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

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2019

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: **GS.TS. NGUYỄN BÌNH**

Phản biện 1: ……

Phản biện 2: ……

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ..... giờ ..... ngày ..... tháng ..... năm

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông

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:

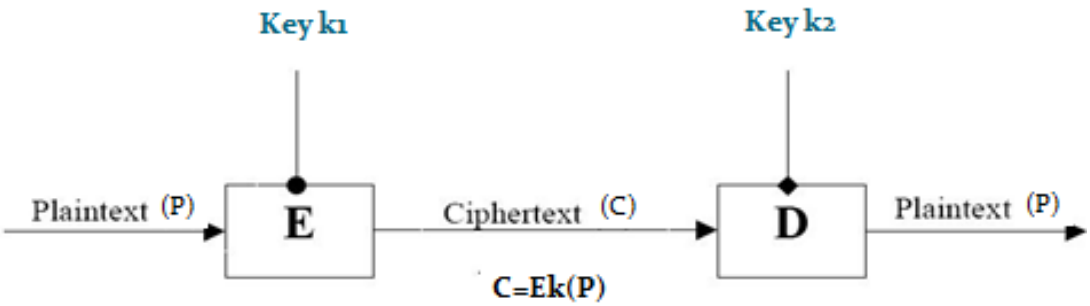
- **P** (Plaintext): Tập hợp hữu hạn các bản rõ có thể chưa được mã hóa
- **C** (Ciphertext): Tập hợp hữu hạn các bản rõ có thể đã mã hóa.
- **K** (Key): Tập hợp các bản khoá mã hóa, khóa giải mã có thể.
- **E** (Encrytion): Tập hợp các qui tắc mã hoá có thể.
- **D** (Decrytion): Tập hợp các qui tắc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

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 $d_k \in D$

$d_k : C \rightarrow P$ 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 cổ điển** (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…

- **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.

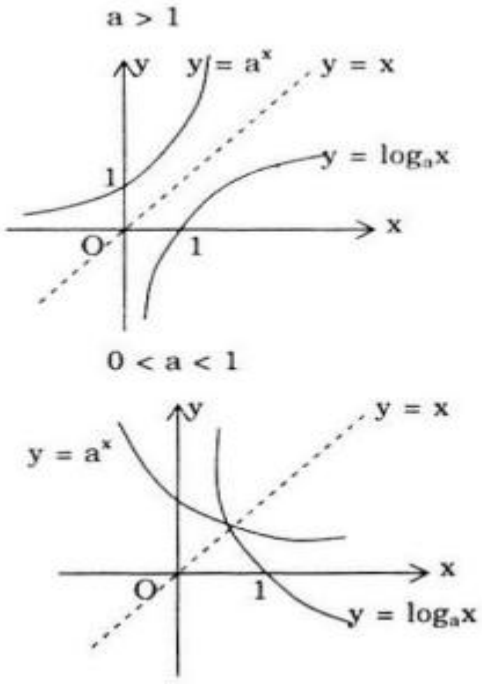
1.2 Giới thiệu bài toán Lôgarit rời rạc

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:

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

- ***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.

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

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

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 $c \equiv a^H (\text{mod } p)$.

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

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

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

$$c^u \equiv b.a^v (\text{mod } p),$$

từ đây $a^{Hu-v} \equiv b(\text{mod } p)$.

Bước 6. Đưa ra giá trị $x \equiv Hu - v(\text{mod } p-1).$

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 $x \equiv Hu - v(\text{mod } p-1)$,

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.

1.2.3.2 Thuật toán Pohlig-Hellman

1.2.3.3 Thuật toán Adleman

1.2.3.4 Thuật toán COS

1.2.3.5 Thuật toán LOGsmooth

1.2.3.6 Thuật toán index-calculus

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:

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

2.1.2 Thám mã mật mã Affine

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 1.1: 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

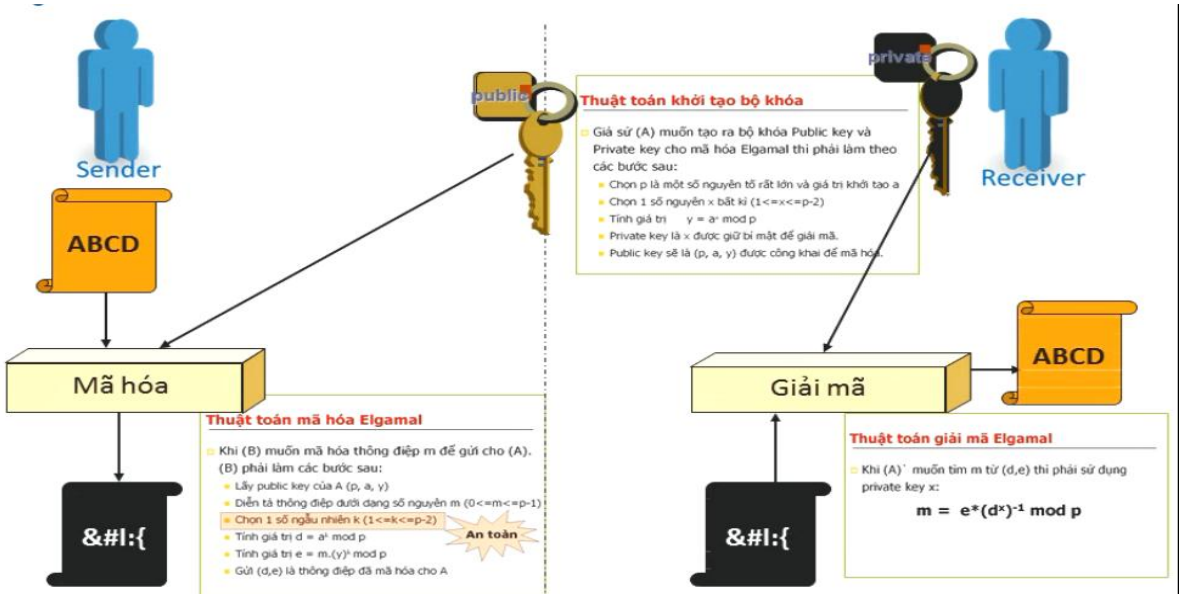
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2.07	8.61	10.42	1.91	0.05	4.77	6.97	8.50	4.16	0.31	2.80	0.04	3.18	0.11

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 Different- 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 α ∈ 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 ∈ 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:

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

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

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

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.

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.

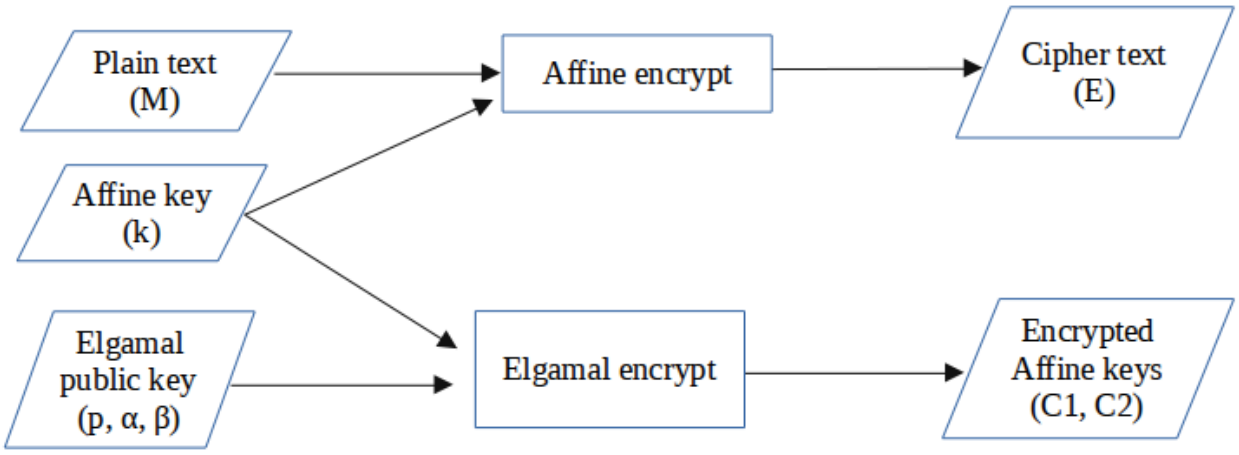
Cụ thể, trong trường hợp này ta sẽ xem xét đến việc phối hợp hệ mật ElGamal và mật mã Affine.

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



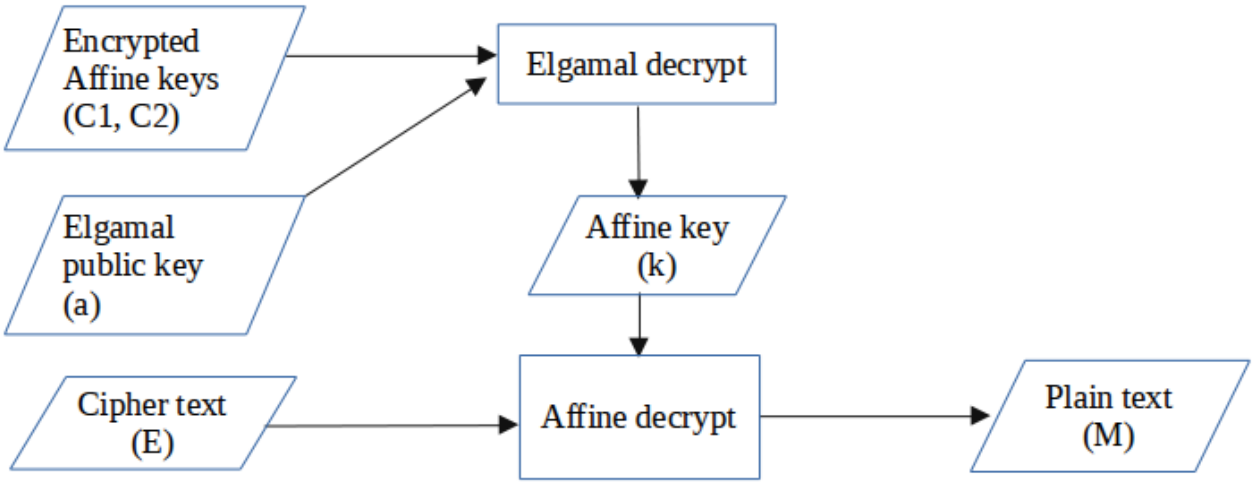
Hình 2.1: Sơ đồ mã hóa Hệ mật Affine – ElGamal

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 2.2: 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)