HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐỖ DUY QUANG

XÂY DỰNG VÀ ĐÁNH GIÁ HỆ MẬT AFFINE- ELGAMAL TRÊN Zp

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI – 2019

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐỖ DUY QUANG

XÂY DỰNG VÀ ĐÁNH GIÁ HỆ MẬT AFFINE- ELGAMAL TRÊN Zp

CHUYÊN NGÀNH : HỆ THỐNG THÔNG TIN

MÃ SỐ: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: GS.TS. NGUYỄN BÌNH

HÀ NỘI - 2019

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất cứ công trình nào.

TÁC GIẢ

Đỗ Duy Quang

LỜI CÁM ƠN

Em xin gửi lời cảm ơn sâu sắc tới GS.TS. NGUYỄN BÌNH đã tận tình hướng dẫn, giúp đỡ và động viên em để hoàn thành tốt nhất Luận văn. Ngoài những kiến thức Thầy truyền đạt, em còn học được ở Thầy một phong cách làm việc rất khoa học, nghiêm túc và đầy trách nhiệm

Em xin chân thành cảm ơn toàn thể các thầy giáo, cô giáo Học viện Công nghệ Bưu chính Viễn thông đã dìu dắt, chỉ bảo tận tình cho em trong suốt thời gian học tập tại trường.

Mặc dù em đã cố gắng hoàn thành luận văn trong phạm vi và khả năng cho phép nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, kính mong nhận được góp ý của quý thầy cô và các bạn.

Trân trọng cám ơn !

Tác giả.

Đỗ Duy Quang

MỤC LỤC

DANH MỤC CÁC BẢNG BIỂU

DANH MỤC CÁC HÌNH VẼ

MỞ ĐẦU

1.1 Lí do chọn đề tài

Cùng với sự phát triển của công nghệ thông tin và truyền thông, mạng máy tính đang trở thành một phương tiện điều hành thiết yếu trong mọi lĩnh vực hoạt động của xã hội. Việc trao đổi thông tin và dữ liệu trong môi trường mạng ngày càng trở nên phổ biến và đang dần thay thế các phương thức truyền tin trực tiếp. Khi ngày càng nhiều thông tin được trao đổi thì nhu cầu về bảo mật thông tin là một vấn đề đặt ra cho nhiều ngành, lĩnh vực và nhiều quốc gia...Để bảo vệ các thông tin khỏi sự truy cập trái phép cần phải kiểm soát được những vấn đề như: *thông tin được tạo ra, lưu trữ và truy nhập như thế nào, ở đâu, bởi ai và vào thời điểm nào.* Giải quyết các vấn đề trên, kỹ thuật mật mã hiện đại phải đảm bảo các dịch vụ an toàn cơ bản: (1) bí mật (Confidential); (2) xác thực (Authentication); (3) đảm bảo tính toàn vẹn (Integrity).

Hệ mật mã ra đời nhằm đảm bảo các dịch vụ an toàn cơ bản trên như: hệ mật mã với khóa sở hữu riêng (Private Key Cryptosystems),hệ mã với khóa bí mật (Secret Key Cryptosystems), hệ mã truyền thống (Conventional Cryptosystems) đều là những hệ mật mã sử dụng mã khóa đối xứng; hệ mật mã với khóa công khai. Hệ mật mã với khóa công khai cho phép người sử dụng trao đổi các thông tin mật mà không cần phải trao đổi các khóa chung bí mật trước đó; mật mã hóa khóa công khai được thiết kế sao cho khóa sử dụng trong quá trình mã hóa khác biệt với khóa sử dụng trong quá trình giải mã; khóa sử dụng dùng để mã hóa và ngược lại, tức là hai khóa này có quan hệ với nhau về mặt toán học nhưng không thể suy diễn được ra nhau. Một trong những thuật toán mã khóa công khai được phát triển dựa trên Hệ mật mã ElGamal cho phép giải quyết tốt các yêu cầu bảo mật thông tin thực hiện đồng thời việc xác thực

về nguồn gốc và tính toàn vẹn của thông tin.Luận văn sẽ trình bày về hệ mật mã kết hợp mã Affine và hệ mật mã ElGamal.

1.2 Mục tiêu nghiên cứu

Mục tiêu nghiên cứu: Tìm hiểu hoạt động của hệ mật mã khóa công khai sử dụng biến thể thuật toán ElGamal: Hệ mật mã Affine –ElGamal. Đánh giá tính bảo mật thông tin, xác thực về nguồn gốc thông tin, xác thực về tính toàn vẹn của thông tin của hệ thống

1.3 Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu :

- Tìm hiểu hệ mật mã Affine –ElGamal.
- Xây dựng hệ mật biến thể Affine –ElGamal sử dụng Diffic- Hellman.

Phạm vi nghiên cứu : đề tài nghiên cứu và đánh giá hiệu quả tính an toàn của hệ mật Affine –ElGamal.

1.4 Phương pháp nghiên cứu

Phương pháp nghiên cứu

* Phương pháp lý thuyết

- Tìm hiểu nghiên cứu về mật mã, cơ sở toán học của hệ mật mã

- Tìm hiểu bài toán logarithm rời rạc và hệ mật ElGamal; thủ tục trao đổi khóa Diffic- Hellman; các phương pháp che giấu dữ liệu và các điều kiện lũy đẳng và giao hoán của các hệ mật

- Lý thuyết chung về hệ mật Affine từ đó xây dựng biến thể của hệ mật Affine- ElGamal.

* Phương pháp thực nghiệm

- Xây dựng hệ mật áp dụng giải thuật Affine- ElGamal

- Đánh giá hiệu quả và tính an toàn của Hệ mật Affine- ElGamal.

1.5 Cấu trúc luận văn

Chương 1 : Bài toán lôgarith rời rạc

Chương 2: Xây dựng hệ mật Affine – ElGamal

Chương 3: Đánh giá hệ mật mã Affine- ElGamal

CHƯƠNG 1: BÀI TOÁN LÔGARIT RỜI RẠC

1.1 Tổng quan mật mã học

Mật mã hóa khóa công khai là một dạng mật mã hóa cho phép người sử dụng trao đổi các thông tin mật mà không cần phải trao đổi các khóa chung bí mật trước đó, được thực hiện bằng cách sử dụng một cặp khóa có quan hệ toán học với nhau là khóa công khai và khóa cá nhân (hay khóa bí mật). Trong mật mã hóa khóa công khai, khóa cá nhân phải được giữ bí mật trong khi khóa công khai được phổ biến công khai. Trong 2 khóa, một dùng để mã hóa và khóa còn lại dùng để giải mã. Điều quan trọng đối với hệ thống là không thể tìm ra khóa bí mật nếu chỉ biết khóa công khai. Hệ thống mật mã hóa khóa công khai có thể sử dụng với các mục đích: Mã hóa; Tạo chữ ký số; Thỏa thuận khóa, cho phép thiết lập khóa dùng để trao đổi thông tin mật giữa 2 bên. Các kỹ thuật mật mã hóa khóa công khai đòi hỏi khối lượng tính toán nhiều hơn các kỹ thuật mã hóa khóa đối xứng nhưng có nhiều ưu điểm nên được áp dụng trong nhiều ứng dụng.

Hệ mật mã được định nghĩa là một bộ năm thành phần (P, C, K, E, D), thỏa mãn các tính chất sau:

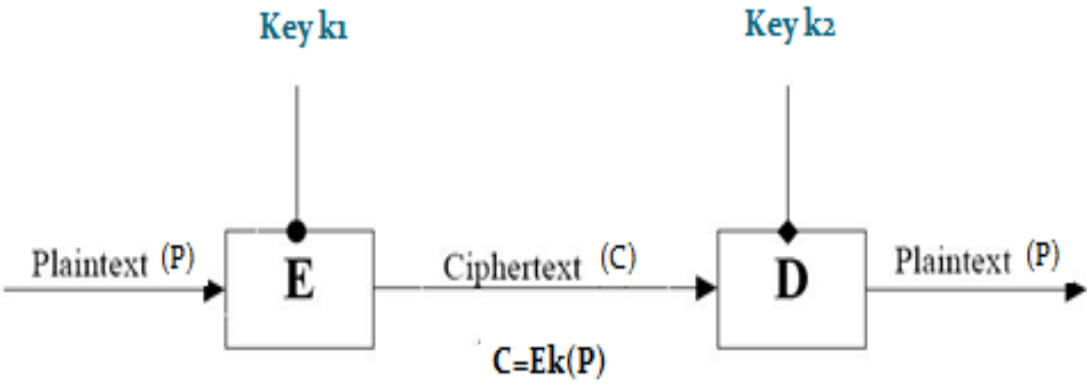
- **P** (Plaintext) : Tập hợp hữu hạn các bản rõ có thể chưa được mã hóa
- **C** (Ciphertext): Tập hợp hữu hạn các bản rõ có thể đã mã hóa.
- **K** (Key): Tập hợp các bản khoá mã hóa, khóa giải mã có thể.
- **E** (Encrytion): Tập hợp các qui tắc mã hoá có thể.
- **D** (Decrytion): Tập hợp các qui tắc giải mã có thể.

Quá trình mã hóa được tiến hành bằng cách áp dụng hàm toán học E lên thông tin P, vốn được biểu diễn dưới dạng số, để trở thành thông tin đã mã hóa C. Đối với mỗi $k \in K$ có một quy tắc mã $e_k \in E$

e_k: P → C

Quá trình giải mã được tiến hành ngược lại: áp dụng hàm D lên thông tin C để được thông tin đã giải mã. Một quy tắc giải mã tương ứng $d_k \in D$

d_k : C → P sao cho $d_k(e_k( )) x = x$ với $\forall x \in P$.



Hình 1.1: Quá trình mã hoá và giải mã

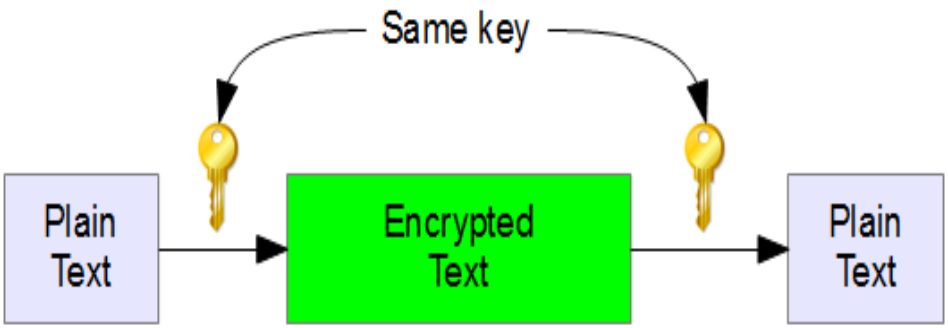
- **Thám mã** (phá mã): Là tìm những điểm yếu hoặc không an toàn trong phương thức mật mã hóa. Thám mã có thể được thực hiện bởi những kẻ tấn công, nhằm làm hỏng hệ thống; hoặc bởi những người thiết kế ra hệ thống (hoặc những người khác) với ý định đánh giá độ an toàn của hệ thống. Thám mã tuyến tính và Thám mã vi phân là các phương pháp chung cho mật mã hóa khóa đối xứng. Khi mật mã hóa dựa vào các vấn đề toán học như độ khó NP, giống như trong trường hợp của thuật toán khóa bất đối xứng, các thuật toán như phân tích ra thừa số nguyên tố trở thành công cụ tiềm năng cho thám mã.

Phân loại hệ mật :

- Hệ mật mã bí mật (hay còn gọi là mật mã đối xứng): Là những hệ mật mà quá trình mã hóa và giải mã một thông điệp sử dụng cùng một khóa gọi là khóa bí mật (secret key) hay khóa đối xứng (symmetric key). Do đó, vấn đề bảo mật thông tin đã mã hóa hoàn toàn phụ thuộc vào việc giữ bí mật nội dung

của mã khóa đã được sử dụng. Một số thuật toán nổi tiếng trong mã hoá đối xứng là: DES, Triple DES(3DES), RC4, AES…

Thông điệp nguồn được mã hóa với mã khóa k được thống nhất trước giữa người gửi A và người nhận B. Người A sẽ sử dụng mã khóa k để mã hóa thông điệp x thành thông điệp y và gửi y cho người B người B sẽ sử dụng mã khóa k để giải mã thông điệp y này. Vấn đề an toàn bảo mật thông tin được mã hóa phụ thuộc vào việc giữ bí mật nội dung mã khóa k. Nếu người C biết được mã khóa k thì C có thể "mở khóa" thông điệp đã được mã hóa mà người A gửi cho người B.



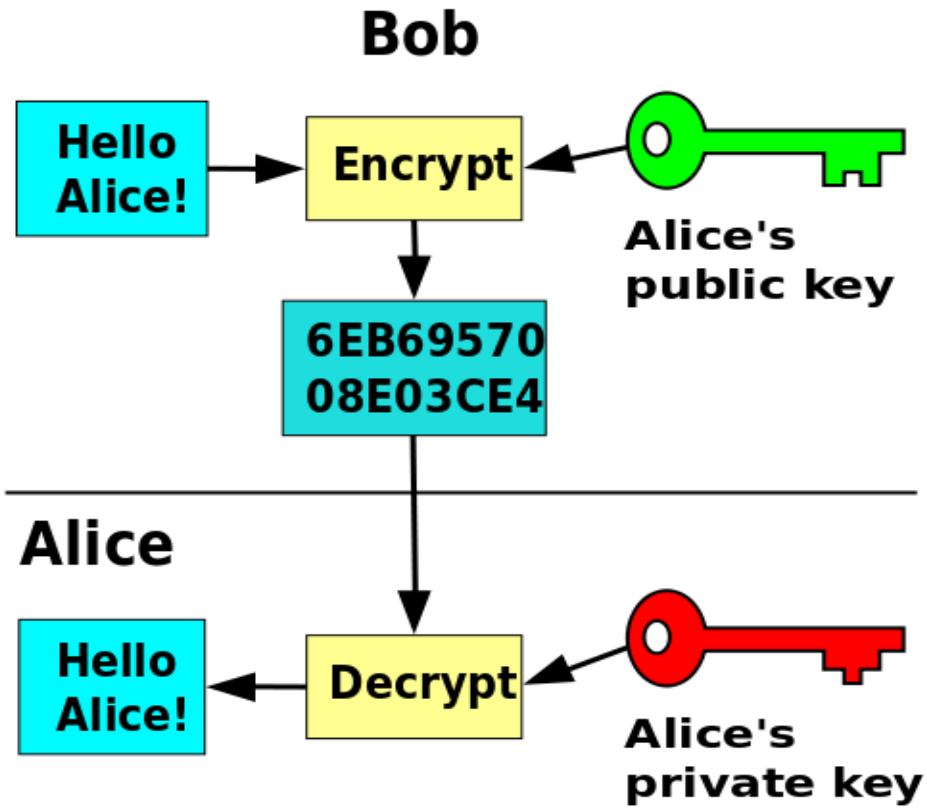
Hình 1.2: Mã hoá Sử dụng khóa công khai P

- Hệ mật mã khóa công khai (hay còn gọi là mật mã bất đối xứng): Nếu như vấn đề khó khăn đặt ra đối với các phương pháp mã hóa cổ điển chính là bài toán trao đổi mã khóa thì ngược lại, các phương pháp mã hóa khóa công cộng giúp cho việc trao đổi mã khóa trở nên dễ dàng hơn. Nội dung của khóa công cộng (public key) không cần phải giữ bí mật như đối với khóa bí mật trong các phương pháp mã hóa quy ước. Sử dụng khóa công cộng, chúng ta có thể thiết lập một quy trình an toàn để truy đổi khóa bí mật được sử dụng trong hệ thống mã hóa quy ước.

Các hệ mật này dùng một khoá để mã hoá sau đó dùng một khoá khác để giải mã, nghĩa là khoá để mã hoá và giải mã là khác nhau. Các khoá này tạo nên từng cặp chuyển đổi ngược nhau và không có khoá nào có thể suy được từ khoá kia. Khoá dùng để mã hoá có thể công khai nhưng khoá dùng để giải mã phải giữ bí mật. Do đó trong thuật toán này có 2 loại khoá: Khoá để mã hoá được gọi là khóa công khai (Public Key), khoá để giải mã được gọi là khóa bí mật (Private Key). Một số thuật toán mã hoá công khai nổi tiếng: Diffle-Hellman, RSA, Rabin, ElGamal,…

Trong mô hình mật mã cổ điển mà cho tới nay vẫn còn đang được nghiên cứu Alice (người gửi) và Bob(người nhận) bằng cách chọn một khóa bí mật K. Sau đó Alice dùng khóa K để mã hóa theo luật e_k và Bod dùng khóa K đó để giải mã theo luật giải d_k. Trong hệ mật này, d_k hoặc e_k hoặc dễ dàng nhận được từ nó vì quá trình giải mã hoàn toàn tương tự như quá trình mã hóa, nhưng thủ tục khóa thì ngược lại. Nhược điểm lớn của hệ mật này là nếu ta để lộ e_k thì làm cho hệ thống mất an toàn, chính vì vậy chúng ta phải tạo cho các hệ mật này một kênh an toàn mà kinh phí để tạo kênh an toàn không rẻ. Ý tưởng xây dựng một hệ mật khóa công khai là tìm một hệ mật có khả năng tính toán để xác định d_k nếu biết được e_k. Nếu thực hiện được như vậy thì quy tắc mã e_k có thể được công khai bằng cách công bố nó trong danh bạ, và khi Alice (người gửi) hoặc bất cứ một ai đó muốn gửi một bản tin cho Bob (người nhận) thì người đó không phải thông tin trước với Bob (người nhận) về khóa mật, mà người gửi sẽ mã hóa bản tin bằng cách dùng luật mã công khai e_k. Khi bản tin này được chuyển cho Bob (người nhận) thì chỉ có duy nhất Bob mới có thể giải được bản tin này bằng cách sử dụng luật giải mã bí mật d_k.

Quá trình này được mô tả trong hình



Hình 1.3: Mã hoá và giải mã thông điệp sử dụng khóa riêng của người nhận

- Hệ ElGamal là một hệ mật mã công khai.

- Hệ ElGamal dựa trên bài toán logarithm rời rạc. Tính an toàn của nó phụ thuộc vào độ phức tạp của bài toán logarithm.

- Hệ ElGamal là 1 biến thể của sơ đồ phân phối khóa Diffie – Hellmal, được đưa ra năm 1984.

- So với RSA, hệ ElGamal không có nhiều rắc rối vấn đề bản quyền sử dụng.

1.2 Giới thiệu bài toán Lôgarit rời rạc

Bài toán Lôgarith rời rạc là sự kết nối của phép tính lôgarith trên trường số thực vào các nhóm hữu hạn. Với hai số thực x,y và cơ số a>0, a#0, nếu a^x – y =0 thì x được gọi là logarith cơ số a của y, ký hiệu x = $\log_a y$. Logarith rời rạc có ứng dụng trong hệ mật mã khóa công khai Hệ mật mã Elgamal.

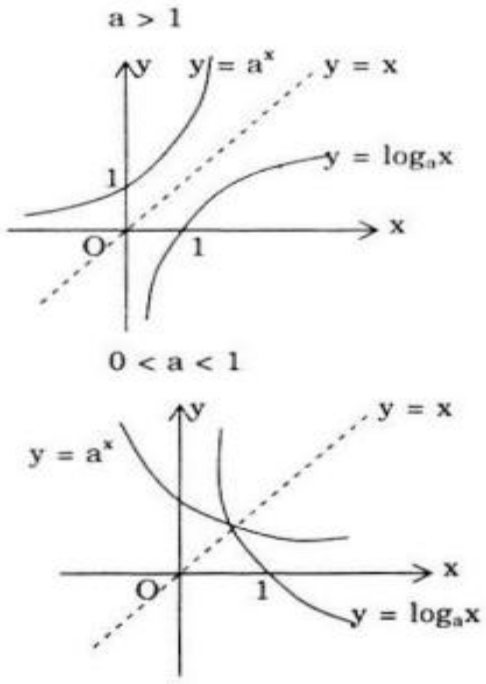
Bài toán lôgarith rời rạc là bài toán khó. Trong khi bài toán ngược lũy thừa rời rạc lại không khó (có thể sử dụng thuật toán bình phương và nhân).

1.2.1 Bài toán Lôgarit trên trường số thực R:

Một số tính chất của hàm logarit.

1. $log_a 1=0;\ log_a a=1$
2. $log_a a^n=n;\ a^{log_a n} = n$
3. $log_a(b.c)=log_a b+log_a c$
4. $log_a \left(\frac{b}{c}\right) = log_a b - log_a c$
5. $log_a b^n = n log_a |b|$
6. $log_{a^n} b = \frac{1}{n} log_{|a|} b$
7. $log_a b = \frac{1}{log_b a}$
8. $log_a b=log_a c.log_c b$
9. $log_a b = \frac{log_c b}{log_c a}$

- **Bài toán thuận:** Hàm số $y = a^x$ với a,x ∈ R việc tính toán hàm mũ này có thể được thực hiện dễ dàng bằng thuật toán nhân và bình phương.
- **Bài toán ngược**: phép tính ngược của hàm mũ chính là hàm logarit $y = \log_a x$, việc tính toán hàm ngược logarit này khó khăn hơn nhiều so với hàm thuận. Tuy nhiên, cả hai phép mũ và logarit đều là các hàm đồng biến cho nên có thể xác định giá trị tương đối của hàm logarit như hình dưới đây



Hình 1.4: Đồ thị hàm số $y=a^x$ và $y = \log_a x$

1.2.2 Bài toán Lôgarit trên trường hữu hạn:

Xét với vành đa thức $\mathbf{Z}_p^*$ với p là số nguyên tố thì theo định lý nếu p là nguyên tố thì ¢ *p* là một trường (¢ *p* = GF(2)).

Tập tất cả các phần tử khác không của trường sẽ tạo nên một nhóm nhân xyclic ¢ **p* .

$¢^*_p = ¢_p / \{0\} = \{1,2,...,p-1\}$

- *Bài toán thuận:* $\boldsymbol{y = a^x \bmod p, (a,x \in ¢^*_p)}$

Ví dụ: Xét *p = 19,a = 2* ta có các giá trị $y = a^x$ như trong bảng dưới đây

Bảng 1.1: Các giá trị của $y = 2^x \bmod 19$ trên $¢^*_{19}$

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2^x	2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1

Chú ý:

+ Nếu a là một phần tử nguyên thủy thì a^x sẽ đi qua tất cả các phần tử của nhóm.

+ Nếu a là phần tử nguyên thủy thì a^i cũng là nguyên thủy với (i,p-1)= 1(p là số nguyên tố).

Trong ví dụ trên các giá trị của i thỏa mãn (i,18)=1 là i = (1,5,7,11,13,17). Số lượng các giá trị của *i bằng giá trị hàm* φ(p –1).

$$N_i = \varphi(p-1) = \varphi(18) = 6$$

Cách tính hàm Phi-Euler φ như sau:

φ(1)= 1, và φ(n) = (p-1)p^{k-1} với n là lũy thừa bậc k của số nguyên tố p; nếu m và n là hai số nguyên tố cùng nhau thì φ(mn)= φ(m).φ(n)

Nếu n = $p_1^{k_1}$ ..... $p_r^{k_r}$ trong đó các p_j là các số nguyên tố phân biệt thì

φ(n) = (p_1- 1)$p_1^{k_1-1}$…….(p_1- 1)$p_r^{k_r-1}$

$$\varphi(n) = n \prod_{p|n} (1 - \frac{1}{p})$$

Như vậy trong nhóm $¢^*_{19}$ có 6 phần tử nguyên thủy:

2 = 2^i ; 13 = 2^5 ; 14 = 2^7 ; 15 = 2^{11}; 3 = 2^{13}; 10 = 2^{17}

Các phần tử nguyên thủy này tạo thành các cặp nghịch đảo như sau:

(2,10) ↔ 2 = 10^{-1}

(13,3) ↔ 13 = 3^{-1}

(14,15) ↔ 14 = 15^{-1}

+ Bài toán ngược: ***y = $log_a x$, (a,x ∈ $¢^*_p$)***

Từ bảng trên ta tính được giá trị hàm $log_2 x$ như sau

Bảng 1.2 . Giá trị $\log_2 x \pmod{19}$ trên $¢^*_{19}$

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2^x	2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
$\log_2 x$	18	1	13	2	16	14	6	3	8	17	12	15	5	7	11	4	10	9

Vì $2^{18} = 1$ nên $\log_2 1 = 18$

Một số tính chất của hàm lôgarit rời rạc a^{-1}

+ $y = log_a bc = (log_a b + log_a c) \bmod p - 1$

+ $y = log_a \frac{b}{c} = (log_a b - log_a c) \bmod p-1$

+ $log_a{}^{-1} x = - log_a x = p - 1 - log_a x$

+ $log_a 1 = 0 = p - 1$ *(coi $0 = p - 1$)*

Nhận xét: Từ hai bảng trên ta thấy hai hàm thuận và ngược đều không phải là hàm đồng biến, khi biết bài toán thuận thì mới tìm được bài toán ngược. Do đó việc giải bài toán ngược giống bài toán vét cạn, phải thử lần lượt các trường hợp.

Việc xác định logarit của một phần tử bất kỳ trong trường hợp là bài toán khó giải.

- ***Bài toán thuận:***

  Cho $¢^*_p$ với p là số nguyên tố, α là một phần tử nguyên thủy $\alpha \in ¢^*_p$.

  Yêu cầu tìm $y = log_a x$ với $\alpha, x \in ¢^*_p$.

  Nhận xét: $\forall x \in ¢^*_p$ *thì*

- Bài toán có nghiệm khi α là phần tử nguyên thủy.
- Bài toán có thể không có nghiệm khi α là phần tử bất kỳ.

Ví dụ: Với trường hợp *p = 19* ta đã tính được 6 phần tử nguyên thủy như trong hình 1.2 ta sẽ đi tìm bài toán lôgarith rời rạc với cơ số 6 phần tử nguyên thủy này.

Tuy nhiên ta có thể áp dụng tính chất của hàm lôgarith rời rạc để tính lôgarith với cơ số là các cặp số nghịch đảo.

$$log_a^{-1}x = - log_a x = p - 1 - log_a x, \text{ hay } log_a^{-1}x + log_a x = p - 1$$

Tức là (2,10) là cặp số nghịch đảo, khi đó $\log_{10}x = p - 1 - log_2 x = 18 - log_2 x$. *Tương tự (13,3) và (14,15)* là các cặp nghịch đảo nên $log_3 x = 18 - log_{13} x$ và $log_{15} x = 18 - log_{14} x$

Với quy tắc như thế có thể tính được các giá trị lôgarit như trong bảng 1.3

Bảng 1.3. Bài toán lôgarit rời rạc trên ¢*$_{19}$

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2^x	2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
$\log_2 x$	18	1	13	2	16	14	6	3	8	17	12	15	5	7	11	4	10	9
$\log_{10} x$	18	17	5	16	2	4	12	15	10	1	6	3	13	11	7	14	8	9
13^x	13	17	12	4	14	11	10	16	18	6	2	7	15	5	8	9	3	1
$\log_{13} x$	18	11	17	4	14	10	12	15	16	7	6	3	1	5	13	8	2	9
$\log_3 x$	18	7	1	14	4	8	6	3	2	11	12	15	17	13	5	10	16	9
14^x	14	6	8	17	10	7	3	4	18	5	13	11	2	9	12	16	15	1
$\log_{14} x$	18	13	7	8	10	2	6	3	14	5	12	15	11	1	17	16	4	9
$\log_{15} x$	18	5	11	10	8	16	12	15	4	13	6	3	7	17	1	2	14	9

Có thể tính 13^x thông qua 2^x , ta thấy $13 = 2^5$do đó $13^x = 2^{5x}$, tương tự như thế có thể tính $14^x = 2^{7x}$.

1.2.3 Thuật toán lôgarit rời rạc

1.2.3.1 Mở đầu

Phương pháp đơn định

Cho G là nhóm nhân Abel, $a, b \in G$. Bài toán tìm kiếm nghiệm của phương trình

$$a^x = b$$

gọi là bài toán lôgarith rời rạc trong nhóm G. Nghiệm x của phương trình gọi là lôgarith rời rạc cơ số a của b, ký hiệu là $\log_a b$, nếu như cơ số a cố định và nếu như nghiệm của phương trình tồn tại; $\log_a b \in Z_{|G|}$, nếu như $|G| < \infty$.

Bài toán lôgarit rời rạc có vai trò rất lớn trong ứng dụng của mật mã. Đặc biệt quan trọng trong trường hợp $G = F(q)^*$, với $q = p^l$, p là số nguyên tố, $l \in N$, tức là trong trường Galois, cũng như trong trường hợp G là một nhóm điểm của đường cong Elliptic trong trường hữu hạn.

Chúng ta xem phương trình

$$a^x \equiv b (\mathrm{mod}\, p) \quad (3.1)$$

trong nhóm Z_p^*, với p là số nguyên tố. Chúng ta giả sử rằng bậc của $a(\mathrm{mod}\, p)$ bằng p-1. Khi đó phương trình giải được, và nghiệm x là một phần tử của Z_{p-1} . Trong phần này chúng ta miêu tả phương pháp đơn định để xác định nghiệm của (3.1).

Nếu với sự giúp đỡ của phương pháp chọn thì có thể giải phương trình (3.1) cần $O(p)$ lệnh số học.

Nghiệm $\log_a b$ của phương trình (3.1) có thể tìm theo công thức sau

$$\log_a b \equiv \sum (1-a^j)^{-1} b^j \pmod{p-1},$$

thế nhưng độ phức tạp nếu tính theo công thức này thi sẽ tồi hơn cách lựa chọn.

Thuật toán tiếp theo giải phương trình (3.1) có độ phức tạp là $O(p^{1/2}\log p)$ lệnh số học.

Thuật toán tương hợp.

Bước 1. Gán $H := \lfloor p^{1/2} \rfloor + 1$.

Bước 2. Tìm $c \equiv a^H \pmod{p}$.

Bước 3. Lập bảng giá trị $c^u \pmod{p}, 1 \le u \le H$, sắp xếp nó.

Bước 4. Lập bảng giá trị $b.a^v \pmod{p}, 0 \le v \le H$, sắp xếp nó.

Bước 5. Tìm sự trùng nhau phần tử từ bảng thứ nhất và bảng thứ hai. Để làm điều này

$$c^u \equiv b.a^v \pmod{p},$$

từ đây $$a^{Hu-v} \equiv b \pmod{p}.$$

Bước 6. Đưa ra giá trị $x \equiv Hu - v \pmod{p-1}$.

Kết thúc thuật toán.

Chúng ta chứng minh sự đúng đắn của thuật toán. Bất kỳ số nguyên x, $0 \le x \le p-2$, có thể biểu diễn dưới dạng $x \equiv Hu - v \pmod{p-1}$,

Ở đây $1 \le u \le H, 0 \le v \le H$, rõ ràng rằng tập số H,H-1,H-2,…,H-H,2H,2H-1,…, $H^2, H^2-1, \ldots, H^2-H$ chứa trong mình tập số 0,1,…,p-2, bởi vì $H^2 > p$.

Từ đây dẫn đến sự đúng đắn của thuật toán. Đánh giá độ phức tạp của thuật toán cũng rõ ràng đúng, bởi vì tập từ N phần tử có thể sắp xếp cần $O(N \log N)$ lệnh số học.

Phương pháp ρ - Pollaid đối với Bài toán lôgarith rời rạc

Bài toán logarithm rời rạc theo modulo là số nguyên tố p. Chúng ta muốn giải phương trình

$$a^x \equiv b (\mathrm{mod}\ p).$$

Để làm việc này chúng ta xem 3 dãy số

$$\{u_i\}, \{v_i\}, \{z_i\},\ i = 0,1,2,...,$$

Được xác định như sau:

$u_0 = v_0 = 0,\ z_0 = 1,$

$u_{i+1} \equiv u_i + 1 (\mathrm{mod}\ p-1)$, nếu như $0 < z_i < p/3$;

$u_{i+1} \equiv 2u_i (\mathrm{mod}\ p-1)$, nếu như $p/3 < z_i < 2/3p$;

$u_{i+1} \equiv u_i (\mathrm{mod}\ p-1)$, nếu như $2/3p < z_i < p$;

$v_{i+1} \equiv v_i (\mathrm{mod}\ p-1)$, nếu như $0 < z_i < p/3$;

$v_{i+1} \equiv 2v_i (\mathrm{mod}\ p-1)$, nếu như $p/3 < z_i < 2/3p$;

$v_{i+1} \equiv v_i + 1 (\mathrm{mod}\ p-1)$, nếu như $2/3p < z_i < p$;

$$z_{i+1} \equiv b^{u_{i+1}} a^{v_{i+1}} (\mathrm{mod}\ p-1).$$

Tiếp theo chúng ta xem tập hợp $(z_i, u_i, v_i, z_{2i}, u_{2i}, v_{2i})$, $i = 1,2,3,...,$ chúng ta tìm vị trí i, sao cho $z_i = z_{2i}$. Từ đẳng thức cuối cùng ta rút ra

$$b^{u_{2i}-u_i} \equiv a^{v_i-v_{2i}} (\text{mod } p).$$

Nếu như $\gcd(u_{2i}-u_i, p-1)=1$, thì khi $l \in Z, l(u_{2i}-u_i) \equiv 1(\text{mod } p-1)$ chúng ta thu được

$$b \equiv a^{l(v_i-v_{2i})} (\text{mod } p),$$

từ đây giá trị x cần tìm bằng $\log_a b \equiv l(v_i-v_{2i})(\text{mod } p-1)$.

Chú thích: Thuật toán

1.2.3.2 Thuật toán Pohlig-Hellman

Bây giờ giả sử chúng ta biết được sự phân tích thành nhân tử của p-1 ra thừa số

$$p-1=\prod_{i=1}^{s} q_i^{\alpha_i}$$

Lúc này phương trình (5.1) có thể giải cần $O\left(\sum_{i=1}^{s} \alpha_i(\log p + q_i)\right)$ lệnh số học với sự giúp đỡ của thuật toán sau.

Thuật toán Pohlig-Hellman

Bản chất của thuật toán nằm ở chổ, tìm số lượng đủ lớn phương trình x theo modulo $q_i^{\alpha_i}$ với tất cả i, sau đó tìm nghiệm của phương trình ban đầu bằng định lý phần dư trung hoa. Để tìm x theo một trong các modulo như thế, chúng ta phải giải đồng dư thức

$$(a^{\frac{p-1}{q_i^{\alpha_i}}})^x \equiv (a^{\frac{p-1}{q_i^{\alpha_i}}})(\text{mod } p)$$

Phương trình này giải được với độ phức tạp thời gian là đa thức trong trường hợp nếu như q_i không quá lớn (có nghĩa là không vượt qúa $(\log p)^c$, c là một hằng số nào đó).

Bước 1. Đối với từng số nguyên tố $q, q \mid p-1$, ta lập bảng giá trị

$$r_{q,j} \equiv a^{j(p-1)/q} (\mathrm{mod}\, p),\ \ j = 0,...,q-1.$$

Bước 2. Đối với từng số nguyên tố q, $q^\alpha \parallel p-1$, chúng ta tìm $\log_a b (\mathrm{mod}\, q^\alpha)$.

Đặt

$x \equiv \log_a b (\mathrm{mod}\, q^\alpha) \equiv x_0 + x_1 q + ... + x_{\alpha-1} q^{\alpha-1} (\mathrm{mod}\, q^\alpha)$, với $0 \le x_i \le q-1$. Lúc này từ (5.1) dẫn đến rằng

$$b^{(p-1)/q} \equiv a^{x_0 (p-1)/q} (\mathrm{mod}\, p).$$

Với sự giúp đỡ của bảng trong bước 1 chúng ta tìm ra x_0.Lúc này rõ ràng ta có

$$(ba^{-x_0})^{(p-1)/q^2} \equiv a^{x_1 (p-1)/q} (\mathrm{mod}\, p).$$

Theo bảng trong bước 1 ta tìm ra giá trị của x_1 và tiếp tục như thế. Giá trị của x_i được tìm thấy từ phương trình

$$(ba^{-x_0 - x_1 q - ... - x_{i-1} q^{i-1}})^{(p-1)/q^{i+1}} \equiv a^{x_i (p-1)/q} (\mathrm{mod}\, p).$$

Bước 3. Khi tìm $\log_a b (\mathrm{mod}\, q_i^{\alpha_i}), i = 1,...,s$, chúng ta tìm $\log_a b (\mathrm{mod}\, p-1)$ theo định lý phần dư trung hoa.

Kết thúc thuật toán

Chúng ta chứng minh đánh giá độ phức tạp của thuật toán. Tập phần tử $a^{(p-1)/q_i} (\mathrm{mod}\, p)$ cần $\sum_{i=1}^{s} O(\log p)$ lệnh số học. Sau đó tập $r_{q_i, j}$ đối với tất cả q_i, j được tính toán cần $\sum_{i=1}^{s} O(q_i)$ lệnh số học. Để tìm giá trị x_i trong bước 3 cần nâng bậc(có nghĩa tìm $a^{x_{i-1} q^{i-1}}$), tìm phần tử nghịch đảo,nhân, nâng bậc và tiến hành theo bảng. Tất cả kết hợp lại là độ phức tạp của thuật toán được nêu ở trên.

Chú ý. Thuật toán Polug-Hellman có độ phức tạp là đa thức $O((\log p)^{c_1})$ trong trường hợp khi tất cả các ước nguyên tố q_i của p không vượt quá $(\log p)^{c_2}$, ở đây c_1, c_2 hằng số dương.

1.2.3.3 Thuật toán Adleman

Tầng 1. Hình thành cơ sở nhân tử, bao gồm tất cả các số nguyên tố q, $q \le B = e^{const\sqrt{\log p \log\log p}}$

Tầng 2. Bằng cách chọn lựa chúng ta tìm số tự nhiên r_i sao cho

$$a^{r_i} \equiv \prod_{q \le B} q^{\alpha_{iq}} (\mathrm{mod}\, p),$$ q là số nguyên tố

Từ đây dẫn đến

$$r_i \equiv \sum_{q \le B} \alpha_{iq} \log_a q (\mathrm{mod}\, p-1).$$ (3.3), q là số nguyên tố

Tầng 3. Chọn số lượng đủ lớn biểu thức (3.3), giải hệ phương trình tuyến tính thu được ứng với các ẩn $\log_a q$ -logarith rời rạc của phần tử của cơ sở nhân tử.

Tầng 4. Bằng cách lựa chọn chúng ta tìm ra một giá trị của r, sao cho

$$a^r \cdot b \equiv \prod_{q \le B} q^{\beta_q} \cdot p_1 ... p_k (\mathrm{mod}\, p),$$

ở đây $p_1, ..., p_k$ - là các số nguyên tố với độ lớn "trung bình", có nghĩa $B < p_i < B_1$, với $B_1 = e^{const\sqrt{\log p \log\log p}}$

Tầng 5. Bằng cách tính toán tương tự như tầng 2 và 3 của thuật toán, tìm ra logarithm rời rạc $\log_a p_i$ đối với các số nguyên tố $p_1, ..., p_k$ ở tầng 4.

Tầng 6. Xác định giá trị cần tìm $\log_a b$:

$$\log_a b \equiv -r + \sum_{q \le B} \beta_q \log_a q + \sum_{i=1}^{k} \log_a p_i (\mathrm{mod}\, p-1).$$

Kết thúc thuật toán.

1.2.3.4 Thuật toán COS

Tầng 1. Đặt

$$H := \left[P^{1/2}\right] + 1, J := H^2 - p > 0, L = e^{\sqrt{\log p \log\log p}}, 0 < \varepsilon < 1$$

Hình thành tập hợp

$$\left\{q \mid q < L^{1/2}\right\} \cup \left\{H + c \mid 0 < c < L^{1/2+\varepsilon}\right\},$$

q là số nguyên tố.

Tầng 2. Bằng cách sàng chúng ta tìm cặp c_1, c_2 sao cho $0 < c_i < L^{1/2+\varepsilon}$, i=1,2

$$(H + c_1)(H + c_2) \equiv \prod_{q \le L^{1/2}} q^{\alpha_q(c_1, c_2)} (\text{mod } p)$$

Trong trường hợp này, bởi vì $J = O(p^{1/2})$ nên

$$(H + c_1)(H + c_2) \equiv J + (c_1 + c_2)H + c_1 c_2 (\text{mod } p)$$

Lôgarit theo cơ số a chúng ta thu được biểu thức sau

$$\log_a(H + c_1) + \log_a(H + c_2) \equiv \sum_{q \le L^{1/2}} \alpha_q(c_1, c_2) \log_a q (\text{mod } p - 1)$$

a có thể tính theo công thức

$$a \equiv \prod_{q \le L^{1/2}} q^{\beta_q} (\text{mod } p)$$

Từ đây

$$1 \equiv \sum_{q \le L^{1/2}} \beta_q \log_a q (\text{mod } p - 1)$$

Tầng 3. Trên tầng 2 chúng ta tìm được số lượng đủ lớn phương trình, chúng ta giải hệ phương trình tuyến tính thu được và tìm ra $\log_a(H + c), \log_a q$.

Tầng 4. Để tìm x, chúng ta đưa ra giới hạn mới L^2. Bằng cách chọn ngẫu nhiên,chúng ta tìm một giá trị w, thỏa mãn biểu thức

$$a^w b \equiv \prod_{q<L^{1/2}} q^{g_q} \prod_{L^{1/2}\le u<L^2} u^{h_u} \pmod p$$,q,u là số nguyên tố

Trong biểu thức này với sự có mặt của số nguyên tố mới là u có độ lớn trung bình.

Tầng 5. Bằng cách tương tự như tầng 2 và 3 chúng ta tìm lôgarith của một số số nguyên tố u, u xuất hiện trong tầng 4.

Tầng 6. Chúng ta tìm đáp số

$$x = \log_a b \equiv -w + \sum_{q<L^{1/2}} g_q \log_a q + \sum_{L^{1/2}\le u<L^2} h_u \log_a u (\mathrm{mod}\ p-1)$$

Thuật toán này có độ phức tạp là$O(\exp((\log p \log\log p)^{1/2}))$ lệnh số học.

1.2.3.5 Thuật toán LOGsmooth

Giả sử q là số nguyên tố, và là ước của p-1. Khi đó tập nghiệm của phương trình $x'^q = 1$ trong trường Z_p gồm các phần tử $1, c, c^2, ..., c^{q-1}$, với $c \equiv a^{\frac{p-1}{q}} \pmod p$. Nếu như cho số d và biết được rằng nó thỏa mãn điều kiện phương trình $x'^q = 1$, thì có thể lựa chọn số t sao cho $d = c^t, 0 \le t \le q-1$.

Giả sử $p-1 = q^k l$, với q và l nguyên tố cùng nhau. Chúng ta sẽ tìm số $u_i, i = 0,1,...,k$, mà chúng thỏa mãn

$$(ba^{-u_i})^{lq^{k-i}} \equiv 1 (\mathrm{mod}\ p) \quad (3.4)$$

Khi i=k thì chúng ta có đồng dư

$$(ba^{-u_k})^l \equiv 1 (\mathrm{mod}\ p)$$

Từ (3.2) sẽ tương đương

$$(a^{(x-u_k)})^l \equiv 1(\mathrm{mod}\, q^k)$$

Bởi vì ord(a)=p-1, nên đẳng thức cuối cùng cho ta $(x-u_k)l$ chia hết cho p-1, có nghĩa

$$x \equiv u_k (\mathrm{mod}\, q^k)$$

Chúng ta tìm các đồng dư thức như vậy đối với các ước q của p-1, có thể tìm được x (mod p-1) bằng định lý phần dư trung hoa.

Vấn đề còn lại là tìm u_i thế nào để thỏa mãn phương trình (3.4). Chúng ta có thể đặt $u_0 = 1$. Nếu như một số u_i tìm được, thì từ (3.4) dẫn đến $(ba^{-u_i})^{lq^{k-i-1}}$ thỏa mãn phương trình $x'^q = 1(\mathrm{mod}\, p)$. Lúc này có thể tìm t sao cho

$$(ba^{-u_i})^{lq^{k-i-1}} \equiv c^t (\mathrm{mod}\, p).$$

Chúng ta đặt $u_{i+1} = u_i + tq^i$. Lúc này

$$(ba^{-u_i+1})^{lq^{k-i-1}} \equiv c^t a^{-tlq^{k-1}} \equiv 1(\mathrm{mod}\, p)$$

Như vậy điều này có nghĩa thỏa mãn (3.4)

Nhờ vậy mà chúng ta tìm u_k bằng cách thực hiện theo sơ đồ: $u_{0=1}, r_i \equiv (ba^{-u_i})^{lq^{k-i-1}} (\mathrm{mod}\, p)$, $t_i = \log_c r_i, u_{i+1} = u_i + t_i q^i$.

Chúng ta xem ví dụ sau

Tìm số n sao cho $2^x \equiv 74(\mathrm{mod}163)$

Ở đây a=2,b=74,p=163, $p-1 = 2.3^4$

Đặt q=3. Khi đó k=4 và l=2. Ngoài ra $c \equiv 2^{\frac{p-1}{3}} = 2^{54} \equiv 104(\mathrm{mod}163)$,chúng ta có thể biểu diễn thuật toán qua bảng sau

I	0	1	2	3

r_i	1	58	1	104
t_i	0	2	0	1
u_{i+1}	1	7	7	34

Từ đây $x \equiv 34(\mathrm{mod}\,81)$ (3.5)

Bây giờ chọn q=2. Lúc này k=1,l=81 và $c \equiv 2^{\frac{p-1}{2}} \equiv -1(\mathrm{mod}\,163)$. Tương tự ta lập bảng

I	0
r_i	-1
t_i	1
u_{i+1}	2

Từ đây chúng ta có $x \equiv 2(\mathrm{mod}\,2)$ (3.6)

Từ (3.5) và (3.6) suy ra $x \equiv 34(\mathrm{mod}\,162)$

1.2.3.6 Thuật toán index-calculus

Cố định số nguyên tố p, số tự nhiên n>1, đặt $q = p^n$. Giả sử a là phần tử sinh của nhóm cyclic $F(q)^*$. Chúng ta muốn giải phương trình

$a^x = b$ trong trường F(q). Để làm điều này chúng ta sử dụng các thuật toán với một cơ sở nhân tử. Chúng ta xem thuật toán index-calculus sau

Ý tưởng của thuật toán này là , từ đẳng thức

$$\prod_{i=1}^{m} x_i = \prod_{j=1}^{n} y_j$$

Với các phần tử x_i, y_j nằm trong trường hữu hạn Z_p, thì

$$\sum_{i=1}^{m} \log_a x_i \equiv \sum_{j=1}^{n} \log_a y_j (\text{mod } p-1) \quad (5.7)$$

Khi nhận được số lượng đủ lớn biểu thức (5.7)(điều kiện là ít nhất là phải có một phần tử g, mà $\log_a g$ đã biết), thì chúng ta có thể giải hệ phương trình tuyến tính với ẩn là $\log_a x_i$ và $\log_a y_j$ trong vành Z_{p-1} với điều kiện là số lượng ẩn trong hệ không quá lớn.

Phương pháp đơn giản để tạo ra biểu thức (5.7) – chọn phần tử bất kỳ $g \in Z_p$, tính $u = a^g (\text{mod } p)$ và bằng cách lựa chọn chúng ta thử tìm số thỏa mãn điều kiện sau

$$u = \prod p_i$$

Từ ý tưởng trên ta có thuật toán cụ thể sau:

Tầng 1. (Tính toán ban đầu). Trường F(q) đồng cấu với $F(p)[y]/f(y)$, với $f(y) \in F(p)[y]$ là đa thức bất khả quy bậc n. Cho nên bất kỳ thành phần của trường F(q) được biểu diễn dưới dạng đa thức bậc không vượt quá n-1. Và nhân các đa thức như vậy sẽ rút gọn theo modulo f(y), điều này chúng ta đã tìm hiểu ở chương trường số. Phần tử $a_1 = a^{(q-1)/(p-1)}$ có bậc là p-1 và tạo thành $F(p)^*$. Với sự hỗ trợ của nó chúng ta lập bảng logarithm "hằng số"- có nghĩa là phần tử của trường nguyên tố $F(p) \subseteq F(q)$. Chúng ta tính $a_1^0 = 1, a_1, a_1^2, ..., a_1^{p-2}$.

Tầng 2. (Lựa chọn cơ sở nhân tử). Cơ sở nhân tử $B \subseteq F(q)$ thành lập từ tất cả các đa thức bất khả quy g bật không lớn hơn t, ở đây t là một số tham số, t<n

Tầng 3. (Tìm biểu thức). Lựa chọn ngẫu nhiên m, $1 \le m \le q-2$, chúng ta tìm các giá trị sao cho thỏa mãn biểu thức

$$a^m \equiv c_0 \prod_{g \in B} g^{\alpha_g(m)} (\text{mod } f(y)),$$

Với $c_0 \in F(p)$, từ đây chúng ta tìm được biểu thức

$$m \equiv \log_a c_0 + \sum_{g \in B} \alpha_a(m) \log_a g (\mathrm{mod}\, q-1),$$

ở đây $\log_a c_0$ chúng ta đã biết, còn $\log_a g$ chúng ta chưa biết độ lớn.

Tầng 4. (tìm thuật toán cho các phần tử của cơ sở nhân tử). Khi tìm ở tầng 3 với số lượng đủ lớn các biểu thức (lớn hơn |B|), chúng ta giải hệ phương trình tuyến tính trong vành Z_{q-1} và tìm ra $\log_a g$ với $g \in B$

Tầng 5. (Tìm lôgarit riêng). Chúng ta tìm một giá trị của m sao cho

$$b \cdot a^m \equiv c_1 \prod_{g \in B} g^{\beta_g} (\mathrm{mod}\ f(x)),$$

ở đây $c_1 \in F(p)$. Từ đây chúng ta tìm ra giá trị cần tìm

$$\log_a b \equiv -m + \log_a c_1 + \sum_{g \in B} \beta_q \log_a g (\mathrm{mod}\, q-1)$$

CHƯƠNG 2: XÂY DỰNG HỆ MẬT AFFINE – ELGAMAL

2.1 Lý thuyết về mật mã Affine

Mật mã Affine là một dạng mật mã thay thế dùng một bảng chữ cái, trong đó mỗi chữ cái được ánh xạ tới một số sau đó mã hóa qua một hàm số toán học đơn giản. Mã Affine là một phép dịch Caesar, trong đó các chữ cái được mã hóa với hàm

y = (x+b) mod 26, với b là bước dịch.

2.1.1 Mô tả

Trong mật mã Affine, đầu tiên bảng chữ cái của thông điệp cần mã hóa có kích thước *m* sẽ được chuyển thành các con số tự nhiên từ *0...m - 1*. Sau đó dùng một hàm mô đun để mã hóa và chuyển thành bản mã.

Hàm mã hóa cho một ký tự như sau:

$$E(x) = (ax + b) \bmod m$$

Với *m* là kích thước bảng chữ cái, *a* và *b* là khóa. Giá trị *a* được chọn sao cho *a* và *m* là nguyên tố cùng nhau. Hàm giải mã là:

$$D(x) = a^{-1} (x - b) \bmod m$$

Với a^{-1} là nghịch đảo của *a* theo module *m*. Tức là

$$1 = a.\ a^{-1} \bmod m$$

Nghịch đảo module của *a* chỉ tồn tại nếu *a* và *m* là nguyên tố cùng nhau. Hàm giải mã là hàm ngược của hàm mã hóa:

$$\begin{aligned} D(E(x)) &= a^{-1} (E(x) - b) \bmod m \\ &= a^{-1} (((ax + b) \bmod m) - b) \bmod m \\ &= a^{-1} (ax + b - b) \bmod m \\ &= a^{-1} ax \bmod m \\ &= x \bmod m \end{aligned}$$

Ví dụ

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

Mã hóa

Giả sử k = (7,3). Như đã nêu ở trên, 7^{-1} *mod* 26 = 15. Hàm mã hóa là

$e_k(x) = 7x + 3$

Và hàm giải mã tương ứng là:

$d_k(x) = 15(y-3) = 15y-19$

Ở đây, tất cả các phép toán đều thực hiện trên Z_{26}. Dùng các tính toán trên Z_{26}, ta có:

$d_k(e_k(x)) = dk(7x+3)$

$= 15(7x + 3) – 19 = x + 45 – 19 = x.$

Mã hóa bản “hot”. Trước tiên biến đổi các chữ h, o, t thành các thặng dư theo modulo 26. Ta được các số tương ứng là 7, 14, 19. Bây giờ sẽ mã hóa:

7*7+3 *mod* 26 = 52 *mod* 26 = 0

7*14 + 3 *mod* 26 = 101 *mod* 26 = 23

7*19 + 3 *mod* 26 = 136 *mod* 26 = 6

Vậy 3 ký tự của bản mã là 0, 23, 6 tương ứng với xâu ký tự là AXG.

Giải mã

Giải mã thông điệp “FBWGC” với m = 7, b = 7

Số hóa thôgn điệp FBWGC ta có dãy số 05 01 22 06 02

Để giải mã, ta tính m^{-1}. Ta có:

$$mm^{-1} = 1[mod\ 26]$$

$$\Leftrightarrow 7m^{-1} = 1[mod\ 26]$$

$$\Leftrightarrow 7m^{-1} = 1 + 4.26[mod\ 26]$$

$$\Leftrightarrow 7m^{-1} = 105[mod\ 26]$$

$$\Leftrightarrow 7m^{-1} = 7.15[mod\ 26]$$

$$\Leftrightarrow 7^{-1}.7m^{-1} = 7^{-1}.7.15[mod\ 26]$$

$$\Leftrightarrow m^{-1} = 15[mod\ 26]$$

Áp dung công thức giải mã P = m(C - b) [*mod* n] Ta tính:

$$P_1 = 15(5 - 7)\ [mod\ 26]$$

$$\Leftrightarrow P_1 = -30\ [mod\ 26]$$

$$\Leftrightarrow P_1 = -30 + 2.26\ [mod\ 26]$$

$$\Leftrightarrow P_1 = 22\ [mod\ 26]$$

Tương tự:

$$P_2 = 15(1 - 7) = 14\ [mod\ 26]$$

$$P_3 = 15(22 - 7) = 17\ [mod\ 26]$$

$$P_4 = 15(6 - 7) = 11\ [mod\ 26]$$

$$P_5 = 15(2 - 7) = 3\ [mod\ 26]$$

Ta được thông điệp giải mã “WORLD”

2.1.2 Thám mã mật mã Affine

Mật mã Affine là một ví dụ đơn giản cho ta thấy cách thám hệ mã nhờ dùng các số liệu thống kê. Giả sử Oscar đã thu trộm được bản mã sau:

Bảng 2.1: Tần suất xuất hiện của 26 chữ cái của bản mã

Kí tự	Tần suất	Kí tự	Tần suất	Kí tự	Tần suất	Kí tự	Tần suất
A	2	H	5	O	1	U	2
B	1	I	0	P	3	V	4
C	0	J	0	Q	0	W	0
D	6	K	5	R	8	X	2
E	5	L	2	S	3	Y	1
F	4	M	2	T	0	Z	0
G	0	N	1				

Ví Dụ: Bản mã nhận được từ mã Affine:

FMXVEDRAPHFERBNDKRXRSREFMORUDSDKDVSHVUFEDKPKDL
YEVLRHHRH

Phân tích tần suất của bản mã này được cho ở bảng 1.2

Bản mã chỉ có 57 ký tự. Tuy nhiên độ dài này cũng đủ phân tích thám mã đối với hệ Affine. Các ký tự có tần suất cao nhất trong bản mã là: R (8 lần xuất hiện), D (6 lần xuất hiện), E, H, K (mỗi ký tự 5 lần) và F, S, V (mỗi ký tự 4 lần).

Trong phỏng đoán ban đầu, ta giả thiết rằng R là ký tự mã của chữ e và D là kí tự mã của t, vì e và t tương ứng là 2 chữ cái thông dụng nhất. Biểu thị bằng số ta có: $e_K(4) = 17$ và $e_K(19) = 3$. Nhớ lại rằng $e_K(x) = ax + b$ trong đó a và b là các số chưa biết. Bởi vậy ta có hai phương trình tuyến tính hai ẩn:

$$4a + b = 17$$

$$19a + b = 3$$

Hệ này có duy nhất nghiệm a = 6 và b = 19 (trong Z_{26}). Tuy nhiên đây là một khoá không hợp lệ do UCLN(a,26) = 2 1. Bởi vậy giả thiết của ta là không đúng.

Phỏng đoán tiếp theo của ta là: R là ký tự mã của e và E là mã của t. Thực hiện như trên, ta thu được a =13 và đây cũng là một khoá không hợp lệ. Bởi vậy ta phải thử một lần nữa: ta coi rằng R là mã hoá của e và H là mã hoá của t. Điều này dẫn tới a = 8 và đây cũng là một khoá không hợp lệ. Tiếp tục, giả sử rằng R là mã hoá của e và K là mã hoá của t. Theo giả thiết này ta thu được a = 3 và b = 5 là khóa hợp lệ.

Ta sẽ tính toán hàm giải mã ứng với K = (3,5) và gải mã bản mã để xem liệu có nhận được xâu tiếng Anh có nghĩa hay không. Điều này sẽ khẳng định tính hợp lệ của khoá (3,5). Sau khi thực hiện các phép toán này, ta có $d_K(y) = 9y - 19$ và giải mã bản mã đã cho, ta được:

algorithmsarequitegeneraldefinitionsof

arithmeticprocesses

Như vậy khoá xác định trên là khoá đúng.

Tăng cường độ an toàn cho mã Affine

Mã Affine nói riêng và các loại mật mã thay thế nói chung có thể bị tấn công bởi việc phân tích tần suất ký tự, và theo đó không an toàn cho các thông

điệp ngắn. Tuy nhiên có một số phương pháp cải thiện độ an toàn cho mã Affine. Hai cách đầu tiên sử dụng một bảng ký tự lớn hơn 26 ký tự tiếng anh (ví dụ bảng ký tự tiếng việt có dấu).

- Thêm các ký tự vô nghĩa trong bảng chữ cái một cách ngẫu nhiên để làm nhiễu phép phân tích tần suất.

- Biến đổi một ký tự trong bản rõ thành một vài ký tự trong bản mã. Ví dụ nếu ta dùng bảng chữ cái 100 ký tự, ta có thể liên kết rất nhiều ký tự với mỗi một ký tự trong bản rõ, với tần suất xuất hiện thông thường của nó (ví dụ 12 ký tự cho chữ e, 9 ký tự cho chữ t...). Sau đó ta chọn ngẫu nhiên ký tự để thay thế mỗi lần xuất hiện. Trong bản mật, mỗi ký tự sẽ xuất hiện với số lần xấp xỉ như nhau. Tuy nhiên bản mật vẫn chưa hoàn toàn ngẫu nhiên, các cặp ký tự và từ thông dụng vẫn có thể được phát hiện.

- Sử dụng ngôn ngữ. Chúng ta có thể làm nhiễu quá trình phân tích xác xuất bằng việc sử dụng các từ thay thế từ các ngôn ngữ khác, hoặc bằng việc chọn các từ thay thế một cách cẩn thận. Ví dụ, có ít nhất hai tiểu thuyết tiếng anh đã được viết mà không sử dụng ký tự e. Một trong số đó là Gadsby của tác giả Ernest Vincent Wright. Tác giả đã gỡ bỏ phím E trên máy đánh chữ để viết quyển sách này. Đoạn đầu tiên của quyển sách như sau:

If youth, throughout all history, had had a champion to stand up for it; to show a doubting world that a child can think; and, possibly, do it practically; you wouldn’t constantly run across folks today who claim that “a child don’t know anything.” A child’s brain starts functioning at birth; and has, amongst its many infant convolutions, thousands of dormant atoms, into which God has put a mystic possibility for noticing an adult’s act, and figuring out its purport.

Với người đọc, không có gì khó khăn để hiểu được ý nghĩa của đoạn trên, nhưng sẽ gây trở ngại lớn với kẻ tấn công nếu nó được mã hóa theo Affine (hay

mã thay thế khác). Bảng thống kê dưới đây cho thấy ký tự xuất hiện nhiều nhất là A và không có ký tự E nào xuất hiện, do đó việc tấn công dựa vào phân tích xác xuất là khó.

Bảng 2.2: Tần suất xuất hiện các bảng mã trong ví dụ

A	B	C	D	E	F	G	H	I	J	K	L
10.96	2.14	2.66	4.12	0.00	2.15	3.61	4.91	8.81	0.23	1.18	5.32

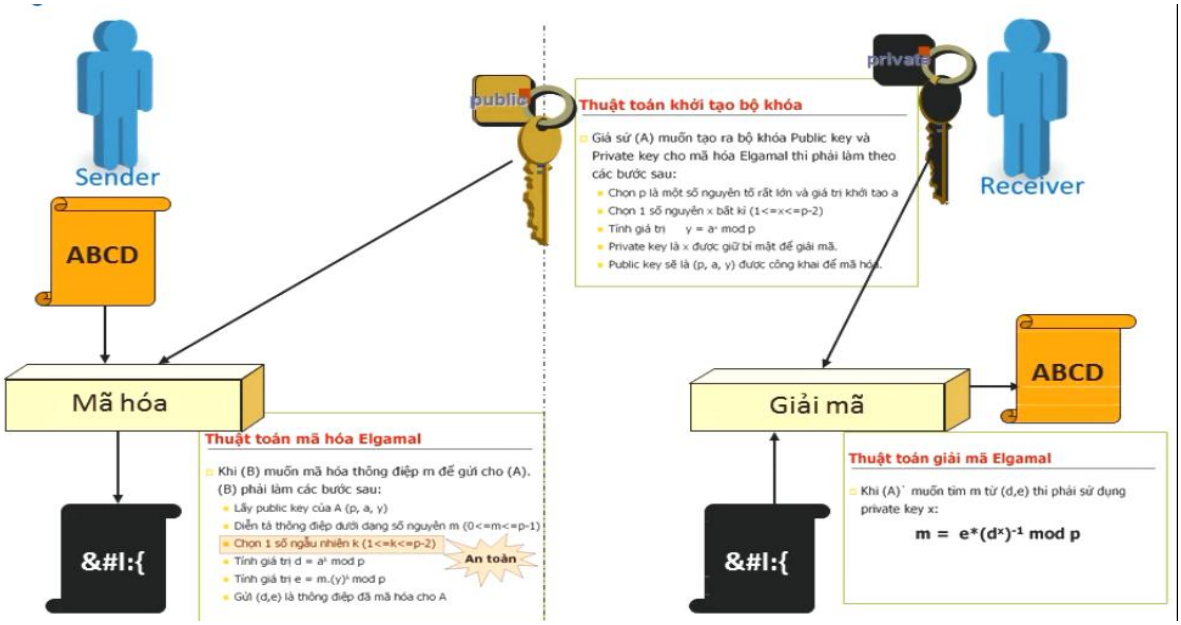
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2.07	8.61	10.42	1.91	0.05	4.77	6.97	8.50	4.16	0.31	2.80	0.04	3.18	0.11

Một phương pháp khác là sử dụng các từ theo ngữ âm, hoặc sử dụng các cách viết tắt. Ví dụ “nite” thay cho “night”, “txt” thay cho “text”, “AFAIK” thay cho “as far as I know”, “2” thay cho “to”, “4” thay cho “for”, “8” thay cho “ate”, hoặc sử dụng các biểu tượng cảm xúc như ;-)như một phần của văn bản. Theo cách đó việc phân tích xác xuất cũng sẽ mang lại kết quả khác với ngôn ngữ thông thường.

- Đổi chỗ ký tự. Phép mã hóa thay thế có thể được phối hợp với việc đổi chỗ, theo đó thứ tự của các ký tự trong bản mã được sắp xếp theo một trật tự đặc biệt. Cách này sẽ phá vỡ các cặp từ ngữ thường xuất hiện, và do đó cũng không thể đoán được dễ dàng.

2.2 Hệ mật mã ElGamal:

2.2.1 Hệ mật mã ElGamal:



Hình 2.1: Hệ mật mã ElGamal

Hệ mật mã ElGamal

Là một hệ mã hóa bất đối xứng dựa trên biến thể của thủ tục trao đổi khóa Diffie - Hellman, trên cơ sở bài toán lôgarith rời rạc. Với các thủ tục trao đổi khóa như sau:

a.Thủ tục tạo khóa

Mỗi bên liên lạc A, B tạo cho mình một cặp khóa công khai – bí mật theo các bước sau:

Bước 1: Chọn một số nguyên tố p lớn sao cho bài toán lôgaritrời rạc trong Z_p là khó giải và α là một phần tử nguyên thủy ($\alpha \in ¢^*_p$)

Bước 2: Chọn một số nguyên a ngẫu nhiên với $1< a < p – 1$ và tính

$\alpha^a \bmod p$

Bước 3: + Khóa công khai là bộ 3 số: (p, α, α^a) của người nhận và gửi đi cho người sử dụng cần mã hóa thông tin bí mật gửi cho mình.

+ Khóa bí mật là a

b. Mã hóa

Giả sử B cần gửi bản tin M cho A, B sẽ thực hiện các bước sau:

Bước 1: B nhận khóa công khai của A: (p, α, α^a)

Bước 2: B chọn số nguyên k ngẫu nhiên với 1< k < p – 1 và tính giá trị theo công thức

$$\begin{cases} \gamma = \alpha^k \bmod p \\ \delta = M\left(\alpha^a\right)^k \bmod p \end{cases}$$

Giả sử bản tin đã được biểu thị dưới dạng một số nguyên M trong dải { 1,…..,p – 1} Phép tính mũ được tính bằng thuật toán nhân và bình phương theo modulo.

Bước 3: B gửi bản mã C = (γ, δ) cho A

Ta nhận thấy bản mã C được ghép từ γ và δ nên nó có độ dài bit bằng 2 lần độ dài của M, đây là nhược điểm của hệ mật này.

c. Giải mã

A nhận bản mã C từ B và tiến hành giải mã theo các bước sau:

Bước 1: A sử dụng khóa bí mật a để tính:

$\gamma^{p-1-a} \bmod p = \alpha^{-ak} \bmod p$ (Chú ý $\gamma^{p-1-a} = (\alpha^k)^{-a} = \alpha^{-ak}$)

Bước 2: A khôi phục bản rõ bằng cách tính:

$\delta^{p-1-a} \bmod p = M\ \alpha^{-ak}\ \alpha^{-ak} \bmod p = M.$

Hệ mật khóa công khai Elgamal trong Zp

a. Tạo khóa:

Cho p là số nguyên tố sao cho bài toán logarithm rời rạc trong Zp là khó giải. Cho $\alpha \in Zp$ là phần tử nguyên thủy. Giả sử P = Zp,

C = Zp*Zp. Ta định nghĩa:

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

Các giá trị p, α, β được công khai, còn a là khóa bí mật.

b. Mã hóa:

Chọn một số ngẫu nhiên bí mật $k \in Zp-1$, ta xác định:

$$e_k(x,k) = (y1, y2)$$

trong đó

$$y1 = \alpha^k \bmod p$$

$$y2 = x\beta^k \bmod p$$

c. Giải mã:

Với $y1, y2 \in Zp$ ta xác định:

$$d_k(y_1, y_2) = y_2 (y_1^a)^{-1} \textit{ mod } p$$

Ví dụ 1:

Cho p = 2579, α = 2, a = 765. Khi đó

$$\beta = 2765 \bmod 2579 = 949$$

Bây giờ ta giả sử Alice muốn gửi thông báo x = 1299 tới Bob. Giả sử số ngẫu nhiên k mà cô chọn là k = 853. Sau đó cô ta tính

$$y_1 = 2^{853} \textit{ mod } 2579 = 435$$

$$y_2 = 1299 \times 949853 \textit{ mod } 2579 = 2396$$

Khi đó Bob thu được bản mã y = (435,2396), anh ta tính

$$x = 2396 \times (435^{765})^{-1} \textit{ mod } 2579 = 1299$$

Đó chính là bản rõ mà Alice đã mã hoá.

Ví dụ 2:

Tạo khóa

Bước 1: A chọn p = 17 và phần tử nguyên thủy α = 3 của $¢^*_{17}$.

Bước 2: A chọn khóa bí mật a = 6 và tính $\alpha^a \bmod p = 3^6 \bmod 17 = 15$

Bước 3:

+ Khóa công khai của A là bộ 3 số (p, α, α^a) = (17,3,15)

+ Khóa bí mật của A là : a = 6

Mã hóa

Giả sử B cần gửi bản tin M = 7 cho A.

Bước 1: B nhận khóa công khai của A: (p, α, α^a) = (17,3,15)

Bước 2: B chọn số nguyên k = 4 và tính:

$$\begin{cases} \gamma = \alpha^k mod\ p = 3^4\ mod\ 17 = 13 \\ \delta = M\ (\alpha^a)^k mod\ p = 7.(15)^4 mod\ 17 = 10 \end{cases}$$

Bước 3: B gửi bản mã C = (γ, δ) = (13,10) cho A.

Giải mã

A nhận được bản mã C = (γ, δ) = (13,10) và tiến hành giải mã.

Bước 1: A sử dụng khóa bí mật a = 6 để tính:

$$\gamma^{p-1-a} \bmod p = 13^{10} \bmod 17 = 16$$

Bước 2: A khôi phục bản rõ bằng cách tính

$\delta\gamma^{p-1-a} \bmod p = 10.16 \bmod 17 = 7 = M$

Nhận xét:

- Để tìm khóa bí mật a (từ α^a) thám mã phải giải bài toán logarit rời rạc (tính $a = \log_\alpha \alpha^a$), với trường hợp p lớn thì không thể giải được, hệ mật là an toàn

- Hiệu quả truyền tin thấp, vì tốc độ mã chỉ đạt $R_{mã}$= ½.

Ưu, nhược điểm

- Ưu điểm:

- Độ phức tạp của bài toán logarit lớn nên độ an toàn cao.
- Bản mã phụ thuộc vào bản rõ x và giá trị ngẫu nhiên nên từ 1 bản rõ ta có thể có nhiều bản mã khác nhau.

- Nhược điểm:

- Tốc độ chậm (Do phải xử lý số nguyên lớn)
- Dung lượng bộ nhớ dành cho việc lưu trữ khóa cũng lớn.

2.2.2 Thám mã hệ ElGamal

Để thám mã hệ Hệ ElGamal, ta cần phải giải bài toán logarit rời rạc. Chúng ta có 2 thuật toán để giải bài toán logarit rời rạc là:

- Thuật toán Shank
- Thuật toán Pohlig - Hellman

Trong đó thuật toán thám mã Shank được sử dụng nhiều hơn cả nên trong luận văn chỉ trình bày thám mã Elgamal bằng thuật toán Shank

Thuật toán Shank

Thuật toán này có tên gọi khác là thuật toán thời gian - bộ nhớ. Tư tưởng của thuật toán là nếu ta có đủ bộ nhớ thì có thể sử dụng bộ nhớ đó để giảm thời gian thực hiện của thuật toán.

Input: Số nguyên tố p, phần tử nguyên thủy a cua Z*p, số nguyên y.

Output : Cần tìm a sao cho $\beta = \alpha^a \bmod p$

Thuật toán :

Gọi $m = [(p-1)^{1/2}]$ (lấy phần nguyên).

Bước 1:Tính $\alpha^{mj} \bmod p$ với $0 <= j <= m-1$.

Bước 2: Sắp xếp các cặp $(j, \alpha^{mj} \bmod p)$ theo $\alpha^{mj} \bmod p$ và lưu vào danh sách L1.

Bước 3: Tính $\beta * \alpha^{-i} \bmod p$ với $0 <= i <= m-1$.

Bước 4: Sắp xếp các cặp $(i, \beta * \alpha^{-i} \bmod p)$ theo $\beta * \alpha^{-i} \bmod p$ và lưu vào danh sách L2.

Bước 5: Tìm trong hai danh sách L1 và L2 xem có tồn tại cặp $(j, \alpha^{mj} \bmod p)$ và $(i, \beta * \alpha^{-i} \bmod p)$ nào mà $\alpha^{mj} \bmod p = \beta * \alpha^{-i} \bmod p$ (tọa độ thứ hai của hai cặp bằng nhau).

Lưu ý: Vì $\alpha^{mj} = \beta * \alpha^{-i} => \alpha^{mj-i} = \beta$ nên bước 5 luôn thành công.

Bước 6: Tính $a = \log_\alpha \beta = (mj + i) \bmod (p - 1)$. Kết quả này có thể kiểm chứng từ công thức:

$$\alpha^{mj} \bmod p = \beta * \alpha^{-i} \bmod p$$

$$=> \alpha^{mj+i} \bmod p = \beta \bmod p$$

$$=> \log_\alpha \beta = (mj + i) \bmod (p - 1) = a.$$

Độ phức tạp của thuật toán Shank

Phụ thuộc vào $m=[(p-1)^{1/2}]$, với giá trị của m, chúng ta cần tính các phần tử thuộc 2 danh sách L1 và L2 , đều là các phép toán lũy thừa phụ thuộc vào j và i ; mà j và I lại phụ thuộc vào m nên có thể nhận thấy là thuật toán này chỉ có thể áp dụng trong những trường hợp p nhỏ

Ví dụ:

Cho $p=79$, $\alpha=2$, $\beta=55$ .Tìm a theo thuật toán Shank

Bài giải

Tính $m=[(p-1)^{1/2}] = 9$

Bước 1: Tính $t= \alpha^{m.j} \bmod p$ với $0\leq j\leq m-1$: $2^{9.j} \bmod 79$ với $0\leq j\leq 8$

j =0 => t=1; t_j (0,1)

j=1=> $2^9 \bmod 79 = 38$; t_j (1,38)

j=2=>$2^{18} \bmod 79 = 13$; t_j (2,22)

j=3=>$2^{27} \bmod 79 = 46$; t_j(3,46)

j=4=>$2^{36} \bmod 79$;

x	a	d=1
36	2	x
18	4	x
9	16	16
4	19	x
2	45	x
1	50	10

=> t_j(4,10)

j=5=>$2^{45} \bmod 79$

x	a	d=1
45	2	2
22	4	x
11	16	32
5	19	55
2	45	x
1	50	64

=> $t_j(5,64)$

J=6 => $2^{54} \bmod 79$

x	a	d=1
54	2	x
27	4	4
13	16	64
6	19	x
3	45	36
1	50	62

=>$t_j(6,62)$

j =7 => $2^{63} \bmod 79$

x	a	d=1
63	2	2
31	4	8
15	16	49
7	19	62
3	45	25
1	50	65

=> $t_j(7,65)$

j =8 =>$2^{72} \bmod 79$

x	a	d=1
72	2	x
36	4	x
18	16	x
9	19	19

4 45 x

2 50 x

1 51 21

=>t_j(8,21)

Bước 2: Sắp xếp các cặp t_j theo hướng tăng dần của $\alpha^{m.j}$ mod p

(0,1);(4,10);(2,13);(8,21);(1,38);(3,46);(6,62); (5,64);(7,65)

Bước 3: tính $\beta.\alpha^{-i}$ mod p với $0 \le i \le m-1$: $55.(2^{-i})$ mod 79 với $0 \le i \le 8$

Hay tính $55.(2^i)^{-1}$ mod 79

i =0: 2^0 mod 79 = 1 => 1^{-1} mod 79=1 => $55.(2^0)^{-1}$ mod 79 =55

= >t_i(0,55)

i =1:2^1 mod 79=2 => 2^{-1}mod 79=40(tính theo EuClic)

=> $55.(2^1)^{-1}$ mod 79 = 55.40 mod 79=67

= >t_i(1,67)

i =2:2^2 mod 79 =4 => 4^{-1} mod 79= 20

=>$55.(2^2)^{-1}$ mod 79= 55.20 mod 79=73

= >t_i(2,73)

i =3: 2^3 mod 79 =8 => 8^{-1} mod 79 = 10

=> $55.(2^3)^{-1}$ mod 79=55.10 mod 79=76

= >t_i(3,76)

i =4:2^4 mod 79=16 => 16^{-1} mod 79 = 5

=> $55.(2^4)^{-1}$ mod 79 = 55.5 mod 79=38

= >t_i(4,38)

i =5:2^5mod 79=32=>32^{-1}mod 79=42

= > $55.(2^5)^{-1}$ mod 79=55.42 mod 79= 74

= >t_i(5,74)

i =6:2^6 mod 79 =64 =>64^{-1}mod 79= 21

= > $55.(2^6)^{-1}$ mod 79 = 55.21 mod 79=49

= >t_i(6,49)

i =7:2^7 mod 79=49 => 49^{-1} mod 79=50

= > $55.(2^7)^{-1}$ mod 79=55.50 mod 79= 64

= >t_i(7,64)

i =8:2^8 mod 79=19 => 19^{-1} mod 79 =25

= > $55.(2^8)^{-1}$ mod 79= 55.25 mod 79 = 32

= >t_i(8,32)

Bước 4: Sắp xếp các cặp t_i: (I, $\beta.\alpha^{-i}$ mod p) theo thứ tự tăng của $\beta.\alpha^{-i}$ mod p và lưu vào danh sách L_2 :

t_i(8,32) ; t_i(4,38) ; t_i(6,49) ; t_i(0,55) ; t_i(7,64) ; t_i(1,67) ; t_i(4,73) ; t_i(5,74) ; t_i(3,76)

Bước 5: Tìm trong hai danh sách L_1và L_2 xem có tồn tại cặp (j, $\alpha^{m.j}$ mod p) và (I, $\beta.\alpha^{-i}$ mod p) nào mà $\alpha^{m,j}$ mod p= $\beta.\alpha^{-i}$ mod p (tọa độ thứ 2 của hai cặp bằng nhau).

Ta thấy cặp t_j(1,38) và cặp t_i(4,38) có tọa độ thứ 2 bằng nhau cùng bằng 38 . và cặp t_j(5,64) với cặp t_i(7,64) có tọa độ thứ 2 bằng 64

= > chon : bộ 1: j=1 ;i=4 ; bộ 2: j=5;i=7

Bước 6:

Với bộ 1: a=(m.j+i) mod (p-1)

a =(9.1+4) mod (p-1) =13

với bộ 2: a=(9.5+7)mod 78 =52

(Kiểm tra: ta có $\beta \equiv \alpha^a \bmod p$ theo trên tính a=13

= > $\beta \equiv 2^{13} \bmod 79$

x	a	d=1
13	2	2
6	4	x
3	16	32
1	19	55

= > β=55 đúng theo bài ra β =55

Tính theo a=52

= > $\beta \equiv 2^{52} \bmod 79$

x	a	d=1
52	2	x
26	4	x
13	16	16
6	19	x
3	45	9
1	50	55

= > β= 55 đúng theo bài ra)

Độ an toàn của Hệ mật ElGamal

Hệ thống elgamal dự trên bài toán logarit rời rạc. Tính an toàn của nó tùy thuộc vào độ phức tạp của bài toán logarit.

Trong bài toán về hệ Elgamal:

- P là số nguyên tố, a là phàn tử nguyên thủy của Zp.
- Bài toán logarit rời rạc có thể được phát triển như sau: Tìm 1 số mũ x duy nhất, 0 <= x <= p – 2 sao cho $a^x = y \ mod$ p, với y thuộc Zp cho trước.

- Bài toán có thể giải được bởi phương pháp vét cạn (Tức là duyệt tất cả phần tử x) để tìm x thỏa mãn. Bài toán có độ phức tạp là: O(p) (bỏ qua thừa số logarit). Vấn đề đặt ra là nếu p lớn, rất lớn thì để thực hiện phương pháp này cần thời gian rất lớn. Suy ra không khả thi.

Tính đúng đắn của Hệ mật ElGamal

Thuật toán mật mã ElGamal hoàn toàn là đúng đắn.Với cách khôi phục bản tin ban đầu M thấy khôi phục bản tin ban đầu M bằng cách tính:

$$\delta\gamma^{p-1-a} \bmod p = M\,\alpha^{-ak}\,\alpha^{-ak} \bmod p = M.$$

Nên bản rõ nhận được sau giải mã chính là bản rõ ban đầu M

Hệ mật mã Elamal dựa trên bài toán lôgarit rời rạc mà độ phức tạp của bài toán lôgarit lớn nên độ an toàn cao. Như trong ví dụ, bản mã được tính phụ thuộc vào bản rõ x và giá trị ngẫu nhiên k nên có thể có nhiều bản mã khác nhau mà người nhận vẫn giải mã đúng.

Ví dụ

Chọn p 653879; g = 2357; x = 2343

Khi đó y = 18815

Khóa công khai (p,g,y) ; bí mật x

Bản rõ “SOHOC” tương ứng trong bảng mã ASCII là (83; 79; 72; 79; 67)

Số k chọn tương ứng với mỗi số trên trong bản rõ là: (123; 234; 321; 452; 673)

Khi đó bản mã cho mỗi số lần lượt là:

(162568, 472636) ; (401349, 535351) ; (605474, 465483) ; (449955, 53471) ; (493243, 135870)

Để giải mã tính $\alpha^k mod\ p$ tương ứng được

344451; 594440; 33710; 639725; 255772.

Nghịch đảo tương ứng mod p là

590177; 536181; 44051; 446811; 379887.

Bản giải mã tương ứng là dãy số

(83; 79; 72; 79; 67)

Từ đó chuyển về bản rõ ban đầu.

Có thể mô tả qua bảng dưới đây

Với R =α^x mod p, C= α^{-ak} mod p, Z= $M\ (\alpha^a)^k mod\ p$, $Z^{-1} = \delta\gamma^{p-1-a}$ mod p

Bản rõ	Số (M)	k	R	C	Z	Z^{-1}	CZ^{-1}
S	83	123	162568	472636	344451	590177	83
O	79	234	401349	535351	594440	536181	79
H	72	321	605474	465483	33710	44051	72
O	79	452	449955	53471	639725	446811	79
C	67	673	493243	135870	255772	379887	67

Có thể thấy đây là hệ mật mã đa biểu vì với hai bản rõ trùng nhau (chữ O ứng với số 7) nhưng do cách chọn k khác nhau cho hai bản mã khác nhau ((401349, 535351 và (449955,53471)) song việc giải mã vẫn đúng.

Như vậy với cùng một khóa, cùng bản rõ nhưng số k được chọn khác ta sẽ có bản mã gửi đi khác mà việc giải mã vẫn đúng.

Hệ mật ElGamal sẽ bị phá vỡ nếu khóa mật x hoặc k có thể tính được. Để tính được x hoặc k, cần phải giải một trong hai bài toán lôgarith rời rạc, việc giải bài toán này là việc khó.

2.3 Phối hợp mã Affine và ElGamal

Trong thực tế, hệ mật ElGamal thường không được sử dụng trực tiếp. Nguyên nhân chủ yếu đến từ việc quá trình mã hóa/giải mã ElGamal là chậm

và tiêu tốn nhiều tài nguyên máy tính do phải tính toán các hàm logarit... Ngoài ra một hạn chế của mật mã ElGamal là việc mã hóa/giải mã các thông điệp dài khá phức tạp, trong trường hợp đó, thông điệp cần phải được chia nhỏ thành các khối (block) sau đó tiến hành mã hóa/giải mã cho từng khối.

ElGamal thường được sử dụng trong các hệ mật mã "lai" (hypbrid). Tức là sử dụng phối hợp với một hệ mật mã đối xứng (symmetric). Đầu tiên thông điệp sẽ được mã hóa bằng mật mã đối xứng với một khóa bí mật K, khóa K này sau đó sẽ được mã hóa bằng ElGamal. Khóa K sau khi được mã hóa sẽ được gửi kèm với bản mật, phía nhận sẽ lần lượt giải mã khóa K và dùng khóa này để giải mã bản mật. Việc mã hóa khóa K (có kích thước nhỏ hơn rất nhiều so với thông điệp cần mã hóa) là rất nhanh, thông điệp được mã hóa với khóa K cũng có độ an toàn tương đương so với mã hóa bằng ElGamal.

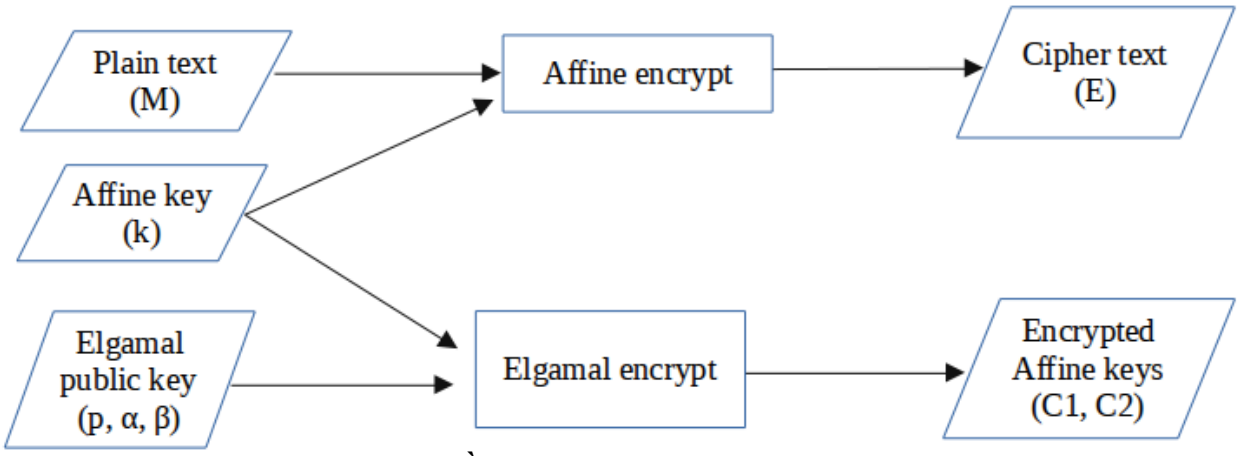
Cụ thể, trong trường hợp này ta sẽ xem xét đến việc phối hợp hệ mật ElGamal và mật mã Affine.

Mã hóa

- Bên giải mã chọn ra cặp khóa Affine là (a, b).

- Mã hóa thông điệp m với cặp khóa (a, b), ta được bản mật e

- Mã hóa cặp khóa a và b với khóa công khai ElGamal (p, α, β) ta được k



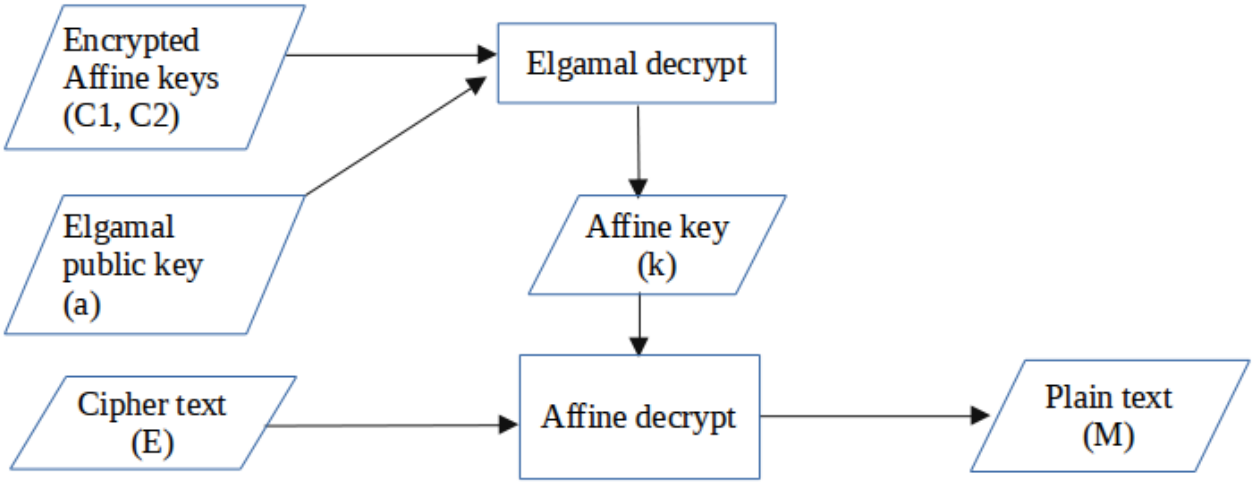
Hình 2.2: Sơ đồ mã hóa Hệ mật Affine – ElGamal

Bên mã hóa gửi bản mật e, khóa k cho bên nhận.

Giải mã

- Dùng khóa bí mật ElGamal để giải mã k, ta được cặp khóa Affine (a, b)

- Dùng a, b để giải mã bản mật e để nhận được m



Hình 2.3: Sơ đồ giải mã Hệ mật ElGamal

Hàm mã hóa

function **encrypt**(plain_text, public_key):

```
affine_key = make_random_affine_key()
add_noise(plain_text)
encrypted_text = affine_encrypt(affine_key, plain_text)
c1, c2 = ElGamal_encrypt(public_key, affine_key)
return (encrypted_text, c1, c2)
```

Hàm giải mã

function **decrypt**(encrypted_text, c1, c2, private_key):

```
affine_key = ElGamal_decrypt(c1, c2, private_key)
```

plain_text = affine_decrypt(affine_key, encrypted_text)

remove_noise(plain_text)

return plain_text

Trong đó, hàm **add_noise** và **remove_noise** có nhiệm vụ thêm và bớt các ký tự gây nhiễu vào văn bản, trong đó tập các ký tự gây nhiễu được định nghĩa trước

function **add_noise**(plain_text):

for i in random_positions_of(plain_text):

plain_text.insert(i, random_noise)

return plain_text

function **remove_noise**(text):

for char in text:

if char is noise:

text.remove(char)

return text

Khóa Affine được sinh ngẫu nhiên như sau, với SYMBOLS là bảng chữ cái được sử dụng để mã hóa.

function **make_random_affine_key**():

while True:

keyA = get_random_integer(2, len(SYMBOLS))

keyB = get_random_integer(2, len(SYMBOLS) - 1)

if UCNL(keyA, len(SYMBOLS)) == 1 and keyB != 0 and keyA > 1:

return keyA, keyB

CHƯƠNG 3: ĐÁNH GIÁ HỆ MẬT MÃ AFFINE- ELGAMAL

3.1 Đánh giá mã Affine

Mã Affine nói riêng và các loại mật mã thay thế nói chung có thể bị tấn công bởi việc phân tích tần suất ký tự, và theo đó không an toàn cho các thông điệp ngắn. Đặc biệt trong trường hợp các văn bản ngắn, kẻ tấn công hoàn toàn có thể sử dụng phương pháp tấn công vét cạn (lần lượt thay thế các ký tự trong bản mã cho đến khi tìm được văn bản có ý nghĩa)! Đối với các văn bản dài, việc tấn công vét cạn này không khả thi.

Việc tấn công dựa trên xác suất có thể được phòng tránh bằng việc thêm nhiễu (các ký tự vô nghĩa đối với nội dung văn bản), hoặc sử dụng các biến thể ngôn ngữ.

Ví dụ với văn bản sau:

In cryptography, the ElGamal encryption system is an asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie - Hellman key exchange. It was described by Taher ElGamal in 1985. ElGamal encryption is used in the free GNU Privacy Guard software, recent versions of PGP, and other cryptosystems. The DSA (Digital Signature Algorithm) is a variant of the ElGamal signature scheme, which should not be confused with ElGamal encryption.

Các ký tự có số lần xuất hiện như sau:

Bảng 3.1: Tần suất xuất hiện của các kí tự trong văn bản

<BS>	)	(	-	,	.	1	5	9	8	a	c	b	e	d	g
15.24	0.21	0.21	0.43	0.86	0.86	0.21	0.21	0.21	0.21	6.87	3.65	1.07	9.01	2.36	3.43

f	i	h	k	m	l	o	n	p	s	r	u	t	w	v	y	x
1.72	6.22	4.29	0.64	2.79	3.65	4.08	5.15	2.79	4.94	5.36	1.72	6.01	1.07	0.64	3.65	0.21

Thêm vào các ký tự gây nhiễu một cách ngẫu nhiên:

I!n*) "!cry&$!!pto#g(*r(+$#a%&p#"'h*$y(), #"'+%t()&'#)h$#!(e'+& (+$E"'!'l#%Gam!al e$*n%$c)ry*p#t+&+#i%o*)$"*n%"+' *sy#(#$%s""%#)('t(e'!$m i**!s !*()"%"'a#n as$%'ym+#m*!e$t$*%&++!ric*$+%)$!)! k"e!$"y(!* $e!nc+r*y$p*#%$&t&&#"#!%+(i)!(*on& al(&$)g+o&r+#ithm% ()f&(or pu+bli+*c"-k#e+%y $"&*c)&&#r'#yp'to"g$&#'#(*%r'a&ph)!(y $w+hic'h"+*%&& "'(&!%$%%#is$ ba*'s"%e+d&# (*%o!+(((("n %the$)+) D*+)i!ffie#++)-$" !!He&l&l)m+"an$$ key* #ex(c!*#*+h#(&)*a"n'$#g)e.$)&%!$+ (+'%I#*$)##t*(& w'a#s *%!"d)(e!&*!'s!&"cr%i!be)&d+ "b%y)$ Ta!&'he!r++ El+$"(ga%$"m+al(#%*+ !"%'in *!"*19&)$&(*$85. &El&)$'&'G!a!m!)(!a'()!++*&l &e*&*nc+r&&"&+%$!)y$+!p!!('ti%#+&o'+n*' i*)$&&&s *us)"ed#)#%)(i%+n(($# "the!)' &'(fr%e*)e'#($G%$*N$""'+U+ P+r&#%#&(i+(!*)#*"'*&)"va#cy G$u*a"+&r*d $&s&(+"()of(#tw'a!!re&$,& rece#'n((t) ve(+++r($"s"!i#(ons!)o"f$'! P*(+GP,(an'd ot'he+r&%# cr!yp%to$*+'%sys%te#m)#s". &Th'e D$SA (D$ig(i*tal *#+Signa!t+u!"re +Algo(r)i+t#hm) is a'$" va)riant(* of t*$he (El'Gamal si%&(gnat#ure schem)%e', %whic%h $sh)ould not be confused with ElGamal & encryption.

Các ký tự có tần suất xuất hiện như sau:

Bảng 3.2: Tần suất xuất hiện của các kí tự sau khi gẫy nhiễu

!	<BS>	#	"	%	$	'	&	)	(	+	*
5.25	6.77	5.06	4.1	4.48	5.63	4.39	5.63	4.68	5.44	5.73	5.34

-	,	.	1	5	9	8	a	c	b	e	d	g
0.19	0.38	0.38	0.1	0.1	0.1	0.1	3.05	1.62	0.48	4.01	1.05	1.53

f	i	h	k	m	l	o	n	p	s	r	u	t	w	v	y	x
0.76	2.77	1.91	0.29	1.24	1.62	1.81	2.29	1.24	2.19	2.39	0.76	2.67	0.48	0.29	1.62	0.1

Sau khi thêm nhiễu, rõ ràng việc phân tích và phỏng đoán nội dung văn bản sẽ gặp khó khăn hơn nhiều. Tuy nhiên cách này vẫn có thể bị tấn công khi các ký tự gây nhiễu được thêm vào không hoàn toàn phá vỡ phân bố xác suất của các ký tự có nghĩa.

3.2 Đánh giá Hệ mật ElGamal

Thuật toán phát triển dựa trên độ khó của bài toàn logarit trong Elgama nên vẫn giữ được ưu điểm khó thám mã tương đương với RSA và Elgamal.

Để thám mã thành công thuật toán Elgamanl độ dài 64 byte, với máy tính đơn có bộ vi xử lý PIV 2.6 GHz, cần thời gian 300000 giờ(khoảng 34 năm). Thế nhưng nếu sử dụng mạng gồm 100000 máy thì thời gian thám mã chỉ còn hơn 3 giờ(theo tài liệu tính toán của RSA Inc)

Thuật toán Elgamal giải quyết tốt vấn đề bảo mạt, nhờ sử dụng vectơ dịch SV theo ma trận hàng. Một số tính năng ưu việt nổi bật của thuật toán này như sau:

Độ bảo mật được tăng cường rất lớn so với các thuật toán khóa mã công khai hiện tại. Với cùng kích thước bài toán 64byte nhu trên, vectơ dịch SV là ma trận 1x64, mỗi phân của SV có giá trị 0 đến 7. Đế thám mã thành công thuật toán Elgamal, ngoài việc vượt qua độ khó của bài toán logarit như trên, cần phải tìm được chính xác SV. Tập không gian SV là$8^{64} = 2^{192} = 10^{192lg2} \approx 10^{58}$ vectơ. Theo trung tâm ứng dụng siêu quốc gia MỸ,(12/2003), một hệ thống siêu mạnh với 1500 máy chủ có thể thực hiện được 20 nghìn tỉ(2.10^{13}), phép tính trên giây. Với hệ thống siêu mạng này, theo ước tính của tác giả, thời gian để tìm ra chính xác SV bằng phương pháp vét cạn để thám mã là $10^{58}/(2.10^{13})$ =5.10^{44}(giây) ≈$1.6.10^{37}$ (năm). Rõ ràng độ bảo mật tăng lên vô cùng lớn.

Kích thước dữ liệu sau mã hóa không thay đổi. So với thuật toán Elgamal, ứng với mỗi dữ liệu x sẽ cho ra văn bản mã c gồm y_1 và y_2. Riêng thuật toán Elgamal, chỉ sinh ra văn bản mã C[i] có kích thước bằng với kích thước văn bản gốc X[i].

Chống thám mã theo xác suất xuất hiện. Các phương pháp mã hóa theo mô hình khóa đối xứng đều có cùng nhược điểm là tạo ra các khối văn bản mã giống nhau với cùng văn bản gốc. Nhờ phép XOR với văn bản mã liền trước, hệ mật Elgamal sẽ tạo ra các văn bản mã khác nhau cho dù văn bản gốc đầu đều giống nhau. Điều nayloại bỏ hoàn toàn thám mã theo xác suất.

Nhận ra sự thay đổi dữ liệu trên đường truyền. Một ai đó cố tình phá hoại hệ thống bảo mật bằng cách tạo ra các khối giống với khối văn bản mã, hay cố tình sủa đối nội dung văn bản mã trên đường truyền. Theo thuật toán Elgamal

và RSA, nơi nhận không phát hiện điều này. Kĩ thuật XOR các văn bản mã với nhau trong hệ mật Elgamal giúp giải quyết triệt để vấn đề này.

Tốc độ thực thi cao nhờ sử dụng các phép gần với ngôn ngữ máy(phép dịch vòng, phép XOR)

Hiệu quả trong thiết kế phần cứng: Sử dụng chung khoảng 2/3 kiến trúc phần cứng cho quá trình mã hóa và giải mã.

3.3 Hệ mật Affine – ElGamal

- Một vấn đề đối với hệ mật Affine và các loại mật mã đối xứng khác đó là việc bảo mật khóa chung trong quá trình gửi khóa này cho bên nhận. Hệ mật Affine - ElGamal đảm bảo việc khóa này khó có thể bị lộ trong quá trình gửi đến người nhận. Do khóa này được mã hóa bằng khóa công khai ElGamal.

- Tốc độ thực thi được cải thiện. Trước hết, văn bản được mã hóa bằng khóa Affine, do không phải tính toàn các phép toàn lôgarit, và độ lớn khóa Affine là không lớn. Bảng dưới đây là kết quả mã hóa Affine và ElGamal cho văn bản có độ dài lớn thực thi trên máy tính Intel Core i7 2.60GHz, 8GiB DDR3 1600 MHz

Bảng 3.3: So sánh tốc độ mã hóa văn bản.

Độ dài văn bản (ký tự)	Thời gian mã hóa Affine	Thời gian mã hóa ElGamal
4680	4 (ms)	1.45 (s)
6095	5.36 (ms)	1.87 (s)
18580	35 (ms)	5.81 (s)

- Về độ an toàn, hệ thống đảm bảo được tính bí mật của khóa chung (khóa Affine) bằng cách mã hóa bằng phương pháp ElGamal trước khi được gửi lên kênh truyền.

- Văn bản trước khi được mã hóa được thêm vào các ký tự gây nhiễu để gây khó khăn cho phép phân tích tần suất.

KẾT LUẬN

1. Kết quả đạt được

Luận văn tiến hành nghiên cứu giải quyết bài toán về mã hóa, xây dựng Hệ mật mã đơn Affine - ElGamal. Từ việc giải quyết bài toán. Bài toán là nền tảng cho nhiều ứng dụng quan trọng thực tế như quảng cáo nhắm mục tiêu, các hệ thống cung cấp tiếp thị dịch vụ thương mại điện tử, thu tín điện tử tới đúng người dùng …

Những kết quả chính mà đồ án đạt được:

- Tìm hiểu được thuật toán mã hóa công khai, hệ mật mã
- Tìm hiểu bài toán Lôgarit rời rạc, các thuật toán trên Lôgarit rời rạc.
- Tìm hiểu mã Affine, mật mã Elgamal – Xây dựng biến thể của hệ mật ElGamal: Hệ mật mã Affine – ElGamal.

2. Hạn chế:

- Nghiên cứu Hệ mã Affine chỉ sử dụng các ký tự là bảng chữ cái, bảng chữ cái không lớn nên bị giới hạn bởi các bảng chữ cái
- Dễ bị tấn công bằng cách phân tích tần số và khó phòng ngừa.
- Không có khả năng phục hồi văn bản gốc
- Hệ mật mã chỉ áp dụng được cho các ký tự trong bảng chữ cái Tiếng anh.
- Tuy tốc độ mã hóa và giải mã được cải thiện rõ nét, nhưng không ngoại lệ, hệ mật Elgamal cũng giống như các thuật toán thuộc hệ mã công khai, vẫn còn cồng kềnh so với thuật toán hệ bí mật. Vì vậy, hệ mật chỉ thích hợp cho các ứng dụng có kích thước nhỏ.

3.Hướng phát triển

- Ứng dụng chữ ký số, đảm bảo tính toàn vẹn của dữ liệu
- Thay thế Affine bằng hệ mã đối xứng khác an toàn hơn như (AES, DES)

DANH MỤC TÀI LIỆU THAM KHẢO

[1]. Nguyễn Bình, Giáo trình Mật mã học, Học viện Công nghệ Bưu chính Viễn thông. [tham khảo tháng 7/2019]

[2]. Nguyễn Bình, Nguyễn Minh Trung, Some hybrid cryoto-systems contructed on DLP,ATC_14,2014. [tham khảo tháng 7/2019]

[3]. Tạp chí khoa học và công nghệ, tập 44,số 2, 2006 . [tham khảo tháng 7/2019]

[4] Man Toung Rhee, Hanyang University, Cryptography and Secure Communications, McGRAW - HILL BOOK CO, 1994. . [tham khảo tháng 7/2019]

[5] Ph.D William Stallings, Network and Internetwork Security Principles and Practice, PRENTICE HALL, 1995. [tham khảo tháng 7/2019]

[6]. Cryptography: Theory and Practice, Third Edition - Douglas R. Stinson [tham khảo tháng 7/2019]

[7]. ElGamal T (1985) A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans Inf Theory [tham khảo tháng 7/2019]

[8]. A.E .Okeyinka (2015); Computational Speeds Analysis of RSA and ElGamal Algorithms on Text Data; World Congress on Engineering and Computer Science. [tham khảo tháng 7/2019]

[9]. https://en.wikipedia.org/wiki/Affine_cipher [tham khảo tháng 7/2019]

[10].https://en.wikipedia.org/wiki/ElGamal_encryption#The_algorithm [tham khảo tháng 7/2019]