kỹ thuật khác. Có môi trường cần phải an toàn tuyệt đối, bất kể thời gian và chi phí. Có môi trường lại cần giải pháp dung hòa giữa bảo mật và chi phí thực hiện.

Mật mã cổ điển chủ yếu dùng để "che giấu" dữ liệu. Với Mật mã hiện đại, ngoài khả năng "che giấu" dữ liệu, còn dùng để thực hiện: Ký số (ký điện tử), tạo đại diện thông điệp, giao thức bảo toàn dữ liệu, xác thực thực thể, xác thực tài liệu, giao thức chứng minh "không tiết lộ thông tin", giao thức thỏa thuận, giao thức phân phối khóa, chống chối cãi trong giao dịch điện tử, giao thức chia sẻ bí mật,…

Theo nghĩa hẹp, "mật mã" dùng để bảo mật dữ liệu, người ta quan niệm:
Mật mã học là khoa học nghiên cứu mật mã: Tạo mã và phân tích mã.
Phân tích mã là kỹ thuật, nghệ thuật phân tích mật mã, kiểm tra tính bảo mật của nó hoặc phá vỡ sự bí mật của nó. Phân tích mã còn gọi là thám mã.

Theo nghĩa rộng, "mật mã" là một trong những công cụ hiệu quả bảo đảm An toàn thông tin nói chung: bảo mật, bảo toàn, xác thực, chống chối cãi,…

1.2.1.2.Khái niệm mã hóa (Encryption)

1/. ***Mã hóa:*** là quá trình chuyển thông tin có thể đọc được (gọi là ***bản rõ***) thành thông tin **"khó"** thể đọc được theo cách thông thường (gọi là ***bản mã***).
Đó là một trong những kỹ thuật để bảo mật thông tin.
2/. ***Giải mã***: là quá trình chuyển thông tin ngược lại từ ***bản mã*** thành ***bản rõ***.
3/. ***Thuật toán mã hóa*** hay ***giải mã*** là thủ tục để thực hiện mã hóa hay giải mã.
4/. ***Khóa mã hóa*** là một giá trị làm cho thuật toán mã hóa thực hiện theo cách riêng biệt và sinh ra bản rõ riêng. Thông thường khóa càng lớn thì bản mã càng an toàn. Phạm vi các giá trị có thể có của khóa được gọi là ***Không gian khóa***.
5/. ***Hệ mã hóa*** là tập các thuật toán, các khóa nhằm che giấu thông tin, cũng như làm rõ nó.

1.2.1.3. Khái niệm hệ mã hóa

Một sơ đồ mã hóa là bộ năm

S = (P, C, K, E, D) thỏa mãn các điều kiện:

P: là một tập hữu hạn các ký tự bản rõ.
C: là một tập hữu hạn các ký tự bản mã.
K: là một tập hữu hạn các khóa.
E: là một ánh xạ từ K x P vào C, được gọi là phép lập mã.
D: là một ánh xạ từ K x C vào P, được gọi là phép giải mã.
Với $k \in K$ ta định nghĩa $e_k \in E$, $e_k: P \rightarrow C$; $d_k \in D$, $d_k: C \rightarrow P$; e_k, d_k được gọi là hàm lập mã và hàm giải mã tương ứng với khóa mật mã k. Các hàm đó phải thỏa mãn hệ thức: $d_k(e_k(x)) = x$ với $\forall x \in P$.

1.2.1.4. Những tính năng của hệ mã hóa

* Cung cấp một mức cao về tính toán bảo mật, toàn vẹn, chống chối bỏ và xác thực.

* Tính bảo mật: Bảo đảm bí mật cho các thông báo và dữ liệu bằng việc che giấu thông tin 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.

1.2.2. Các phương pháp mã hóa

Hiện nay có 2 loại mã hóa chính: mã hóa khóa đối xứng và mã hóa khóa công khai.

Hệ mã hóa khóa đối xứng có khóa lập mã và khóa giải mã "giống nhau", theo nghĩa biết được khóa này thì **"dễ"** tính được khóa kia. Vì vậy phải giữ bí mật cả 2 khóa.

Hệ mã hóa khóa công khai thì có khóa lập mã khác khóa giải mã **(ke ≠ kd)** biết được khóa nay cũng **"khó"** tính được khóa kia. Vì vậy chỉ cần bí mật khóa giải mã, còn công khai khóa lập mã.

1.2.2.1. Hệ mã hóa khóa đối xứng

1/. Khái niệm

Hệ mã hóa khóa đối xứng là hệ mã hóa mà biết được khóa lập mã thì có thể "dễ" tính được khóa giải mã và ngược lại. Đặc biệt một số hệ mã hóa có khóa lập mã và khóa giải mã trùng nhau (ke = kd), như hệ mã hóa "dịch chuyển" hay DES.

Hệ mã hóa khóa đối xứng còn gọi là ***Hệ mã hóa khóa bí mật***, hay ***khóa riêng***, vì phải giữ bí mật cả 2 khóa.

Trước khi dùng hệ mã hóa khóa đối xứng, người gửi và người nhận phải thỏa thuận thuật toán mã hóa (lập mã hay giải mã) và ***khóa chung*** (phải giữ bí mật).

Độ an toàn của Hệ mã hóa loại này ***phụ thuộc vào khóa***, nếu để lộ ra khóa này nghĩa là bất kỳ người nào cũng có thể mã hóa và giải mã thông báo trong hệ thống mã hóa.

Sự mã hóa và giải mã của hệ thống mã hóa khóa đối xứng biểu thị bởi:

$$E_k: P \rightarrow C \text{ và } D_k: C \rightarrow P$$

2/. Ví dụ:

+ *Hệ mã hóa cổ điển* là Mã hóa khóa đối xứng: dễ hiểu, dễ thực thi, nhưng có độ an toàn không cao. Vì giới hạn tính toán chỉ trong phạm vi bảng chữ cái, sử dụng trong bản tin cần mã, ví dụ Z_{26} nếu dùng các chữ cái tiếng anh. Với hệ mã hóa cổ điển, nếu biết khóa lập mã hay thuật toán lập mã, có thể "dễ" xác định được bản rõ, vì "dễ" tìm được khóa giải mã.

+ *Hệ mã hóa* DES (1973) là Mã hóa khóa đối xứng ***hiện đại***, có độ an toàn cao.

3/. Đặc điểm

**** Ưu điểm:***

Hệ mã hóa khóa đối xứng mã hóa và giải mã nhanh hơn hệ mã hóa khóa công khai.

**** Hạn chế:***

(i). Mã hóa khóa đối xứng chưa thật an toàn với lý do sau:

Người mã hóa và người giải mã có "chung" một khóa. Khóa phải được giữ bí mật tuyệt đối, vì biết khóa này "dễ" xác định được khóa kia và ngược lại.

(ii). Vấn đề thỏa thuận khóa và quản lý khóa chung là khó khăn và phức tạp. Người gủi và người nhận phải luôn thống nhất với nhau về khóa. Việc thay đổi khóa là rất khó và dễ bị lộ. Khóa chung phải được gửi cho nhau trên kênh an toàn.

Mặt khác khi hai người (lập mã, giải mã) cùng biết "chung" một bí mật, thì càng khó giữ được bí mật !

4/. Nơi sử dụng hệ mã hóa khóa đối xứng

Hệ mã hóa khóa đối xứng thường được sử dụng trong môi trường mà khóa chung có thể dễ dàng trao chuyển bí mật, chẳng hạn trong cùng một mạng nội bộ. Hệ mã hóa khóa đối xứng thường dùng để mã hóa những bản tin lớn, vì tốc độ mã hóa và giải mã nhanh hơn hệ mã hóa công khai.

1.2.2.2. Hệ mã hóa khóa phi đối xứng (hệ mã hóa khóa công khai)

1/. Khái niệm

Hệ mã hóa khóa phi đối xứng là Hệ mã hóa có khóa lập mã và khóa giải mã khác nhau (ke ≠ kd), biết được khóa này cũng **"khó"** tính được khóa kia.

Hệ mã hóa này còn được gọi là **Hệ mã hóa khóa công khai** vì:

+ ***Khóa lập mã*** cho ***công khai***, gọi là ***khóa công khai (Public key)***.

+ ***Khóa giải mã*** giữ bí mật, còn gọi là ***khóa riêng (Private key)*** hay ***khóa bí mật.***

Một người bất kỳ có thể dùng khóa công khai để mã hóa bản tin, nhưng chỉ người nào có đúng khóa giải mã thì mới có khả năng đọc được bản rõ.

Hệ mã hóa khóa công khai hay ***Hệ mã hóa phi đối xứng*** do Diffie và Hellman phát minh vào những năm 1970.

2/. Ví dụ:

Hệ mã hóa RSA, hệ mã hóa ELGAMAL,….

3/. Đặc điểm

**** Ưu điểm:***

(i). Thuật toán được viết một lần, công khai cho nhiều lần dùng, cho nhiều người dùng, họ chỉ cần giữ bí mật cho khóa riêng của mình.

(ii). Khi biết các tham số ban đầu của hệ mã hóa, việc tính ra cặp khóa công khai và bí mật phải là "dễ" , tức là trong thời gian đa thức.

Người gửi có bản rõ P và khóa công khai, thì "dễ" tạo ra bản mã C.

Người nhận có bản mã C và khóa bí mật, thì "dễ" giải được thành bản rõ P.

(iii). Người mã hóa dùng khóa công khai, người giải mã giữ khóa bí mật. Khả năng lộ khóa bí mật khó hơn vì chỉ có một người giữ gìn.

Nếu thám mã biết khóa công khai, cố gắng tìm khóa bí mật, thì chúng phải đương đầu với bài toán "khó".

(iv). Nếu thám mã biết khóa công khai và bản mã C, thì việc tìm ra bản rõ P cũng là bài toán "khó", số phép thử là vô cùng lớn, không khả thi.

**** Nhược điểm:***

Hệ mã hóa khóa công khai: mã hóa và giải mã ***chậm hơn*** hệ mã hóa khóa đối xứng.

4/. Nơi sử dụng hệ mã hóa khóa công khai

Hệ mã hóa khóa công khai thường được sử dụng chủ yếu trên các mạng công khai như Internet, khi mà việc trao đổi chuyển khóa bí mật tương đối khó khăn.

Đặc trưng nổi bật của hệ mã hóa công khai là khóa công khai (public key) và bản mã (ciphertext) đều có thể gửi đi trên một kênh truyền tin ***không an toàn.***
Biết cả khóa công khai và bản mã, thám mã cũng không dễ khám phá được bản rõ.

Nhưng vì có tốc độ mã hóa và giải mã ***chậm,*** nên hệ mã hóa khóa công khai chỉ dùng để mã hóa những bản tin ngắn, ví dụ như mã hóa khóa bí mật gửi đi.

Hệ mã hóa khóa công khai thường được sử dụng cho cặp người dùng thỏa thuận khóa bí mật của hệ mã hóa khóa riêng.

1.3. MỘT SỐ BÀI TOÁN TRONG MẬT MÃ

Trong phần này sẽ xét ba bài toán có vai trò quan trọng trong lý thuyết mật mã, đó là ba bài toán: Kiểm tra số nguyên tố, phân tích một số nguyên thành tích của các thừa số nguyên tố, tính logarit rời rạc của một số theo modulo nguyên tố. Ở đây ta mặc định rằng các số nguyên tố là rất lớn.

1.3.1. Bài toán kiểm tra số nguyên tố lớn

Cho n là số nguyên bất kỳ. Làm thế nào để biết n là số nguyên tố hay không? Bài toán được đặt ra từ những buổi đầu của số học, và trả qua hơn 2000 năm đến nay vẫn là một bài toán chưa có được những cách giải dễ dàng. Bằng những phương pháp đơn giản như phương pháp sàng Eurratosthène, từ rất sớm người ta đã xây dựng được các bảng số nguyên tố đầu tiên, rồi tiếp tục bằng nhiều phương pháp khác tìm thêm được nhiều số nguyên tố lớn.

Tuy nhiên chỉ đến giai đoạn hiện nay của lý thuyết mật mã hiện đại, nhu cầu sử dụng các nguyên tố và thử tính nguyên tố của các số mới trở thành một nhu cầu to lớn và phổ biến, đòi hỏi nhiều phương pháp mới có hiệu quả hơn.

Trong mục này sẽ lược qua vài tính chất của số nguyên tố và một vài phương pháp thử tính nguyên tố của một số nguyên bất kỳ.

1/. Tiêu chuẩn Euler-Solovay-Strassen:

a) Nếu n là số nguyên tố, thì với mọi số nguyên dương a ≤ n-1:

$$\left(\frac{a}{b}\right) \equiv a^{(n-1)/2} \bmod n .$$

b) Nếu n là hợp số, thì:

$$\left|\left\{a : 1 \le a \le n-1, \left(\frac{a}{b}\right) \equiv a^{(n-1)/2} \bmod n\right\}\right| \le \frac{n-1}{2}$$

2/. Tiêu chuẩn Solovay-Strassen-Lehmann:

a) Nếu n là số nguyên tố, thì với mọi số nguyên dương a ≤ n-1:

$$a^{(n-1)/2} \equiv \pm 1 \bmod n$$

b) Nếu n là hợp số thì

$$\left|\left\{a : 1 \le a \le n-1, a^{(n-1)/2} \equiv \pm 1 \bmod n\right\}\right| \le \frac{n-1}{2}$$

3/. Tiêu chuẩn Miler-Rabin:

a) Cho n là số nguyên lẻ, ta viết (n-1) = 2^e.u, với u là số lẻ. Nếu n là số nguyên tố, thì với mọi số nguyên dương a ≤ n-1:

$$(a^u \equiv a \bmod n) \vee \exists k < e(a^{2^k.u} \equiv -1 \bmod n).$$

b) Nếu n là hợp số, thì

$$\left|\{a : 1 \le a \le n-1, (a^u \equiv 1 \bmod n) \vee \exists k < e(a^{2^k.u} \equiv -1 \bmod n)\}\right| \le \frac{n-1}{2}$$

Các tiêu chuẩn kể trên là cơ sở để ta xây dựng các thuật toán xác suất kiểu Monte-Carlo thử tính nguyên tố (hay hợp số) của các số nguyên.

Thuật toán Euler-Solovay-Strassen.

Dữ liệu vào: số nguyên dương n và t số ngẫu nhiên $a_1, ..., a_t$

() $(1 \le a_i \le n-1)$,

1. **for i = 1 to t do**
2. **if** $(a_i / n) \equiv a_i^{(n-1)/2} \bmod n$, then
3. **answer** "n là số nguyên tố"
4. **else**
5. **answer** "n là hợp số" and **quit**

Nếu thuật toán cho trả lời "*n* là hợp số" thì đúng *n* là hợp số. Nếu thuật toán cho trả lời "*n* là số nguyên tố", thì trả lời đó có thể sai với xác suất Monte-Carlo thiên về có, nếu xem nó là thuật toán thử tính là hợp số. Thuật toán xác suất thiên về không, nếu nó là thuật toán thử tính nguyên tố của các số nguyên.

Tương tự, dựa vào các tiêu chuẩn 2 và 3, người ta đã xây dựng các thuật toán xác suất Solovay-Strassen-Lehmann và Miler-Rabin kiểu Monte-Carlo để thử tính nguyên tố (hay hợp số) của các số nguyên.

Hai thuật toán đó chỉ khác thuật toán Euler-Solovay-Strassen ở chỗ công thức trong hàng lệnh 2 cần được thay tương ứng bởi:

$$a^{(n-1)/2} \equiv \pm 1 \bmod n$$

Hay $\left|\{(a^u \equiv 1 \bmod n) \vee \exists k < e(a^{2^k.u} \equiv -1 \bmod n)\}\right|$

trong đó u và e được xác định bởi: (n-1) = 2^e.u, u là số lẻ.

Xác suất sai lầm ε khi nhận được kết quả "n là số nguyên tố" trong các thuật toán trên được tính như sau: Giả sử n là số lẻ trong khoảng N và 2N, tức N < n < 2N. Gọi A là sự kiện "n là số nguyên tố", và B là sự kiện "thuật toán cho kết quả trả lời n là số nguyên tố". Ta phải tính xác suất ε = p(A|B). Theo tính chất (b) của tiêu chuẩn Euler-Solovay-Strassen, nếu n là hợp số, thì sự kiện

$$\left(\frac{a}{b}\right) \equiv a^{(n-1)/2} \bmod n$$

đối với mỗi a ngẫu nhiên (1 ≤ a ≤ n-1) có xác suất ≤ 1/2, vì vậy ta có:

$$p(B/A) \le \frac{1}{2^t}.$$

Theo công thức Bayes ta có:

$$p(A/B) = \frac{p(B/A).p(A)}{p(B)} = \frac{p(A/B).p(A)}{p(B/A).p(A).p(B/\overline{A}).p(\overline{A})}$$

Theo định lý về số nguyên tố, số các số nguyên tố giữa N và 2N xấp xỉ
N/ lnN ≈ n/ ln n, số các số lẻ là N/2 ≈ n/2, do đó $p(\overline{A}) \approx 2/\ln n$ và $p(\overline{A}) \approx 1\text{-}2/\ln n$. Dĩ nhiên ta có $p(B/\overline{A}) = 1$. Thay các giá trị đó vào công thức trên, ta được:

$$p(A/B) \le \frac{2^{-t}(1-\frac{2}{\ln n})}{2^{-t}(1-\frac{2}{\ln n})+\frac{2}{\ln n}} = \frac{\ln n - 2}{\ln n - 2 + 2^{t+1}} \qquad (1.1)$$

Chú ý:

Đánh giá đó cũng đúng đối với thuật toán Solovay-Strassen-Lehmann. Đối với thuật toán Miler-Rabin, ta được một đánh giá tốt hơn, cụ thể là:

$$p(A/B) = \frac{\ln n - 2}{\ln n - 2 + 2^{t+1}} \qquad (1.2)$$

Chú ý rằng khi t = 50 thì đại lượng ở vế phải của (1.1) ≈ 10^{-13} , và vế phải của (1.2) ≈ 10^{-28}; do đó nếu chọn cho dữ liệu vào năm mươi số ngẫu nhiên a_i thì các thuật toán Euler-Solovay-Strassen và Solovay-Lehmann sẽ thử cho ta một số nguyên tố với xác suất sai lầm ≤ 10^{-13} và thuật toán Miler-Rabin với xác suất sai lầm là ≤ 10^{-28}.

Có thể tính được độ phức tạp tính toán về thời gian của các thuật toán xác suất kể trên vào cỡ $\log_n$, tức là đa thức theo độ dài biểu diễn của dữ liệu vào (số n). Tuy nhiên các thuật toán đó chỉ cho ta thử tính nguyên tố của một số với xác suất sai lầm ε nào đó, dù ε là rất bé. Trong nhiều ứng dụng ta muốn có được số nguyên tố với độ chắc chắn 100% là số nguyên tố. Khi đó ta có thể dùng các thuật toán xác suất như trên và sau đó tìm kiếm những thuật toán tất định để thử tính nguyên tố với độ chính xác tuyệt đối. Adleman, Pomerance và Rumely đã đề xuất một số thuật toán kiểu như vậy, trong đó nổi bật là thuật toán thử tổng Jacobi, sau đó được đơn giản hóa bởi Cohen và Lenstra. Gold Wasser, Kilian, Adleman và Hoang đề xuất thuật toán thử bằng đường cong Elliptic, và được tiếp tục hoàn thiện bởi Atkin và Morain. Các thuật toán này đã được dùng để tìm nhiều số nguyên tố lớn.

4/. Thuật toán Agrawal-Kayal-Saxene

Tháng 8-2002, các nhà toán học Ấn độ Agrawal, Kayal và Sexena đưa ra thuật toán tất định thử tính nguyên tố có độ phức tạp thời gian đa thức, khá đơn giản

Thuật toán Agrawal-Kayal-Saxena:

Input: integer n > 1

1. If (n is of the form a^b, b > 1) output COMPOSITE;
2. r = 2;
3. While (r < n) {
4. if (gcd(n, r) ≠ 1) output COMPOSITE;
5. if (r is prime)
6. let q be the largest prime factor of r -1 ;
7. if ($q \geq 4\sqrt{r}\log n$) and ($n^{\frac{r-1}{q}} \neq 1 (\mathrm{mod} r)$)
8. break;
9. r ← r + 1;
10. }
11. for a = 1 to $2\sqrt{r}\log n$
12. if $(x-a)^n \neq (x^n - a)(\mathrm{mod} x^r - 1, n)$ output COMPOSITE;
13. output PRIME;

Thuật toán này đã được một số nhà toán học kiểm nghiệm, đánh giá cao và xem là thuật toán tốt, có thể dùng cho việc kiểm thử tính nguyên tố của các số nguyên.

Trong thực tiễn xây dựng các giải pháp mật mã, có ngu cầu các số nguyên tố rất lớn. Để tìm được số như vậy, người ta chọn ngẫu nhiên một số n rất lớn, dùng một thuật toán xác suất, chẳng hạn như thuật toán Miller-Rabin. Nếu thuật toán cho kết quả "n là số nguyên tố" với một xác suất sai ε nào đó, thì dùng tiếp một thuật toán tất định (chẳng hạn thuật toán Thuật toán Agrawal-Kayal-Saxena) để đảm bảo chắc chắn 100% rằng số n là nguyên tố.

Thuật toán Agrawal-Kayal-Saxena được chứng tỏ là có độ phức tạp thời gian đa thức cỡ $O((\log n)^{12})$ khi thử trên số n. Nếu số nguyên tố được thử có dạng Sophie Gerrmain, tức dạng 2p+1, thì độ phức tapk thời gian sẽ chỉ cỡ $O((\log n)^{6})$.

1.3.2. Bài toán phân tích thành thừa số nguyên tố

Bài toán phân tích một số nguyên thành thừa số nguyên tố cũng được xem là bài toán khó, thường được sử dụng trong lý thuyết mật mã. Biết số n là hợp số thì việc phân tích n thành các thừa số, mới là có nghĩa; do đó để phân tích n thành các thừa số, ta thử trước n có phải là hợp số hay không.

Bài toán phân tích n thành các thừa số có thể dẫn về bài toán ***tìm một ước số của n***. Vì biết một ước số d của n, thì tiến trình phân tích n được tiếp tục thực hiện bằng cách phân tích d và n/d.

Bài toán phân tích thành các thừa số, hay bài toán tìm ước số của một số nguyên cho trước, đã được nghiên cứu nhiều, nhưng cũng chưa có thuật toán hiệu quả nào để giải nó trong trường hợp tổng quát. Do đó người ta có khuynh hướng tìm thuật toán giải nó trong những trường hợp đặc biệt, chẳng hạn khi n có một ước số nguyên tố p với p – 1 là ***B-min***, hoặc khi n kà số Blum, tức là số có dạng tích của hai số nguyên tố lớn nào đó n = p.q.

Một số nguyên n được gọi là ***B-min*** nếu tất cả các ước số nguyên tố của nó đều ≤ B với một cận B > 0 nào đó.

1/. Trường hợp 1

Giải sử n là ***B-min***. Ký hiệu Q là bội chung bé nhất của các lũy thừa của các số nguyên tố ≤ B mà bản thân chúng ≤ n. Nếu $q^l \le n$ thì l ln(q) ≤ ln n, tức l≤ $\left\lfloor \frac{\ln n}{\ln q} \right\rfloor$ (⌊x⌋ là số nguyên bé nhất lớn hơn x).

Ta có $Q = \prod_{q \le B} q^{\lfloor \ln n / \ln q \rfloor}$

trong đó tích lấy theo tất cả các số nguyên tố khác nhau q ≤ B.

Nếu p là thừa số nguyên tố của n sao cho p-1 là ***B-min***, thì p-1|Q, và do đó với mọi a bất kỳ thỏa mãn gcd(a, p) = 1. Theo định lý Fermat ta có $a^Q \equiv q \bmod p$

Vì vậy nếu lấy d = gcd(a^Q-1, n) thì p|d.

Nếu d = n thì coi như thuật toán không cho ta điều mong muốn, tuy nhiên điều đó chắc không xảy ra nếu n có ít nhất hai thừa số nguyên tố khác nhau.

(p-1)-Thuật toán Pollard phân tích thành thừa số:

INPUT: Một hợp số n không phải là lũy thừa của một số nguyên tố.

OUTPUT: Một thừa số không tầm thường của n.

1. Chọn một cận cho độ mịn B.
2. Chọn ngẫu nhiên một số nguyên a, 2 ≤ a ≤ n-1, và tính d = gcd(a, n).

   Nếu d ≥ thì cho ra kết quả (d).
3. Với mỗi số nguyên tố q ≤ B thực hiện:

   3.1. Tính $l = \left\lfloor \frac{\ln n}{\ln q} \right\rfloor$.

   3.2. Tính a ← $a^{q^l} \bmod n$
4. Tính d = gcd(a-1, n).
5. Nếu 1 < d < n thì cho kết quả (d).

   Nếu ngược lại thì coi như không có kết quả.

2/. Trường hợp 2

Xét trường hợp số nguyên Blume, tức là số có dạng n = p.q, tích của hai số nguyên tố lớn. Chú ý rằng nếu biết hai số nguyên khác nhau x và y sao cho $x^2 \equiv y^2 (\mathrm{mod} n)$ thì dễ tìm được một thừa số của n. Thực vậy, từ $x^2 \equiv y^2 (\mathrm{mod} n)$ ta có thể suy ra rằng $x^2 - y^2 = (x+y)(x-y)$ chia hết cho n, do n không là ước số của x + y hoặc x – y, nên gcd(x-y, n) phải là một ước số của n, tức bằng p hoặc q.

Ta biết nếu n = p.q là số Blume, thì phương trình đồng dư $x^2 \equiv a^2 (\mathrm{mod} n)$ có 4 nghiệm, hai nghiệm tầm thường là x = a và x = -a. Hai nghiệm không tầm thường khác là ±b, chúng là nghiệm của hai hệ phương trình đồng dư bậc nhất sau:

$$\left\{\begin{matrix} x = a(\mathrm{mod}\, p) \\ x = -a(\mathrm{mod} q) \end{matrix}\right\} \qquad \left\{\begin{matrix} x = -a(\mathrm{mod}\, p) \\ x = a(\mathrm{mod} q) \end{matrix}\right\}$$

Bằng lập luận như trên, ta thấy rằng n là số Blume, a là số nguyên tố với n, và ta biết một nghiệm không tầm thường của phương trình $x^2 \equiv a^2 (\mathrm{mod} n)$, tức là biết x ≠ ±a sao cho $x^2 \equiv a^2 (\mathrm{mod} n)$ thì gcd(x-a, n) sẽ là một ước số của n.

Từ những điều rút ra ở trên, người ta đã tìm ra một số phương pháp tìm ước số nguyên tố của một số nguyên dạng Blum. Các phương pháp đó dựa vào việc tìm một nghiệm không tầm thường của phương trình $x^2 \equiv 1 (\mathrm{mod} n)$. Ta giả thiết a.b = 2^S.r với r là số lẻ. Ta phát triển một thuật toán xác suất kiểu Las Vegas như sau: Chọn một số ngẫu nhiên v ($1 \le v \le n-1$). Nếu v may mắn là bội số của p hay q, thì ta được ngay một ước số của n là gcd(v, n). Nếu v nguyên tố với n, thì ta tính các bình phương liên tiếp kể từ v^r, được v^r, v^{2r}, v^{4r}, ... cho đến khi được $v^{2^t.r} \equiv 1 (\mathrm{mod} n)$ với một t nào đó. Số t như vậy bao giờ cũng đạt được, vì có $2^s.r \equiv 0 (\mathrm{mod} \phi(n))$ nên có $v^{2^t.r} \equiv 1 (\mathrm{mod} n)$. Như vậy ta đã tìm được số $x = v^{2^{t-1}.r}$ sao cho $x^2 \equiv 1 (\mathrm{mod} n)$. Tất nhiên có x ≠ 1 mod n. Nếu cũng có x ≠ -1mod n thì x là nghiệm không tầm thường của $x^2 \equiv 1 (\mathrm{mod} n)$, từ đó ta có thể tìm ước số của n. Nếu không thì thuật toán cho kết quả không đúng. Người ta có thể ước lượng xác suất cho kết quả không đúng với một lần thử với một số v là < ½, do đó nếu thiết kế thuật toán với m số ngẫu nhiên $v_1, v_2, ..., v_m$, thì sẽ đạt được xác suất kết quả không đúng là < $1/2^m$.

1.3.3. Bài toán tính logarit rời rạc theo modulo

Cho p là số nguyên tố và α là phần tử nguyên thủy theo mod p. Bài toán tính logarit rời rạc theo mod p là bài toán tìm, với mỗi số $\beta \in Z_p^*$, một số a ($1 \le a \le p-1$) sao cho $\beta = \alpha^a \bmod p$, tức là $a = \log_\alpha \beta (\bmod p-1)$.

Một thuật toán tầm thường để giải bài toán này là duyệt toàn bộ các số a từ q đến p-1, cho đến khi tìm được a thỏa mãn $\beta = \alpha^a \bmod p$.

Tuy nhiên thuật toán này sẽ không hiệu quả nếu p là số rất lớn. Một biến dạng của thuật toán đó với ít nhiều hiệu quả hơn là thuật toán Shanks.

1/. Thuật toán Shanks

Đặt $m = \lfloor \sqrt{p-1} \rfloor$. Ta tìm a dưới dạng a = mj + i, $0 \le i, j \le m-1$.

Rõ ràng $\beta = \alpha^a \bmod p$ khi và chỉ khi $\alpha^{mj} = \beta\alpha^i \bmod p$.

Ta lập hai danh sách gồm có các cặp (j, α^{mj}) và (i, $\beta\alpha^{-i}$) với i, j chạy từ 0 đến m-1. Khi phát hiện hai cặp từ hai danh sách đó có phần tử thứ hai bằng nhau là ta được kết quả a = mj + i, đó chính là giá trị $\log_\alpha \beta$ mà ta cần tìm. Thuật toán Shanks có độ phức tạp cỡ O(m) phép toán nhân và O(m) bộ nhớ (chứ kể $O(m^2)$) phép so sánh).

2/. Thuật toán Polig-Hellman

Được dùng có hiệu quả trong trường hợp p-1 chỉ có các thừa số nguyên tố bé. Giả thiết rằng p-1 có dạng phân tích chính tắc là:

$$p-1 = \prod_{i=1}^{k} p_i^{c_i}$$

Để tìm a = $\log_\alpha \beta$ (mod p-1), ta tìm các số a_i sao cho $a_i \equiv$ a mod $P_i^{c_i}$ với i = 1,...,k.

Sau khi tìm được các a_i, thì hệ phương trình $x \equiv a_i \bmod P_i^{c_i}$ (i = 1,..., k), được giải theo định lý số dư Trung quốc, sẽ cho lời giải x = a (mod p-1) cần tìm.

Vấn đề là xác định các số $a_i \bmod P_i^{c_i}$ (i = 1,..., k). Vấn đề này phát biểu như sau:

Giả sử q là một ước số nguyên tố của p-1, và q^c | p-1, nhưng không còn q^{c+1}|p-1. Ta cần tìm $x = \bmod q^c$.

Ta biểu diễn x dưới dạng sau:

$$X = \sum_{i=0}^{c-1} X_i q_i \qquad (0 \le x_i \le \text{q-1})$$

Vì x = mod q^c nên a viết dưới dạng a = x + q^c .s và vì $\alpha^{p-1} \equiv 1 (\text{mod}\, p)$, nên ta có

$\beta^{(p-1)/q} \equiv \alpha^{\alpha(p-1)/q} \equiv (\alpha^{p-1})^{a/q} \equiv \alpha^{(p-1)x_0/q} (\text{mod}\, p)$

Ta đặt $\gamma = \alpha^{(p-1)/q}$, và tính lần lượt $\gamma^0, \gamma^1, \gamma^2, ...,$ đồng thời so sánh với $\beta^{(p-1)/q} \bmod p$.

Ta lấy số i đó là x_0, tức x_0 = i.

Nếu c = 1 thì x = x_0, ta tìm xong x. Nếu c > 1 thì đặt $\beta^i = \beta\alpha^{-x}$

Tương tự như trên, tính lần lượt $\gamma^0, \gamma^1, \gamma^2, ...,$ đồng thời so sánh với $\beta^{(p-1)/q^2}$, ta tìm được x_1.

Cứ làm như vậy, ta tìm được dần các giá trị x_i với i = 0, 1, ..., c-1, tức tính được x.

Sau khi tìm được tất cả các giá trị của x ứng với mọi số nguyên tố q của p, thì theo một số nhận xét ở trên, chỉ cần giải tiếp một hệ phương trình đồng dư bậc nhất theo các modulo từng cặp nguyên tố với nhau (bằng phương pháp số dư Trung quốc), ta tìm được số a cần tìm, a = $\log_\alpha \beta$ theo mod p.

Thuật toán Polig-Hellman cho ta cách tính logarit rời rạc khá hiệu quả, nhưng chỉ khi p-1 chỉ có các thừa số nguyên tố bé. Nếu p-1 có ít nhất một thừa số nguyên tố lớn, thì thuật toán đó khó hiệu quả, trong trường hợp đó bài toán tính logarit rời rạc theo mod p vẫn là bài toán khó.

Một lớp các số nguyên tố p mà p-1 có ít nhất một thừa số nguyên tố lớn và lớp các số nguyên tố dạng p = 2q+1, trong đó q là số nguyên tố. Đó gọi là số nguyên tố dạng Sophie Germain, có vai trò quan trọng trong việc xây dựng các hệ mật mã khóa công khai.

Người ta đã nghiên cứu phát triển khá nhiều thuật toán khác, cả thuật toán tất định, cả thuật toán xác suất, để tính logarit rời rạc, nhưng chưa có thuật toán nào được chứng tỏ là có độ phức tạp thời gian đa thức.

1.4. VẤN ĐỀ AN TOÀN CỦA HỆ MÃ HÓA

1.4.1. Các phương pháp thám mã

Mật mã được sử dụng trước hết là để đảm bảo tính bí mật cho các thông tin được trao đổi, và do đó bài toán quan trọng nhất của thám mã cũng là bài toán phá bỏ tính bí mật đó, tức là từ bản mật mã có thể thu được dễ dàng (trên các kênh truyền tin công cộng), thám mã phải phát hiện được nội dung thông tin được che giấu trong bản mật mã đó, mà tốt nhất là tìm ra được bản rõ gốc của bản mật mã đó.

Ngày nay ta có thể phân loại bài toán thám mã thành các bài toán thám mã thành các bài toán cụ thể sau:

1/. Chỉ biết bản mã:

Khi thám mã chỉ biết một bản mật mã Y.

2/. Biết được cả bản rõ:

Khi thám mã biết một bản mật mã Y và bản rõ tương ứng X.

3/. Bản rõ được lựa chọn:

Thám mã có thể chọn một bản rõ X, và biết bản mật mã tương ứng Y. Điều này có thể xảy ra khi thám mã chiếm được (tạm thời) máy lập mã.

4/. Bản mã được lựa chọn:

Thám mã có thể chọn một bản mật mã Y, và biết bản rõ tương ứng X. Điều này có thể xảy ra khi thám mã chiếm được (tạm thời) máy giải mã.

1.4.1.1.Thám mã chỉ biết bản mã

Thám mã chỉ biết bản mã (Ciphertext only attack) (COA) là mô hình thám mã, trong đó giả sử rằng thám mã chỉ biết duy nhất tập các bản mã.

Xảy ra trường hợp này khi (như thời xưa) bắt được kẻ đưa thư, hoặc (thời ngày nay) chặn được thông tin truyền trên mạng.

Thám mã là thành công hoàn toàn nếu như các bản rõ tương ứng có thể được suy ra, hay tốt hơn là có thể tìm được khóa giải mã. Khả năng để tìm được bất kì thông tin gì về bản rõ cơ sở cũng được xem là thành công.

Những hệ mã hóa trước đây thực hiện bằng bút và giấy, thường bị phá bởi việc dùng bản mã đơn độc. Đối với hệ mã cổ điển đơn giản, nhà lập mã phát triển kỹ thuật thống kê cho việc tấn công bản mã như "phân tích tần số" (frequency analysis)

Các hệ mã hiện đại cố gắng cung cấp khả năng bảo vệ chống lại tấn công dạng này, bằng cách sử dụng giả thuyết giải bản mã tương ứng với việc giải bài toán "khó", quá trình kiểm tra khả năng của hệ mã mới thường kéo dài nhiều năm, bao gồm việc kiểm tra toàn bộ mọi khía cạnh của một số lượng lớn các bản mã từ bất kì một sự thống kên ngẫu nhiên nào.

Ví dụ như hệ mã RSA, để tìm ra bản rõ từ bản mã thì phải giải bài toán "khó" là bài toán RSA.

1.4.1.2. Thám mã biết bản rõ

Thám mã biết bản rõ (Known plaintext attack) (KPA) là một mô hình tấn công để giải mã, trong đó thám mã biết cả bản rõ và bản mã, và tự do dùng chúng để tìm kiếm những thông tin khác về hệ mã, đặc biệt là khóa bí mật.

Tuy nhiên bản mã và bản rõ là biết được một cách ngẫu nhiên ngoài ý muốn của thám mã, ví dụ thông tin được cắt thành nhiều bản rõ để mã hóa thành nhiều bản mã tương ứng, nhiều người đưa thư khác nhau mang đến một đích nào đó, thám mã vô tình bắt được kẻ đưa thư cầm một bản mã và một bản rõ tương ứng với nó, sự bắt được này là hoàn toàn vô tình, thám mã không thể lựa chọn trước được.

Nhiều hệ mã hóa cổ điển dễ bị tấn công đối với thám mã loại này. Ví dụ như đối với hệ mã dịch chuyển (shift cipher) (hệ mã Caesar), nếu biết một cặp bản mã và bản rõ bất kì, tức là biết được khoảng chuyển dịch hay sẽ biết được khóa K, lúc đó ta sẽ giả mã được toàn bộ hệ mã.

Những file lưu trữ dạng ZIP cũng dễ bị tấn công theo kiểu này. Ví dụ, thám mã với file ZIP đã được mã hóa chỉ cần biết một file ZIP chưa mã hóa (được hiểu là biết bản rõ (Known plaintext)), sau đó dùng một số phần mềm sẵn có, họ có thể tính ngay được khóa để giải mã toàn bộ.

Để tìm được file chưa mã hóa này, thám mã có thể tìm kiếm trên Website một file thích hợp, tìm kiếm nó từ một kho lưu trữu, hoặc cố gắng xây dựng lại một file rõ (plaintext file) với những tri thức của tên file từ kho lưu trữ đã được mã hóa.

1.4.1.3. Thám mã với bản rõ được chọn

Thám mã với bản rõ được chọn (Chosen Plaintext attack) (CPA) là mô hình tấn công để giải mã, trong đó rằng thám mã có khả năng chọn một bản rõ tùy ý và biết bản mã tương ứng. Điều này có thể xảy ra khi thám mã chiếm được tạm thời máy lập mã, với hệ mã hóa công khai điều này là hiển nhiên, vì biếtđược khóa công khai thì thám mã có thể mã hóa bất kỳ bản rõ nào mà họ chọn. Đối với các hệ mã đối xứng thì như mô hình trên, thám mã có "quyền" bắt được bất kỳ kẻ đưa thư nào mà họ muốn, ta đừng hiểu theo nghĩa là thám mã bắt tất cả các kẻ đưa thư có cặp bản rõ bản mã tưng ứng, hợp lại sẽ được bản rõ đầy đủ, mà phải hiểu là nếu thám mã có một bản rõ bất kỳ nào đó thì có thể "bắt" được kẻ đưa thư mang bản mã tưng ứng với bản rõ đó.

Ta thấy thám mã ở trường hợp này có "năng lực mạnh" hơn hẳn thám mã ởmục trên, vì ở trên chỉ biết được cặp bản rõ bản mã một cách ngẫu nhiên, còn ở đây có thể biết được do thám mã chọn trước.

Một cách hình thức, thám mã có một plaintext checking oracle (máy tư vấn kiểm tra bản rõ) nhận đầu vào là cặp (m, c) và trả lời có phải là bản mã của m không. Trong trường hợp xấu nhất, thám mã với bản rõ được chọn có thểkhám phá ra khóa bí mật của hệ mã.

Thám mã với bản rõ được chọn trở nên rất quan trọng trong các hệ mã hóa công khai, tại vì các khóa mã hóa đã được công bố và thám mã có thể mã hóa bất kì bản rõ nào mà họ chọn.

Bất kì hệ mã nào an toàn đối với "thám mã với bản rõ được chọn", thì cũng an toàn với "thám mã biết bản rõ" (Known plaintext attack) và "thám mã chỉ biết bản mã"(Ciphertext only attack), đây là vấn đề an toàn kéo theo.

Có hai kiểu phân biệt "thám mã với bản rõ được chọn":

+ "Thám mã với bản rõ được chọn" theo khối, trong đó thám mã chọn tất cả các bản rõ trước, rồi sau đó mới mã hóa.

+ "Thám mã với bản rõ được chọn" thích hợp, trong đó thám mã tạo ra một chuỗi các bản rõ (queries) liên quan lẫn nhau, việc chọn các bản rõ tiếp theo dựa trên thông tin về những mã hóa trước đó.

Những giải thuật mã hóa khóa công khai không ngẫu nhiên (tất định) (ví dụ như RSA dùng thuật toán mã hóa tất định, một bản rõ có duy nhất một bản mã) dễ bị xâm phạm bởi các kiểu thám mã "từ điển" đơn giản. Đó là thám mã xây dựng một bảng các bản rõ và bản mã tương ứng có thể, biết bản mã để tìm ra bản rõ tương ứng, thám mã chỉ cần tìm kiếm trên bảng bản mã, ánh xạ sang bản rõ tương ứng, tất nhiên là với yêu cầu rằng không gian của các bản rõ là "đủ nhỏ", hặc thám mã có thể đoán trước được tập bản rõ nằm trong khoảng đủ nhỏ nào, mà người lập mã sẽ sử dụng.

Vì vậy việc định ngĩa bí mật hoàn toàn cho hệ mã hóa công khai dưới "sự thám mã với bản rõ được chọn" yêu cầu hệ mã phải là hệ mã xác xuất (ví dụ mã hóa ngẫu nhiên).

1.4.1.4. Thám mã với bản mã được chọn

1/. Kiểu tấn công CCA

Thám mã với bản mã được chọn (Chosen ciphertext attack) (CCA) là mô hình tấn công để giải mã, trong đó thám mã chọn bản mã giải mã bản mã đó với khóa chưa được biết. Điều này đạt được khi thám mã giành được quyền kiểm soát tạm thời máy giải mã.

Hai dạng cụ thể của loại tấn công này còn được gọi là tấn công "lunchtime" (CCA1) và tấn công "midnight" (CCA2). Thiết bị cung cấp khả năng giải mã các bản mã được chọn (ngẫu nhiên hay có chủ định), gọi là thiết bị giải mã hoặc máy tư vấn giải mã (decryption oracle).

Kẻ địch có thể giải mã các bản mã đã chọn trước (sử dụng một vài thiết bị giải mã(decryption oracle)), thì có thể đánh bại sự an toàn của một lược đồ mã hóa. Tuy nhiên thám mã với bản mã được chọn có thể kéo theo nhiều ý nghĩa hơn đối với một số sơ đồ mã hóa. Ví dụ trong trường hợp đặc biệt thám mã có thể khôi phục lại được khóa giải mã của lược đồ, bằng cách đưa ra những bản mã chọn trước một cách phù hợp và phân tích những kết quả đã được giải mã. Thành công của thám mã với bản mã được chọn là có thể gây mất an toàn tới lược đồ mã hóa, ngay khi thiết bị giải mã (decryption oracle) trở nên không còn hiệu lực. Vì thế tấn công dạng này có thể có hiệu lực trong cả những trường hợp mà thiết bị giải mã (decryption oracle) không thể được dùng để giải mã một cách trực tiếp các bản mã mục tiêu.

Một số lược đồ an toàn khác có thể không còn tác dụng trước kiểu tấn công này. Ví dụ như lược đồ Elgamal là "an toàn ngữa nghĩa" (semantically secure) trước mô hình tấn công CPA nhưng không còn "an toàn ngữ nghĩa" trước tấn công này.

Những phiên bản trước đây của hệ RSA, dùng giao thức an toàn SSL (Secure socket layer) dễ bị xâm phạm bởi tấn công với bản mã chọn trước thích hợp (Adaptive chosen ciphertext attack), cách tấn công này khám phá ra khóa phiên SSL. Nhà thiết kế ra thẻ thông minh (Smart card) phải có hiểu biết cụ thể về loại tấn công này, những thiết bị này hoàn toàn có thể nằm dưới sự điều khiển của kẻ địch nếu chúng đưa ra một số lượng lớn các bản mã chọn trước để cố gắng khôi phục lại khóa bí mật.

Khi một hệ mã hóa dễ bị xâm phạm với CCA, thì khi thực hiện nên tránh những trường hợp để cho kẻ địch giải các bản mã đã chọn trước (ví dụ như tránh cung cấp cho kẻ địch máy giải mã (decryption oracle). Đôi khi còn khó khăn hơn nữa, khi chỉ cần giải một phần những bản mõ được chọn trước (không cần hoàn toàn), cũng có thể cho cơ hội tấn công hệ mã một cách khôn ngoan. Hơn nữa một vài hệ mã hóa (như RSA) dùng cùng một kỹ thuật để ký và giải mã những thông báo. Việc này cho cơ hội tấn công khi việc "băm" (hashing) không được dùng trên thông báo đã được ký. Vì thế nên dùng những hệ mã, mà có thể là an toàn trước tấn công của CCA, bao gồm RSA-OAEP và Cramer-Shoup.

Hiện tại hai hệ được quan tâm nhất là OAEP và Cramer-Shoup:

- OAEP hiệu quả hơn, dựa trên giả thuyết toán học mạnh hơn là hàm một phía. Cramer-Shoup dùng giả thuyết toán học DDH (Decisionnal Diffie-Hellman) nhưng phải coi hàm băm như một máy tư vấn ngẫu nhiên(RO-Random Oracle) trong chứng minh tính bảo mật.
- Một số nghiên cứu chỉ ra nhược điểm của RO, do vậy Cramer-Shoup được ưa chuộng hơn trong một số trường hợp.

2/. Kiểu tấn công CCA1

Trong kiểu tấn công với bản mã được chọn trước bất kỳ (Non-adaptive chosen ciphertext attack), hay còn gọi là tấn công có bản mã chọn trước không phân biệt (indifferent chosen ciphertext attack) hoặc tấn công "lunchtime" (CCA1), trong đó kẻ địch chỉ có thể chiếm được máy tư vấn giải mã trước khi chọn bản mã cụ thể để tấn công.

Mục tiêu tấn công là thu thập đủ thông tin để làm giảm độ an toàn của hệ mã. Nếu thành công thì có thể khám phá ra khóa bí mật và do đó có thể phá vỡ sự an toàn của hệ mã.

3/. Kiểu tấn công CCA2

a/. Khái niệm kiểu tấn công CCA2

Kiểu tấn công với bản mã được chọn thích hợp (adaptive chosen ciphertext attack) hay còn gọi là tấn công "Midnight" (CCA2), là mở rộng của kiểu tấn công trên, cho phép kẻ địch dùng máy tư vấn giải mã, thậm chí sau khi họ đã họn bản mã để tấn công. Nghĩa là khi biết được bản mã mục tiêi để tấn công, thì vẫn còn khả năng sử dụng máy tư vấn giải mã, tất nhiên với giới hạn rằng không giải mã trực tiếp bản mã mục tiêu trên máy giải mã, vì như thế thì không còn gi để nói.

Mục đích của tấn công này là lấy được các thông tin có ích để giải mã bản mã mục tiêu. Tấn công kiểu này có thể được nâng lên chống lại nhiều lược đồ mã hóa khác nhau (RSA, Elgamal,...). Chúng có thể bị ngăn cản thông qua cách dùng đúng đắn hệ mã có thêm các thông số an toàn hơn hoặc những kiểm soát dư thừa.

Trong kiểu tấn công này, thám mã gửi một số bản mã để giải mã, sau đó dùng chính kết quả của sự giải mã này, để chọn những bản mã tiếp theo.

Đối với các hệ mã hóa công khai, CCA2 được dùng chỉ khi thám mã có *tính chất mềm dẻo của bản mã* (ciphertext of malleability). Đó là bản mã có thể được thay đổi trong những cách cụ thể, để mà có thể dự đoán trước được hiệu quả trên sự giải mã chính thông tin đó. Ví dụ bản mã ở hệ mã RSA là có *tính chất mềm dẻo của bản mã*, vì ta có thể thay đổi bản mã $c = m^e$ thành bản mã $c' = c.2^e$, do ta đã dự đoán trước được kết quả của phép giải mã bản mã c'.

Nghe hơi khó tưởng tượng ra trường hợp chiếm được máy giải mã, nhưng có thể xét trường hợp la thám mã muốn giải mã bản mã, và đóng giả là đối tác giao tiếp với người giải mã, sau đó giả vờ đến nhà người giải mã chơi, và "vô tình" nhìn thấy bản giải mã để trên bàn làm việc.

Hoặc là trong hệ thống có kẻ phản bội tiếp tay cho thám mã.

Hoặc là ví dụ dễ hiểu: Trước kỳ thi vấn đáp, sinh viên (chưa biết đề thi- tất nhiên) có thể hỏi tùy ý giáo sư bất kì câu hỏi nào, và tất nhiên là giáo sư sẽ trả lời kết quả của câu hỏi đó. Ở đây câu hỏi xem là bản mã và câu trả lời là bản rõ giáo sư là máy tư vấn giải mã, sinh viên là thám mã, và đề thi là bản mã mục tiêu mà thám mã (sinh viên) muốn phá. Đó là giai đoạn sinh viên chưa biết đề thi hay thám mã chưa biết bản mã mục tiêu.

Sau khi sinh viên "ăn cắp" được đề thi (bản mã mục tiêu đã được xác định), trong mô hình tấn công CCA1 sinh viên (thám mã) không được quyền hỏi giáo sư (máy tư vấn giải mã) tiếp nữa, trong mô hình tấn công CCA2 thì sinh viên vẫn có quyền hỏi tiếp giáo sư (máy tư vấn giải mã), tất nhiên là không được phép hỏi trên chính đề thi (bản mã mục tiêu). VÌ như vậy thì giáo sư sẽ biết là đề thi đã bị ăn cắp. Điều này cũng phù hợp thực tế, vì sinh viên khôn ngoan sẽ không làm như vậy để giáo sư biết và đổi đề thi.

b/. Tấn công kiểu CCA2 thực tế

CCA2 được quan tâm một cách rộng rãi và trở thành một vấn đề về lý thuyết cho tới năm 1998, khi Daniel Bleychenbacher của phong thí nghiệ Bell đã thử nghiệm thành công một tấn công thực tế.

Để ngăn cản tấn công CCA2, cần thiết phải dùng lược đồ mã hóa mà giới hạn tính mềm dẻo của bản mã (malleability). Một số lượng lớn lược đồ mã hóa đã được đề xuất, nhưng hiệu quả và dễ cài đặt trong thực tế là hai lược đồ Cramer-shoup và OAEP.

c/. Kiểu tấn công CCA mở rộng: (i, j) – CCA

Kẻ địch (Adversary) được gọi là (i, j) chosen-ciphertext adversary ((i, j)-CCA), nếu nó có thể truy vấn máy tư vấn giải mã (decryption oracle) i lần trước khi bản mã mục tiêu được biết, j lần sau khi bản mã mục tiêu được biết, và tất nhiên là vẫn có giới hạn là không được truy vấn trên chính bản mã mục tiêu.

Kiểu tấn công này chỉ khác kiểu tấn công CCA2 là giới hạn một cách chính xác số lần kẻ địch có thể truy vấn trên máy giải mã (với CCA2 số lần kẻ địch truy vấn có thể là tùy ý).

Kết luận

Mô hình tấn công CCA2 là mạnh nhất hiện nay, hệ mã nào mà đạt được an toàn trước tấn công này, thì cũng đạt an toàn trước các kiểu tấn công còn lại. Đây là vấn đề an toàn kéo theo.

Mô hình tấn công bản chất là ta giả sử cho thám mã có được "năng lực" gì, chỉ biết bản mã (COA), biết bản rõ (KPA),... "năng lực" còn hiểu là "tình huống xấu nhất" có thể xảy ra trong thực tế.

Cho đến nay "tình huống xấu nhất" có thể xảy ra mà các nhà lập mã có thể nghĩ đến, đó là thám mã chiếm được tạm thời "máy giải mã" tương đương với mô hình tấn công CCA2. Vì vậy nếu hệ mã mà an toàn với "hình huống xấu nhất ", thì hiển nhiên cũng an toàn với các "tình huống tốt hơn".

1.4.2. Tính an toàn của một hệ mật mã

1.4.2.1. An toàn một chiều (One - Wayness)

Đây là yêu cầu cơ bản về tính an toàn của hệ mã hóa khóa công khai. Khi Bob dùng khóa công khai cua Alice mã hóa thông tin của mình và gửi cho Alice, thám mã bằng cách nào đó lấy được bản mã, thì cũng "khó" thể giải được bản mã (việc giải bản mã là không thể thực hiện được trong thời gian chấp nhận), nếu không biết khóa bí mật mà Alice nắm giữ.

Một cách hình thức với bất lỳ thám mã A trong việc tìm ra bản rõ từ bản mã cho trước, mà không có khóa bí mật, thì xác suất thành công là *"không đáng kể"* trên không gian xác xuất M x Ω, trong đó M là không gian của các bản rõ (message) và Ω là không gian những thành phần ngẫu nhiên r.

Kí hệu $Succ^{ow}$ (A) là xác xuất thành công của kẻ thám mã A sử dụng giải thuật thời gian đa thức để tìm ra bản rõ m.

G_k là giải thuật tạo cặ khóa công khai và bí mật (p_k và s_k), có đầu vào là chuỗi $z \in \{0,1\}^k$, có nghĩa z là chuỗi có độ dài k bít, mỗi bit có thể là bit 0 hoặc bit 1.

E là giải thuật mã hóa, $E_{pk}(m)$ là bản mã của m. A là kẻ thám mã dùng giả thuật thời gian đa thức có hai đầu vào là khóa công khai p_k và bản mã $E_{pk}(m)$.

$Succ^{ow}(A) = Pr[(G_k(z) \rightarrow (p_k, s_k), M \xrightarrow{R} m): A(p_k, E_{pk}(m)) = m] < \varepsilon$, trong đó ε là lượng không đáng kể.

Nếu giải thuật mã hóa là đơn định (một đầu vào duy nhất có một đầu ra), thì Ω = Φ. Nếu Ω = Φ thì để đảm bảo tính an toàn, không gian M phỉa lớn. Đôi khi M lớn, nhưng nếu thám mã đoán trước được tần suất của không gian con trong M, hay được dùng làm bản rõ, thì cũng dễ gây nguy hiểm.

Thực tế những hệ mã có Ω = Φ (không phỉa là hệ mã xác xuất), thì có tính an toàn không cao, và ít được dùng trong thực tế.

1.4.2.2. An toàn ngữ nghĩa (Semantic Security)

Một cách không hình thức, một hệ thống được gọi là an toàn ngữ nghĩa (Semantic Security), nếu bất cứ khi nào thám mã có thể tính toán bản rõ với nản mã cho trước, thì cũng có thể làm việc đó mà không cần biết trước bản mã. Nói cách khác, việc biết bản mã cũng không đưa lại dù chỉ là một bit thông tin cho thám mã.

Sematic Security cũng tương đương với khái niệm an toàn đa thức (Polynomial Security).

An toàn đa thức có nghĩa là cho trước bản mã, không thể tính toán được bất cứ thứ gì về bản rõ trong thời gian đa thức.

Gọi f là một hàm bất kì được định nghĩa trên không gian các bản tin M. Chúng ta nói f(m) là bao gồm những thông tin về bản tin $m \in M$. những hàm f điển hình được quan tâm như hàm băm, hàm xác thực,...

Chúng ta muốn rằng việc trích rút bất kì thông tin nào của những bản tin từ sự mã hóa của chúng là "khó", thậm chí ngay cả khi xác xuất phân bố của không gian bản tin được biết.

Với tất cả $m \in M$, đặt $Pr_m = Prob(x = m \mid x \in M)$ là xác xuất mà x = m với $x \in M$.

Xét tập V = f(M), định nghĩa $\Pr^M = \max_{v \in V}(\sum_{m \in f^{-1}(v)} \Pr_m)$ và v^M là giá trị thuộc V= f(M) mà ở đó đạt xác xuất lớn nhất Pr^M. Gọi E là giải thuật mã hóa. Xét ba trường hợp sau với E là công khai được thám mã biết trước.

Trường hợp 1: Chọn ngẫu nhiên $m \in M$ (mỗi $x \in M$ có xác xuất là Pr_x). Trong trường hợp này thám mã phải dự đoán f(m) (thông tin về bản rõ m) mà không được biết m.

Nếu thám mã luôn luôn dự đoán $f(m) = v^M$ thì dự đoán đó sẽ luôn đúng với xác xuất là Pr^M, và theo định nghĩa ở trên ta thấy rằng sẽ không có cách nào khác cho thám mã có thể dự đoán với xác xuất cao hơn.

***Trường hợp 2*:** Chọn ngẫu nhiên $m \in M$. Tính bản mã $\alpha \in E(m)$. Cho thám mã biết α. Bây giờ thám mã phải dự đoán f(m).

***Trường hợp 3*:** Cho thám mã chọn hàm f_E được định nghĩa trong M. Chọn ngẫu nhiên $m \in M$. Tính bản mã $\alpha \in E(m)$. Cho thám mã biết α. Bây giờ thám mã phải dự đoán f(m).

Ở đây f(m) là tập hợp những "thông tin" về bản rõ m, ở trường hợp tốt nhất đó chính là bản rõ m.

Ký hiệu G_k là giải thuật tạo khóa công khai và bí mật, E là giải thuật mã hóa, D là giải thuật giải mã.

Một cách không hình thức, chúng ta nói rằng lược đồ $\Pi = (G_k, E, D)$ là lược đồ mã hóa khóa công khai an toàn ngữ nghĩa, nếu thám mã trong trường hợp 3 dự đoán giá trị f(m) với xác xuất không lớn hơn trong trường hợp 1.

Kết luận:

Một hệ thống an toàn ngữ nghĩa là một hệ thống mà cho kẻ thám mã biết bản mã thì cũng không đem lại dù chỉ một bit thông tin cho kẻ thám mã.

Ví dụ trong một chiến dịch quân sự bản rõ bao gồm: thông tin về số quân, giờ đánh, vị trí đánh, vũ khí hạng nặng có những gì.

An toàn ngữ nghĩa có nghĩa là cho kẻ thám mã biết bản mã của bản rõ trên, thì cũng không thể tìm ra một chút thông tin gì về bản rõ, ví dụ như không thể biết được về số quân chẳng hạn.

Thông thường rong định nghĩa an toàn cổ điển, hệ mã được gọi là an toàn nếu cho trước bản mã không tìm được bản rõ tương ứng. Nhưng ở đây định ngĩa chặt hơn là, chẳng những không tìm được bản rõ tương ứng mà chỉ một phần nhỏ (một bit), thông tin của bản rõ tương ứng cũng không thể tìm được, như ở trường hợp trên chỉ thông tin về số quân cũng không biết được.

1.4.2.3. Tính không phân biệt được (Indistinguishability: IND)

Mục tiêu của mã hóa là duy trì tính bí mật của thông tin: Thám mã sẽ khó thể dùng bản mã để biết được thông tin về bản rõ tương ứng ngoại trừ độ dài của bản mã đó. Chúng ta định nghĩa như sau:

Giả sử A = (A_1, A_2) là hai giai đoạn tấn công của thám mã. Giải thuật A_1 chạy trên đầu vào là khóa công khai p_k, cho đầu ra là bộ ba (x_0, x_1, s), hai thành phần đầu tiên x_0 và x_1 là những thông tin có cùng độ dài, và thành phần cuối cùng là s là thông tin về trạng thái mà thám mã muốn duy trì (có thể bao gồm p_k). Chọn ngẫu nhiên x_0 hoặc x_1, gọi là x_b. Mã hóa x_b dùng khóa công khai p_k ta thu được bản mã y.

Giải thuật A_2 với đầu vào là y, s, x_0, x_1 cho đầu ra là b (có nghĩa là xác định xem bit b là 1 hay là 0).

Cho đơn giản, ta định nghĩa đồng thời các kiểu tấn công CPA, CCA1 và CCA2. Sự khác nhau duy nhất nằm ở chỗ có hoặc không A_1 và A_2 được cho trước những máy tư vấn giải mã (decryption oracles).

Ta đặt chuỗi atk (viết tắt của từ attack), là đại diện cho cả cpa, cca1, cca2, trong khi ATK tương ứng đại diện cho CPA, CCA1, CCA2.

Khi ta nói $O_i = \varepsilon$, $i \in \{1,2\}$, có nghĩa O_i là hàm, trong đó trên bất kì đầu vào nào đều trả về chuỗi rỗng ε, tức là không có tư vấn (hàm O_i ở đây chẳng qua là đại diện cho việc có hoặc không máy tư vấn giải mã).

Ký hiệu $Adv^{atk}(A)$ là xác xuất thành công để dự đoán giá trị của b của kẻ thám mã A dùng giải thuật thời gian đa thức, cùng với phương pháp thám mã atk (như ta ký hiệu ở trên nếu atk là cpa thì đó là phương pháp thám mã cpa, tương tự nếu atk là cca1 thì kẻ thám mã đó dung phương pháp thám mã cca1,...).

Ký hiệu G_k là giải thuật tạo cặp khóa công khai và bí mật (p_k và s_k), có đầu vào là chuỗi $z \in \{0,1\}^k$, có nghĩa z là chuỗi có độ dài k bit, mỗi bít có thể là bit 0 hoặc bit 1.

E là giải thuật mã hóa, D là giải thuật giải mã, Pr[…] là xác xuất mà A cho ra dự đoán b là bằng 0 hoặc bằng 1.

Như ở trường hợp dưới $A_2^{02}(x_0, x_1, s, y) = b$ là sự kiện của xác suất Pr.

Hay hiểu theo một cách khác Pr[… : $A_2^{02}(x_0, x_1, s, y) = b$] chính là xác suất của sự kiện $A_2^{02}(x_0, x_1, s, y) = b$.

Còn ký hiệu $A_2^{02}(x_0, x_1, s, y) = b$ có nghĩa là thám mã A chạy trong giai đoạn hai (dùng giải thuật A_2 như trên ta đã định nghĩa), có đầu vào là chuỗi z, và dùng hàm tư vấn O_2, có đầu ra là b.

Kết luận:

An toàn IND có nghĩa là việc dự đoán từng bit một của chuỗi bản rõ đó của kẻ thám mã A dùng giải thuật thời gian đa thức chỉ có xác xuất là ½ + ε, trong đó ε là một lượng "không đáng kể".

Nếu kẻ thám mã dùng phương pháp thám mã atk (với atk là ký hiệu của cpa, cca1,cca2), mà hệ mã vẫn đạt an toàn IND, thì ta nói hệ mã đạt an toàn IND trước phương pháp thám mã atk.

Mức an toàn (i,j,t) – IND:

Chúng ta quan tâm đến trường hợp cụ thể hơn bằng việc đưa ra mức độ an toàn (i,j)- IND, trong đó thám mã có thể hỏi máy tư vấn giải mã (decryption oracle) nhiều nhất i (j tương ứng) câu hỏi trước khi (sau khi tương ứng) nhận bản mã mục tiêu (ở đây là y).

Có nghĩa là trước khi nhận bản mã mục tiêu, được hỏi tối đa là i câu hỏi, sau khi bản mã mục tiêu, thì chỉ được hỏi tối đa là j câu hỏi tới máy tư vấn giải mã, tất nhiên là không hỏi máy tư vấn giải mã bản mã mục tiêu. Rõ ràng là trong tấn công CCA1 thì j = 0, và trong tấn công CPA thì i = j = 0.

Mỗi câu hỏi đề có vai trò khác nhau, không thể thay thế câu hỏi trước khi nhận bản mã mục tiêu bằng câun hỏi sau khi nhận bản mã mục tiêu được.

Cụ thể hơn nữa, người ta đưa ra khái niệm (i,j,t)- IND, trong đó kẻ thám mã chỉ có thể thực hiện một cách chính xác i (j tương ứng) câu hỏi giải mã trước khi (sau khi tương ứng), nhận bản mã mục tiêu trong phạm vi thời gian t.

Gọi A là kẻ thám mã, ta ký hiệu $Adv^{atk}(A,t)$ là lợi thế lớn nhất (max) trên tất cả các kẻ thám mã A chạy trong thời gian giới hạn t.

Ta nói lược đồ Π là an toàn (t, ε)- IND nếu $Adv^{atk}(A,t) \leq \varepsilon$.

1.4.2.4. An toàn ngữ nghĩa tương đương với IND

Hệ mã RSA có sơ đồ sau:

Chọn số nguyên tố p, q.

n = p.q. $\Phi(n) = (p - 1)(q - 1)$.

(e,d) là cặp khóa (công khai và bí mật).

$ed = 1 \bmod \Phi(n)$.

mã hóa $c = m^e \bmod n$.

Giải mã $m = c^d \bmod n$.

RSA là đơn định, do đó nó không thể an toàn ngữ nghĩa. Ta đã biết an toàn ngữ nghĩa là việc biết bản mã cũng không mang lại dù chỉ một bit thông tin cho bản mã. Nhưng RSA là hàm đơn định, tức là một bản rõ chỉ có duy nhất một bản mã, nên nếu thám mã biết trước cặp bản mã và bản rõ, thì sau này nếu thấy một bản mã giống hệt như vậy, nó dễ dàng suy ra bản rõ. Mâu thuẫn với khái niệm an toàn ngữ nghĩa.

Hệ mã xác xuất do ngẫu nhiên chọn r, nên một bản rõ có thể có nhiều bản mã. Do tính ngẫu nhiên của việc chọn r nên việc có hai bản mã giống hệt nhau của cùng một bản rõ trong hai lần mã hóa khác nhau là rất khó xảy ra (xác xuất không đáng kể).

Một lược đồ mã hóa công khai xác suất bao gồm:

* G_k giải thuật tạo khóa, là một giải thuật xác xuất mà trên đầu vào là một chuỗi bit 0 hoặc 1 có độ dài n (có nghĩa là: $\{0,1\}^n$), n là tham số an toàn, đầu ra là cặp (p_k, s_k) (p_k là khóa công khai và s_k là khóa bí mật).
* E: hàm mã hóa, nhận ba đầu vào: thứ nhất là khóa công khai p_k, thứ hai là $b \in \{0,1\}$, một chuỗi ngẫu nhiên r có độ dài p(n) ($r \in \{0,1\}^{p(n)}$), với p là một đa thức nào đó. $Ep_k(b, r)$ có thể tính toán trong thời gian đa thức.
* D: hàm giải mã, nhận hai đầu vào: c là bản mã và khóa bí mật s_k.
* s_k được tạo ra bởi G_k, $Ds_k(c)$ có thể tính toán trong thời gian đa thức.
* Nếu đầu ra của G_k là (p_k, s_k) thì

  $\forall b \in \{0,1\}, \forall i\ r \in \{0,1\}^{p(n)}$ ta có: $Ds_k(Ep_k(b,r)) = b$

* Hệ thống có tính chất không phân biệt (indistinguishability): Với tất cả giải thuật thời gian đa thức M, với tất cả $c > 0$, $\exists\ j_0$ để cho $\forall\ j > j_0$ và

$\Pr [M(p_k, Ep_k(0, r)) = 0] < ½ + 1/ j^c$.

Có nghĩa là xác xuất để cho ra dự đoán đúng giá trị của bản rõ, từ bản mã và khóa công khai là bé hơn $½ + 1/ j^c$.

$1/ j^c$ là bé không đáng kể.

Đây là lược đồ mã hóa bit, một lược đò cụ thể của dạng này được dùng trong thực tế là lược đồ mã hóa xác xuất của Goldwasser-Micali đề xuất năm 1984([1]).

Kết luận:

Hai khái niệm an toàn ngữ nghĩa và tính không phân biệt được là tương đương với nhau.

1.4.2.5. Khái niệm an toàn mạnh nhất IND-CCA

Chúng ta bắt đầu bằng việc mô tả kịch bản các giai đoạn tấn công:

***Giai đoạn 1*:** Giải thuật sinh khóa tạo ra khóa công khai và khóa bí mật cho hệ mã. Thám mã biết khóa công khai, nhưng không biết khóa bí mật.

***Giai đoạn 2*:** Thám mã tạo ra một chuỗi các truy vấn tùy ý tới máy tư vấn giải mã (decryption oracle), mỗi truy vấn là một bản mã, sẽ được giải mã bởi máy tư vấn giải mã (dùng khóa bí mật). Kết quả giải mã của máy tư vấn sẽ được đưa cho thám mã. Ở đây thám mã có thể tự do xây dựng những bản mã, để truyền tới máy tư vấn giải mã.

Ví dụ: Trước một kỳ thi vấn đáp, sinh viên (chưa biết đề thi – tất nhiên) có thể hỏi tùy ý giáo sư bất kì câu hỏi nào, và tất nhiên là giáo sư sẽ trả lời kết quả của câu hỏi đó. Ở đây câu hỏi ta xem là bản mã và câu trả lời sẽ là bản rõ, giáo sư là máy tư vấn giải mã, và đề thi là bản mã mà thám mã (sinh viên) muốn phá.

***Giai đoạn 3*:** Thám mã chuyển hai message x_0 và x_1 cho máy tư vấn mã hóa (encryption oracle), máy tư vấn mã hóa sẽ chọn $b \in \{0,1\}$ ngẫu nhiên, sau đó mã hóa x_b và chuyển kết quả mã hóa là bản mã y* cho thám mã, thám mã có thể chọn tùy ý x_0 và x_1, nhưng với điều kiện x_0 và x_1 phải có cùng độ dài.

***Giai đoạn 4*:** Thám mã có quyền tạo các try vấn (bản mã y) tùy ý tới máy tư vấn giải mã và nhận câu trả lời, tất nhiên giới hạn rằng truy vấn (bản mã y) phải khác y* (Trong IND-CCA2 có giai đoạn này, trong IND-CCA1 không có giai đoạn này).

***Giai đoạn 5*:** Thám mã cho đầu ra là một giá trị u $\in$ {0,1} là kết quả dự đoán b của thám mã.

"Lợi thế" (Advantage) của thám mã trong kịch bản tấn công trên được định nghĩa là: $\left| \Pr[b = u] - 1/2 \right|$.

"Lợi thế" ở đây là bằng trị tuyệt đối xác suất để kẻ thám mã cho ra dự đoán đúng giá trị của b, trừ đi ½ , có nghĩa "Lợi thế" đúng bằng lượng $1/ j^c$.

Định nghĩa:

Một hệ mã được gọi là an toàn IND- CCA, nếu bất kì thám mã CCA nào (tức là thám mã có năng lực CCA), dùng giải thuật thời gian đa thức để phá hệ mã, đều có "lợi thế" là không đáng kể.

Hay hệ mã mà cho phép kẻ thám mã có được năng lực CCA, vẫn đạt an toàn IND thì ta nói hệ mã đó đạt an toàn IND-CCA.

Khái niệm an toàn IND-CCA còn được gọi là *an toàn chống lại tấn công với bản mã được chọn trước* (SEcurity against chosen ciphertext attack).

***Kết luận*:**

Đây là khái niệm an toàn mạnh nhất trong các khái niệm an toàn có hiện nay.

Chương 2: TẤN CÔNG BẢN MÃ

2.1. TẤN CÔNG HỆ MÃ HÓA RSA

2.1.1. Hệ mã hóa RSA

1/. Sơ đồ

* Thuật toán sinh khóa:

+ Chọn bí mật hai số nguyên tố lớn p và q có giá trị xấp xỉ nhau.

+ Tính n = p.q. Đặt P = C = Z_n. Và tính bí mật $\Phi(n) = (p - 1).(q - 1)$.

+ Chọn một số ngẫu nhiên b, $1 < b < \Phi(n)$, sao cho $gcd(b, \Phi(n)) = 1$.

+ Sử dụng thuật toán Euclide để tính số a, $1 < a < \Phi(n)$,
sao cho $a.b \equiv 1 (\text{mod } \Phi(n))$.

+ Khóa công khai là K' = (n, b), Khóa bí mật là K" = a.

* Thuật toán mã hóa RSA

(i). Lập mã:

+ Lấy khóa công khai K' = (n, b) theo thuật toán trên.

+ Chọn một bản rõ $x \in P$.

+ Tính bản mã $y = e_k(x) = x^b \text{ mod } n$.

(ii). Giải mã:

Với bản mã $y \in C$

Sử dụng khóa bí mật K'' = a, để giải mã: $x = d_k(y) = y^a \text{ mod } n$

2/. Ví dụ:

* Sinh khóa:

+ Chọn bí mật số nguyên tố p = 2357, q = 2551.

+ Tính n = p * q = 6012707, và tính bí mật $\Phi(n) = (p - 1).(q - 1) = 6007800$.

+ Chọn ngẫu nhiên b, $1< b < \Phi(n)$ là số nguyên tố với $\Phi(n)$ tức UCLN(b, $\Phi(n)$) = 1, ví dụ chọn b = 3674911.

+ Tính a là phần tử nghịch đảo của b theo modulo $\Phi(n)$: $a.b \equiv 1 (\text{mod } \Phi(n))$. Ta được a = 422191.

+ Khóa công khai K' = (6012707, 3674911). Khóa bí mật K" = (422191).

* Mã hóa:

(i). Lập mã

+ Cho bản rõ x = 5234673.

+ Tính được bản mã $y = x^b \bmod n = 3650502$.

(ii). Giải mã

Từ bản mã y, tính được bản rõ $x = y^a \bmod n = 5234673$.

2.1.2. Các loại tấn công vào hệ mã hóa RSA.

2.1.2.1. Tấn công loại 1: Tìm cách xác định khóa bí mật

1/. Trường hợp 1: Khi thám mã chỉ biết bản mã

Khi biết được n, kẻ thám mã sẽ tìm cách tính giá trị của p, q theo thuật toán phân tích thành thừa số nguyên tố, từ đó chúng sẽ tính $\Phi(n) = (p-1)(q-1)$. Khi đã tính được $\Phi(n)$ chúng sẽ tính được khóa bí mật a theo công thức: $a.b \equiv 1 (\text{mod } \Phi(n))$. Khi đã tính được khóa bí mật thì chúng sẽ giải mã được bản mã và tìm ra bản rõ.

→ Giải pháp phòng tránh:

Chọn số nguyên tố p,q lớn, để việc phân tích n thành tích 2 thừa số nguyên tố là khó có thể thực hiện được trong thời gian thực. Thường sinh ra các số lớn (khoảng 100 chữ số) sau đó kiểm tra tính nguyên tố của nó.

2/. Trường hợp 2: Khi thám mã biết bản mã và cả bản rõ

Kẻ thám mã biết một bản mã Y cùng với bản rõ tương ứng X. Vì đã biết được bản rõ rồi nên việc chính của thám mã là phải xác định được khóa bí mật K" đang sử dụng, để giải mã các gói tin đã mã hóa mà nó bắt được sau này.

Kẻ thám mã sẽ dựa vào bài toán tính logarit thông thường để tính ra khóa bí mật K" = a theo công thức: $K'' = \log_y^x (\text{mod } n - 1)$.

Từ đây toàn bộ thông tin được mã hóa bằng khóa K' đều bị thám mã bắt được.

→ Giải pháp phòng tránh:

Sử dụng khóa khác nhau ở mỗi lần mã hóa, để nếu kẻ thám mã biết được khóa giải mã ở lần mã hóa này, thì khi bắt được các gói tin ở lần mã hóa sau, chúng cũng không thể sử dụng khóa đó để giải mã tiếp được nữa.

3/. Trường hợp 3: Khi thám mã có bản rõ được chọn trước

Kẻ thám mã có thể chọn một bản rõ X, và biết bản mã Y tương ứng (khi kẻ thám mã chiếm được (tạm thời) máy lập mã), việc xác định khóa bí mật lại quay về trường hợp thứ hai.

→ ***Giải pháp phòng tránh:***

Sử dụng khóa khác nhau ở mỗi lần mã hóa, để nếu kẻ thám mã biết được khóa giải mã ở lần mã hóa này, thì khi bắt được các gói tin ở lần mã hóa sau, chúng cũng không thể sử dụng khóa đó để giải mã tiếp được nữa.

4/. Trường hợp 4: Khi thám mã có bản mã được chọn trước

Kẻ thám mã có thể chọn một bản mật mã Y, và biết bản rõ tương ứng X. Trong trường hợp này việc xác định khóa bí mật lại quay về trường hợp thứ hai.

→ ***Giải pháp phòng tránh:*** như trường hợp 2.

2.1.2.2. Tấn công dạng 2: Tìm cách xác định bản rõ

1/. Dùng modul n chung

Giả sử có hai người tham gia A và B cùng sử dụng một modul chung n trong khóa công khai của mình, chẳng hạn A chọn khóa công khai (n, e) và giữ khóa bí mật d, B chọn khóa công khai (n, a) và giữ khóa bí mật b. Một người tham gia thứ ba C gửi một văn bản cần bảo mật x đến cả A và B thì dùng các khóa công khai nói trên để gửi đến A bản mật mã $y = x^e \bmod n$ và gửi đến B bản mật mã $z = x^a \bmod n$. Ta sẽ chứng tỏ rằng một người thám mã O có thể dựa vào những thông tin n, e, a, y, z trên đường công khai mà phát hiện ra bản rõ x như sau:

(i). Tính $c = e^{-1} \bmod a$,

(ii). Sau đó tính $h = (ce - 1)/ a$,

(iii). Và ta được $x = y^c (z^h)^{-1} \bmod n$.

Thực vậy theo định nghĩa trên, ce – 1 chia hết cho a, và tiếp theo ta có:

$y^c (z^h)^{-1} \bmod n = x^{ce}.(x^{a(ce-1)/a})^{-1} \bmod n = x^{ce}.(x^{ce-1})^{-1} \bmod n = x \rightarrow$ bản rõ cần tìm.

Như vậy, trong trong trường hợp này việc truyền tin bảo mật không còn an toàn nữa.

→ Giải pháp phòng tránh:

Dùng *modul n* khác nhau cho mỗi người tham gia.

2/. Dùng số mũ lập mã bé

Để cho việc tính toán hàm lập mã được hiệu quả, ta dễ có xu hướng chọn số mũ b của hàm lập mã là một số nguyên bé, chẳng hạn b = 3. Tuy nhiên, nếu trong một mạng truyền tin bảo mật dùng các hệ mã RSA, nếu có nhiều người cùng chọn số mũ lập mã b bé giống nhau thì sẽ có nguy cơ bị tấn công bởi thám mã như sau: Giả sử có ba người tham gia chọn ba khóa công khai là (n_1, b), (n_2, b), (n_3, b) với cùng số mũ b = 3. Một người tham gia A muốn gửi một thông báo x cho cả ba người đó, và để bảo mật, gửi bản mã $y_i = x^3 \bmod n_i$, cho người thứ i. Ba modul n_i là khác nhau, và có phần chắc là từng cặp nguyên tố với nhau. Một người thám mã có thể dùng định lý số dư Trung Quốc để tìm một số m ($0 \le m \le n_1n_2n_3$) thỏa mãn:

$$\begin{cases} m \equiv y_1 \bmod n_1 \\ m \equiv y_2 \bmod n_2 \\ m \equiv y_3 \bmod n_3 \end{cases}$$

vì $x \le n_i$, nên $x^3 \le n_1 n_2 n_3$, do đó ắt có $m = x^3$. Vậy là thám mã đã đưa được bài toán tìm căn bậc ba theo nghĩa đồng dư $mod n_i$ về bài toán tìm căn bậc ba theo nghĩa số học thông thường: tìm căn bậc ba của m thám mã được bản rõ x.

→ *Giải pháp phòng tránh:*

Chọn các số lập mã và giải mã: b và a là những số nguyên lớn, có kích cỡ lớn gần như bản thân số n.

3/. Lợi dụng tính nhân của hàm lập mã

Ta chú ý rằng hàm lập mã $f(x) = x^e \bmod n$ có tính nhân (multiplicative property), nghĩa là $f(x.y) = f(x).f(y)$. Dựa vào tính chất đó, ta thấy rằng nếu y là mật mã của bản rõ x , thì $\bar{y} = y.u^e \bmod n$ sẽ là bản mật mã của bản rõ xu. Do đó, khi lấy được bản mật mã y, để phát hiện bản rõ x người thám mã có thể chọn ngẫu nhiên một số u rồi tạo ra bản mã $\bar{y}$, và nếu người thám mã có khả năng thám mã theo kiểu (có bản mã được chọn), tức có khả năng với $\bar{y}$ được chọn tìm ra bản rõ tương ứng là $\bar{x} = xu$, thì bản rõ gốc cần phát hiện sẽ là $x = \bar{x}.u^{-1} \bmod n$. Tất nhiên, khả năng người thám mã có năng lực giải quyết bài toán thám mã theo kiểu có bản mã được chọn là rất hiếm, nhưng dầu sao đấy cũng là một trường hợp mà vấn đề bảo mật dễ bị tấn công, ta không thể không tính đến để tìm cách tránh.

→ *Giải pháp phòng tránh:*

Không để thám mã có khả năng thám mã theo kiểu có bản mã được chọn.

2.2. TẤN CÔNG HỆ MÃ HÓA ELGAMAL

2.2.1. Hệ mã hóa ELGAMAL

1/. Sơ đồ

* Thuật toán sinh khóa:

+ Sinh ngẫu nhiên số nguyên tố lớn p và α là phần tử sinh của Z^*_p.

+ Chọn ngẫu nhiên số nguyên a, $1 \leq a \leq p-2$, tính $h = \alpha^a \bmod p$.

+ Khóa công khai là (p, α, α^a). Khóa bí mật là (a).

* Thuật toán mã hóa:

(i) Lập mã:

+ Lấy khóa công khai (p, α, α^a) theo thuật toán trên.

+ Chọn một bản rõ x, trong khoảng [0, p-1].

+ Chọn ngẫu nhiên số nguyên k, $1 \leq k \leq p-2$.

+ Tính $\gamma = \alpha^k \bmod p$ và $\delta = x.(h)^k \bmod p$.

+ Nhận được bản mã là (γ, δ).

(ii) Giải mã:

+ Sử dụng khóa bí mật (a) và tính $\gamma^{p-1-a} \bmod p$.

+ Lấy bản rõ: $x = \delta.(\gamma^a)^{-1} \bmod p$.

Vì $((\gamma^{-a}).\delta \equiv (\alpha^{-ak}).x.(\alpha^{ak}) \equiv x \pmod p)$.

2/. Ví dụ:

Chọn số nguyên tố p = 2357 và một phần tử sinh $\alpha = 2$ của tập Z^*_{2357}. Chọn khóa bí mật a = 1751 và tính $\alpha^a \bmod p = 2^{765} \bmod 2579 = 949$.

Khóa công khai $(p = 2579, \alpha = 2, \alpha^a = 949)$.

(i) Lập mã: Mã hóa bản rõ x = 1299, chọn số nguyên k = 853.

$\gamma = 2^{853} \bmod 2579 = 453$.

$\delta = 1299 * 949^{853} \bmod 2579 = 2396$.

Bản mã là: (453, 2396).

(ii) Giải mã:

$x = (453^{-765}) * 2396 \bmod 2579 = 1299$.

2.2.2. Các dạng tấn công vào hệ mã hóa ELGAMAL

2.2.2.1. Tấn công dạng 1: Tìm cách xác định khóa bí mật

1/. Trường hợp 1: Sử dụng modul p nhỏ

Khi sử dụng số nguyên tố p nhỏ, thì tập Z^*_p nhỏ, do đó việc tìm được phần tử sinh $\alpha \in Z^*_p$ cũng không khó khăn lắm. Khi biết được α và biết được giá trị α^a từ khóa công khai thám mã sẽ tính được khóa bí mật a.

→ Giải pháp phòng tránh:

Chọn modul p là số nguyên tố sao cho *p – 1* có ít nhất một ước số nguyên tố lớn. Điều đó là thực hiện được nếu số nguyên tố p được chọn là số nguyên tố Sophie Germain (tức có dạng *2q+1*, với *q* cũng là số nguyên tố lớn).

2/. Trường hợp 2: Bị lộ số k được dùng

Do cẩu thả trong việc sử dụng số ngẫu nhiên *k*, đặc biệt là khi *để lộ số k được dùng*. Thì khóa bí mật *a* được tính ra ngay theo công thức:

$$a = (x - k\,y_2)\,y_1^{-1} \bmod (p-1).$$

Như vậy, một kẻ thám mã có khả năng tấn công theo kiểu "biết cả bản rõ", có thể phát hiện ra khóa bí mật *a* nếu biết *k*.

→ Giải pháp phòng tránh:

Cẩn thận trong việc sử dụng số ngẫu nhiên k, không để lộ số k được dùng.

2.2.2.2. Tấn công dạng 2: Tìm cách xác định bản rõ

Dùng cùng một số k cho nhiều lần lập mã: Giả sử dùng cùng một số ngẫu nhiên k cho hai lần lập mã, một lần cho x_1, một lần cho x_2, và được các bản mã tương ứng (y_1, y_2) và (z_1, z_2). Vì dùng cùng một số *k* nên $y_1 = z_1$. Và do đó theo công thức lập mã ta có: $z_2/y_2 = x_2/x_1$, tức *là* $x_2 = x_1.z_2/y_2$. Như vậy, một người thám mã, một lần biết cả bản rõ dễ dàng phát hiện được bản rõ trong các lần sau.

→ Giải pháp phòng tránh:

Mỗi lần lập mã thì sử dụng một số k khác nhau.

2.3. TẤN CÔNG HỆ MÃ HÓA: DỊCH CHUYỂN

2.3.1. Mã dịch chuyển

1./ Sơ đồ

Đặt P = C = K = Z_{26}. Bản mã y và bản rõ x$\in Z_{26}$.

Với khóa k$\in$ K, ta định nghĩa:

Hàm mã hóa: $y = e_k(x) = (x + k) \bmod 26$.

Hàm giải mã: $x = d_k(y) = (y – k) \bmod 26$.

2/. Ví dụ:

* Bản rõ chữ: T O I N A Y T H A V I R U S

* Chọn khóa: k = 3

* Bản rõ số: 19 14 8 26 13 0 24 26 19 7 0 26 21 8 17 20 18

* Với phép mã hóa $y = e_k(x) = (x + k) \bmod 26 = (x + 3) \bmod 26$, ta nhận được:

* Bản mã số: 22 17 11 3 16 3 1 3 22 10 3 3 24 11 20 23 21

* Bản mã chữ: W R L D Q D B D W K D D Y L U X V

Với phép giải mã $x = d_k(y) = (y - k) \bmod 26 = (y - 3) \bmod 26$, ta nhận lại được bản rõ số, sau đó là bản rõ chữ.

2.3.2. Dạng tấn công vào mã dịch chuyển: Tìm cách xác định khóa k

Trong tiếng Anh khóa K = Z_{26}. Do chỉ có 26 khóa nên việc thám mã có thể thực hiện phá mã theo kiểu "biết bản mã" bằng duyệt tuần tự các khóa cho tới khi nhận được bản rõ có nghĩa.

Ví dụ: Khi thám mã có trong tay một bản mã là: "qnwcxrcqdkjh". Thám mã sẽ thực hiện duyệt lần lượt từ k =1 → k = 26 để tìm ra bản rõ

Với k = 1 được bản rõ: "pmvbwqbpcjig" không có nghĩa.

Thám mã tiếp tục thử với k = 2, 3,.....8

Khi thử đến k = 9 được bản rõ: "hentoithubay" có nghĩa → thám mã thành công.

→ Giải pháp phòng tránh:

Mở rộng vùng không gian khóa lớn. Ví dụ như bảng chữ cái tiếng Việt có thanh (gồm 94 ký tự), thì việc thử tất cả các khóa cũng lâu hơn bảng tiếng Anh.

2.4. TẤN CÔNG MÃ THAY THẾ

2.4.1. Mã thay thế

1/. Sơ đồ

Đặt **P = C = Z_{26}. Bản mã y** và bản rõ **x** $\in$ **Z_{26}.**

Tập khóa **K** là tập mọi hoán vị trên **Z_{26}.**

Với khóa **k = Л** $\in$ **K,** tức là một hoán vị trên **Z_{26},** ta định nghĩa :

Mã hóa : **$y = e_{Л}(x) = Л(x)$.**

Giải mã : **$x = d_{Л}(x) = Л^{-1}(y)$.**

2/. Ví dụ:

* Bản rõ chữ : **T O I N A Y T H A V I R U S**

* Chọn khóa **k = Л** là hoán vị:

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

* Mã hóa theo công thức: **$y = e_{Л}(x) = Л(x)$**

* Bản mã chữ: **E J P Z K Y V Z E Q Y Z C P G D F**

* Giải mã theo công thức : **$x = d_{Л}(x) = Л^{-1}(y)$** ta nhận lại được bản rõ chữ.

2.4.2. Dạng tấn công vào mã thay thế: Tìm cách xác định bản rõ

Do đặc điểm của mã thay thế là sự thay thế kí tự này bằng kí tự khác. Thám mã sẽ sử dụng phương pháp thống kê ngôn ngữ để xác định ra bản rõ từ bản mã.

Ví dụ với bản mã tiếng Anh:

Đầu tiên thám mã xác định tần suất xuất hiện của các kí tự trong bản mã. Sau đó dựa vào tần suất xuất hiện của từng kí tự, của các bộ đôi và bộ ba trong bản mã kết hợp với tần suất xuất hiện của các kí tự trong tiếng Anh, các bộ đôi và bộ ba thông dụng để đưa ra các giả thiết của sự thay thế. Từ đó xác định được bản rõ.

Tần suất xuất hiện của các chữ cái trong tiếng Anh:

E: có xác suất khoảng 0,127

T, A, O, I, N, S, H, R: mỗi kí tự có xác suất khoảng 0,06 – 0,09

D, L: mỗi kí tự có xác suất khoảng 0,04

C, U, M, W, F, G, Y, P, B: mỗi kí tự có xác suất khoảng 0,015 – 0,023.

V, K, J, X, Q, Z: mỗi kí tự có xác suất nhỏ hơn 0,01.

Việc xét các bộ đôi hoặc bộ ba cũng rất hữu ích.

30 bộ đôi thông dụng nhất theo thứ tự giảm dần là: TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, OF.

12 bộ ba thông dụng nhất theo thứ tự giảm dần là: THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, DTH.

Ví dụ: Thám mã có bản mã nhận được từ mã thay thế:

YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

NZUCDRJXYYSRMTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

Bảng tần số xuất hiện của 26 chữ cái trong bản mã.

Kí tự	Tần số	Kí tự	Tần số
A	0	N	9
B	1	O	0
C	15	P	1
D	13	Q	4
E	7	R	10
F	11	S	3
G	1	T	2
H	4	U	5
I	5	V	5
J	11	W	8
K	1	X	6
L	0	Y	10
M	16	Z	20

Do Z xuất hiện nhiều nhất trong bản mã nên thám mã có thể phóng đoán rằng $d_k(Z = e)$. C, D, F, M, R, Y mỗi kí tự xuất hiện ít nhất 10 lần. Thám mã phỏng đoán rằng chúng có thể là mã của các kí tự t, a, c, o, i, n, s, h, r nhưng chỉ dựa vào tần suất xuất hiện của từng chữ không đủ để có được phỏng đoán thích hợp. Lúc này thám mã sẽ xem xét đến các bộ đôi, đặc biệt là các bộ đôi có dạng –z hoặc z-. Các bộ đôi thường gặp nhất dạng này là DZ và WZ (4 lần mỗi bộ), NZ và ZU (mỗi bộ 3 lần). RZ, HZ, XZ, FZ, ZR, ZV, ZC, ZD, ZI (mỗi bộ 2 lần). Vì ZW xuất hiện 4 lần còn WZ không xuất hiện lần nào, nói chung W xuất hiện ít hơn so với nhiều kí tự khác nên phỏng đoán $d_k(W) = d$. Vì DZ xuất hiện 4 lần và ZD xuất hiện 2 lần nên thám mã phỏng đoán rằng $d_k(D) \in \{r, s, t\}$ nhưng chưa xác định chính xác kí tự nào.

Nếu tiến hành theo giả thiết $d_k(Z) = e$, $d_k(W) = d$ thì ta phải nhìn trở lại bản mã và thấy rằng cả hai bộ ba ZRW và RZW xuất hiện ở gần đầu của bản mã và RW xuất hiện lại sau đó. Vì nd là bộ đôi thường gặp, nên thử $d_k(R) = n$ có thể coi là phỏng đoán khả thi nhất.

YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

----------END----------E--------NED--------E----------

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

--------------E---------E----------N----D-----EN---------E---------E

NZUCDRJXYYSRMTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

--E------N---------N--------------ED-------E------E-----NE--ND---E--E--

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

--ED----------N----------E-------ED----------D------E---N

Tiếp theo thử $d_k(N) = h$ vì NZ là một bộ đôi thường gặp còn ZN không xuất hiện. Nếu điều giả sử này đúng thì gợi ý rằng $d_k(C) = a$ vì ne--ndhe. Bây giờ xét tới M là kí tự thường gặp nhất sau Z. Từ đoạn mã RNM ta đang giải mã thành nh--. Để ý rằng sau h là một nguyên âm. Ta đã sử dụng a và e nên phỏng đoán rằng $d_k(M) = i$ hoặc o. Vì ai là bộ đôi thường gặp hơn so với ao nên thử $d_k(M) = i$ thu được:

YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

----------IEND----------A--I---E-------I NED HI----E----------A---------

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

H---------I---EA---I---E--A----- A--I---NHAD--A-EN----A--E--HI----E

NZUCDRJXYYSRMTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

HE--A--N---------NI----I--------ED-------E------E----INEANDHE--E--

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

--ED---A-----INHI------HAI------A--E--I----ED---------A--D---HE---N

Tiếp theo thử xác định rằng chữ nào được mã thành o. Vì o là chữ thường gặp nên giả định rằng chữ cái tương ứng trong bản mã là một trong các kí tự D, F, J, Y. Trong đó Y được xem là thích hợp nhất. Vì thế giả thiết $d_k(Y) = o$. Ba kí tự thường gặp nhất còn lại trong bản mã: D, F, J ta phán đoán sẽ giải mã thành: r, s, t theo một thứ tự nào đó. Hai lần xuất hiện của bộ ba NMD cho kẻ thám mã phán đoán rằng $d_k(D) = s$ ứng với bộ ba his trong bản rõ.

Đoạn mã HNCMF có thể là bản mã của chair, điều này sẽ cho $d_k(F) = r$ và $d_k(H) = c$ và có $d_k(J) = t$. Từ đó xác định bản rõ hoàn chỉnh như sau:

Our friend from Paris examined his empty glass with surprise as if evaporation had taen place while he wasn't looking. I poured some more wine and he settle back in his chair face tilted up towards the sun.

→ Giải pháp phòng tránh:

Mở rộng vùng không gian khóa lớn. Ví dụ như bảng chữ cái tiếng Việt có thanh (gồm 94 ký tự). Số kí tự nhiều thì tần suất xuất hiện của các chữ cái, các bộ đôi và bộ ba cũng không khác biệt nhau nhiều lắm, do đó để phát hiện được kí tự "nổi bật" cũng khó khăn hơn.

2.5. TẤN CÔNG HỆ MÃ HÓA: AFFINE

2.5.1. Mã Affine

1/. Sơ đồ

Đặt **P = C = Z_{26}. Bản mã y** và bản rõ **x** $\in$ **Z_{26}.**

Tập khóa **K** = {(**a, b**), với **a, b** $\in$ **Z_{26},** UCLN(**a**, 26) = 1}

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

Phép mã hóa: **y = e_k(x) = (a.x + b)** mod 26.

Phép giải mã: **x = d_k(y) = a^{-1}(y - b)** mod 26.

2/. Ví dụ:

* Bản rõ chữ : CHIEUNAYOVUONHOA

* Chọn khóa : k = (a, b) = (3, 6).

* Bản rõ số: x = 2 7 8 4 20 13 0 24 14 21 20 14 13 7 14 0

Mã hóa theo công thức: **y = e_k(x) = (a.x + b)** mod 26 = (3x +6) mod 26

* Bản mã số: y = 12 1 4 18 14 19 6 0 22 17 14 22 19 1 22 6

* Bản mã chữ: MBESOTGAWROWTBWG

Giải mã theo công thức: **x = d_k(y) = a^{-1}(y - b)** mod 26

= **3^{-1}(y - 6)** mod 26 = **9 * (y - 6)** mod 26.

2.5.2. Dạng tấn công vào mã Affine: Tìm cách xác định khóa

Khóa mã Affine có dạng k = (a, b) với a, b$\in Z_{26}$ và gcd(a, 26) = 1. Ký tự mã y và kí tự bản rõ x tương ứng có quan hệ:

y = a.x + b mod 26.

Thám mã sử dụng phương pháp sắc xuất thống kê: dựa vào tần suất xuất hiện của các kí tự trong bản mã và tần suất xuất hiện của các kí tự trong tiếng Anh đưa ra các giả thiết. Từ đó biết được 2 cặp (x, y) khác nhau và có được hệ phương trình tuyến tính hai ẩn, giải hệ đó tìm ra giá trị a, b tức tìm ra khóa k. Kết hợp với có 12 số thuộc Z_{26} nguyên tố với 26 nên số khóa có thể có 12 x 26 = 312. Thám mã có thể sử dụng máy tính thử các trường hợp để tìm ra khóa thích hợp đúng nhất.

Ví dụ: Thám mã có bản mật mã:

FMXVEDKAPHFERBNDKRXRSREFMORUDSDKDVSHVUFEDKAPRKDLYEVLRHHRH

Bảng tần số xuất hiện của 26 chữ cái trong bản mã.

Kí tự	Tần số	Kí tự	Tần số
A	2	N	1
B	1	O	1
C	0	P	2
D	7	Q	0
E	5	R	8
F	4	S	3
G	0	T	0
H	5	U	2
I	0	V	4
J	0	W	0
K	5	X	2
L	2	Y	1
M	2	Z	0

Bản mã có 57 kí tự. Kí tự có tần suất cao nhất trong bản mã là R (8 lần), D (7 lần), E, H, K (mỗi kí tự xuất hiện 5 lần) và F, V (mỗi kí tự xuất hiện 4 lần).

+ Đầu tiên thám mã giả thiết rằng R là kí tự mã của chữ e và D là kí tự mã của chữ t vì e và t tương ứng là hai chữ cái thông dụng nhất. Biểu thị bằng số thám mã có: $e_k(4) = 17$ và $e_k(19) = 3$.

Từ đó có hệ phương trình tuyến tính hai ẩn:

$$\begin{cases} 4a + b = 17 \\ 19a + b = 3 \end{cases}$$

Giải ra được a = 6 , b = 19. Vì UCLN (a, 26) = 2 ≠ 1 nên (a, b) không thể là khóa được, giả thiết trên không đúng.

+ Thám mã lại giả sử: R là kí tự mã của e và E là kí tự mã của t. Làm tương tự như trên thì thu được a = 13.

Vì UCLN(13, 26) = 13 $\neq$ 1 nên giả thiết này cũng không hợp lệ.

+ Thám mã lại giả sử: R là kí tự mã của e và H là kí tự mã của t. Thực hiện tương tự nhận được a = 8 → không thỏa mãn điều kiện UCLN(a, 26) =1.

+ Tiếp theo thám mã lại giả thiết: R là kí tự mã của e và K là kí tự mã của t. Theo giả thiết này thám mã có hệ phương trình:

$$\begin{cases} 4a + b = 17 \\ 19a + b = 10 \end{cases}$$

giải ra được a = 3, b = 5. Vì UCLN(a, 26) = 1 nên k = (3,5) có thể là khóa cần tìm. Thám mã giải mã bản mã trên:

$d_k(y) = a^{-1}(y - b) \bmod 26$

Thay số :

9(5 - 5) mod 26 = 0 → kí tự rõ nhận được là a. Thực hiện tương tự thám mã sẽ nhận được bản rõ :

algorithmsarequitegeneraldefinitionsofarithmeticprocesses

Thám mã có thể kết luận khóa đúng là K = (3, 5) và dòng trên là bản rõ cần tìm.

→ Giải pháp phòng tránh:

Mở rộng vùng không gian khóa lớn. Ví dụ như bảng chữ cái tiếng Việt có thanh (gồm 94 ký tự). Số kí tự nhiều thì tần suất xuất hiện của các chữ cái cũng không khác biệt nhau nhiều lắm, do đó để phát hiện được kí tự "nổi bật" cũng khó khăn hơn. Và khi đó số khóa có thể có lớn hơn 312, việc thử tất cả các trường hợp sẽ lâu hơn.

KẾT LUẬN

Tìm hiểu, nghiên cứu qua các tài liệu để trình bày có hệ thống lại các vấn đề sau:

+ Các khái niệm trong mã hóa: mã hóa, tính năng của hệ mã hóa, các phương pháp mã hóa.

+ Một số bài toán trong mật mã: kiểm tra số nguyên tố lớn, phân tích thành thừa số nguyên tố, tính logarit rời rạc theo modulo.

+ Các phương pháp tấn công chủ yếu với hệ mã hiện nay bao gồm: KPA, COA, CPA, CCA1, CCA2.

+ Các khái niệm an toàn đối với một hệ mã hóa (an toàn một chiều, an toàn ngữ nghĩa, tính không phân biệt được, an toàn ngữ nghĩa tương đương với IND, IND-CCA).

+ Một số loại tấn công vào các hệ mã hóa (RSA, ELGAMAL, dịch chuyển, thay thế, AFFINE). Từ đó giúp ta có những giải pháp phòng tránh khi sử dụng để cho hệ mã an toàn hơn.

BẢNG CHỮ CÁI VIẾT TẮT

STT	Chữ viết tắt	Ý nghĩa
1	COA	Thám mã chỉ biết bản mã (Cyphertext only attack)
2	KPA	Thám mã biết bản rõ (Known plaintext attack)
3	CPA	Thám mã với bản mã được chọn (Chosen plaintext attack)
4	CCA	Thám mã với bản mã được chọn (Chosen ciphertext attack)
5	CCA1	Thám mã với bản mã được chọn trước bất kỳ (Non-adaptive chosen ciphertext attack)
6	CCA2	Thám mã với bản mã được chọn trước thích hợp (adaptive chosen ciphertext attack)
7	IND	Tính không phân biệt được (Indistinguishability)
8	Pr	Xác suất

TÀI LIỆU THAM KHẢO

[1]. Phan Đình Diệu – Lý thuyết mật mã và An toàn thông tin.

[2]. Kaoru Kurosiawa, Yvo Desmedt. A New Paradigm of Hybrid Encryption Scheme.2004.

[3]. Security Fundamentals for E-Commerce, Vesna Hasler, Pedrick Moore.

[4]. The Internatiional Handbook of computer Security, Jae K.Shim, Ph.D Anique A.Qureshi, Joel G.Giegel,... Glenlake Publishing Company, Ltd.

[5]. Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack.

[6]. Trịnh Nhật Tiến – Trường dh công nghệ – Giáo trình An toàn dữ liệu (2008).