BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐẠI HỌC ĐÀ NẴNG

LƯƠNG KHÁNH TÝ

TỐI ƯU HÓA GIẢI THUẬT XỬ LÝ SỐ HỌC TRONG HỆ MÃ HÓA RSA

Chuyên ngành : **KHOA HỌC MÁY TÍNH**

Mã số : **60.48.01**

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

Đà Nẵng - Năm 2012

Công trình được hoàn thành tại

ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn khoa học: **PGS.TSKH. TRẦN QUỐC CHIẾN**

Phản biện 1: **PGS.TS. PHAN HUY KHÁNH**

Phản biện 2: **TS. TRƯƠNG CÔNG TUẤN**

Luận văn được bảo vệ tại Hội đồng chấm Luận văn tốt nghiệp thạc sĩ kỹ thuật họp tại Đại học Đà Nẵng vào ngày 03 tháng 03 năm 2012

Có thể tìm hiểu luận văn tại:

- Trung tâm Thông tin - Học liệu, Đại học Đà Nẵng
- Trung tâm Học liệu, Đại học Đà Nẵng

MỞ ĐẦU

1. Lý do chọn đề tài

Trong hầu hết lịch sử mật mã học, khóa dùng trong các quá trình mã hóa và giải mã phải được giữ bí mật và cần được trao đổi bằng một phương pháp an toàn khác (không dùng mật mã) như gặp nhau trực tiếp hay thông qua một người đưa thư tin cậy. Vì vậy quá trình phân phối khóa trong thực tế gặp rất nhiều khó khăn, đặc biệt là khi số lượng người sử dụng rất lớn. Mật mã hóa khóa công khai đã giải quyết được vấn đề này vì nó cho phép người dùng gửi thông tin mật trên đường truyền không an toàn mà không cần thỏa thuận khóa từ trước.

Trong mật mã học, RSA là một thuật toán mật mã hóa khóa công khai. Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa.Nó đánh dấu một sự tiến bộ vượt bậc của lĩnh vực mật mã học trong việc sử dụng khóa công cộng. RSA đang được sử dụng phổ biến trong thương mại điện tử và được cho là đảm bảo an toàn với điều kiện độ dài khóa đủ lớn.

Hệ mã RSA thực hiện tính toán với số nguyên lớn, có thể lên tới hàng trăm chữ số.Độ phức tạp của việc giải mã của hệ mã này tỉ lệ thuận với độ lớn của các số nguyên tham gia vào việc tạo khóa mã hóa và khóa công khai. Vì vậy, để hệ mã được an toàn cần tăng kích thước của số nguyên. Vấn đề tăng kích thước của số nguyên sẽ dẫn đến thời gian xử lý chương trình mã hóa cũng tăng lên. Mặt khác thông tin mã hóa ngày càng đa dạng và có khối lượng lớn đòi hỏi hệ mã giảm thiểu thời gian xử lý.

Bên cạnh đó, do ngày càng có nhiều công cụ, phần mềm hỗ trợ nhằm tìm cách bẻ khóa để lấy cắp các thông tin vì thế hệ mã cần được nâng cấp tính bảo mật.

Đó là những lý do mà tôi chọn nghiên cứu và thực hiện đề tài **"Tối ưu hóa giải thuật xử lý số học trong hệ mã hóa RSA"**dưới sự hướng dẫn của thầy giáo PGS.TSKH. Trần Quốc Chiến.

2. Mục đích nghiên cứu

Mục tiêu của đề tài là nghiên cứu lý thuyết về hệ mật mã hóa công khai RSA, xây dựng thuật toán tối ưu hóa nhằm tăng hiệu quả các phép tính toán với số nguyên lớn, từ đó tăng tốc độ xử lý, tính bảo mật của hệ mã và thực hiện mã hóa – giải mã các tập tin văn bản.

3. Đối tượng và phạm vinghiên cứu

**** Đối tượng nghiên cứu***

Nghiên cứu lý thuyết cơ bản về hệ mã hóa công khai, đặc biệt hệ mã hóa RSA là đối tượng nghiên cứu chính của đề tài nhằm phát hiện các phép toán xử lý số học cần tối ưu.Từ đó, bước đầu được thử nghiệm hệ mã hóa RSA cho kết quả tối ưu hóa.

**** Phạm vi nghiên cứu***

Trong phạm vi nghiên cứu của đề tài này, tác giả thực hiện tối ưu hóa với một số phép toán số nguyên lớn và xây dựng ứng dụng mã hóa - giải mã tập tin văn bản.

Đề tài còn trong phạm vi đưa ra giải pháp, vì vậy để ứng dụng vào thực tiễn cần có nhiều thời gian hơn nữa.

4. Phương pháp nghiên cứu

- Thu thập và phân tích các tài liệu sơ cấp, tài liệu trên Internet liên quan đến đề tài.

- Thảo luận, lựa chọn hướng giải quyết vấn đề.

- Tìm hiểu các thuật toán xử lý số nguyên lớn của hệ mã hóa công khai RSA.

- Tối ưu hóa các phép toán xử lý số học của hệ mã RSA làm tăng khả năng xử lý ở từng bước.

- Thực nghiệm cài đặt ứng dụng để đánh giá và so sánh kết quả trước và sau khi tối ưu hóa.

5. Ý nghĩa khoa học và thực tiễn

**** Ý nghĩa khoa học***

Kết quả nghiên cứu có thể làm tài liệu tham khảo cho việc phân tích các thuật toán của hệ mã hóa RSA.

Phần nghiên cứu lý thuyết sẽ đưa ra một cách nhìn tổng quát về mã hóa công khai và vấn đề tối ưu hóa phép toán xử lý số học với số nguyên lớn trong hệ mã RSA.

**** Ý nghĩa thực tiễn***

Cài đặt thử nghiệm các phép tính toán với số nguyên có giá trị lớn và sử dụng thuật toán tối ưu hóa xây dựng ứng dụng mã hóa – giải mã các tập tin văn bản.

6. Cấu trúc của luận văn

Ngoài phần mở đầu, kết luận và tài liệu tham khảo trong luận văn gồm có các chương như sau :

Chương 1 : Lý thuyết và thực tiễn mã hoá dữ liệu

Chương 2 : Phân tích cơ chế hoạt độngcủa hệ mã với khóa công khai

Chương 3 : Tối ưu hóa giải thuật xử lý số học và cài đặt thử nghiệm hệ mật mã RSA

CHƯƠNG 1 - LÝ THUYẾT VÀ THỰC TIỄN VỀ MÃ HÓA DỮ LIỆU

1.1 KHÁI NIỆM VỀ MÃ HÓA DỮ LIỆU

1.1.1 Lịch sử phát triển

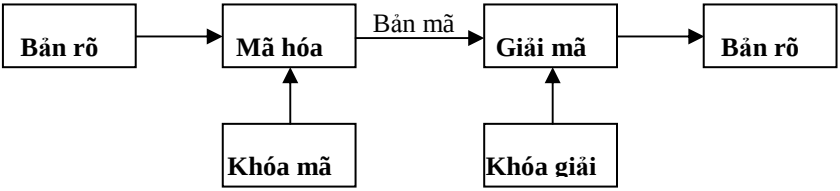
1.1.2 Khái niệm chung về mật mã

1.1.3 Những yêu cầu đối với hệ mật mã hiện đại

1.1.4 Các phương pháp mã hóa

1.1.4.1 Hệ thống mã hóa đối xứng

1.1.4.2 Hệ thống mã hóa bất đối xứng



Hình 1.2: Sơ đồ hoạt động của mã hóa khóa bất đối xứng

1.2 KHÁI NIỆM ĐỘ PHỨC TẠP THUẬT TOÁN

1.2.1 Một số định nghĩa

Định nghĩa 1.1:

Định nghĩa 1.2:

Một thuật toán được gọi là có độ phức tạp đa thức, hoặc có thời gian đa thức, nếu số các phép tính cần thiết khi thực hiện thuật toán không vượt quá O(P(n)),trong đó P(n) là đa thức bậc cao (từ 2 trở lên).

Các thuật toán với thời gian $O(\alpha^n)$, trong đó $\alpha > 1$, được gọi là các thuật toán với độ phức tạp mũ, hoặc thời gian mũ.

Tồn tại những thuật toán có độ phức tạp trung giangiữa đa thức và mũ.Các thuật toán đó được gọi là thuật toán dưới mũ.

Bảng 1.1: Bảng chi phí thời gian phân tích số nguyên n ra thừa số nguyên tố

Số chữ số thập phân	Số phép tính trên bit	Thời gian
50	$1,4.10^{10}$	3,9 giờ
75	$9,0.10^{12}$	104 ngày
100	$2,3.10^{15}$	74 năm
200	$1,2.10^{23}$	$3,8.10^{9}$ năm
300	$1,5.10^{29}$	$4,9.10^{15}$ năm
500	$1,3.10^{39}$	$4,2.10^{25}$ năm

1.2.2 Các bài toán khó tính toán và ứng dụng trong mật mã học

Định nghĩa 1.3:

Định nghĩa 1.4:

Ví dụ về hàm một chiều:

- Với N = p*q, trong đó p và q là các số nguyên tố lớn, khi đó ta dễ dàng tìm được N khi biết p và q, nhưng nếu cho trước số N thì khó mà tìm được 2 số p và q.

- $f_{g,N} : x \rightarrow g^x \bmod N$ là hàm một chiều. Thật vậy, phép tính $g^x \bmod N$ có độphức tạp đa thức; nhưng tính f^{-1} lại là bài toán cực khó (bài toán logarithm rời rạc).

1.3 CƠ SỞ TOÁN HỌC CỦA MẬT MÃ HỌC

1.3.1 Một số định nghĩa

1.3.2 Lý thuyết đồng dư thức

Định nghĩa 1.5: Cho a và b là các số nguyên, a được gọi là đồng dư với b theo modulo n, ký hiệu là a ≡ b (mod n) nếu n chia hết (a-b). Số nguyên n được gọi là modulo của đồng dư.

1.3.3 Hàm phi Euler

Định nghĩa 1.6

Cho n ≥1, đặt φ(n) là tập các số nguyên trong khoảng [1, n] nguyên tố cùng nhau với n. Hàm φ như thế được gọi là hàm phi Euler.

Tính chất của hàm phi Euler:

1. Nếu p là số nguyên tố thì φ(p) = p – 1

2. Hàm phi Euler là hàm có tính nhân: Nếu gcd(m,n) = 1 thì φ(m.n) =φ(m).φ(n). (trong đó gcd(m, n) là ký hiệu ước số chung lớn nhất của m và n)

3. Nếu $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$ trong đó p_1, p_2,…p_k là các thừa số nguyên tố của n, e_i (i=1...k) là dạng biến đổi số mũ của n thì:

$$\varphi(n) = (p_1 - 1)p_1^{e_1 - 1}.(p_2 - 1)p_2^{e_2 - 1} \dots (p_k - 1)p_k^{e_k - 1}$$

Công thức này là một tích Euler và thường được viết là

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

với tích chạy qua các số nguyên tố p là ước của n.

Ví dụ:

1. φ(7) = 7 – 1 = 6 (7 là số nguyên tố)

2. gcd(3,5) = 1 thì φ(3.5) = φ(3) φ(5) = (3-1).(5-1) = 8

3. $\varphi(36) = (3^2 2^2) = 36\left(1 - \frac{1}{3}\right)\left(1 - \frac{1}{2}\right) = 36.\frac{2}{3}.\frac{1}{2} = 12$

Định lý 1.1 (Euler)

1.3.4 Không gian Z_n

** Các định nghĩa trong không gian Z_n*

** Các tính chất trong không gian Z_n*

1.3.5 Nhóm nhân Z^*_n

1.3.6 Thặng dư

Định nghĩa 1.7

1.3.7 Các thuật toán trong Z_n

** Thuật toán tính nghịch đảo nhân trong Z_n*

Bài toán phát biểu như sau: Cho a $\in$ Z_n, hãy tìm a^{-1} mod n nếu có.

Bước đầu, dùng thuật toán Euclid mở rộng sau để tìm các số nguyên x và y sao cho:

$$ax + ny = d \text{ với } d = \gcd(a,n).$$

Nếu d > 1 thì a^{-1} mod n không tồn tại.Ngược lại, return (x).

Thuật toán Euclid mở rộng(N^+={1,2,3,…,}

Algorithm Euclid

INPUT: a, b $\in$ N^+; //N^+*là tập các số nguyên dương*

OUTPUT: x, y $\in$ Z thỏa ax + by = gcd(a, b)

Method

x_0=1 : x_1=0 : y_0=0 : y_1=1;

While b>0

r= a mod b;

if r=0 then Exit While;

q= a / b;x= x_0-x_1*q;y= y_0-y_1*q;

a=b;b=r;x_0=x_1;x_1=x;y_0=y_1;y_1=y;

endwhile;

return (x, y);

end.

*Ví dụ:*Với a=29, b=8, giải thuật trải qua các bước như sau

- Bước *i*	r_i	r_{i+1}	r_{i+2}	q_{i+1}	x_i	x_{i+1}	x_{i+2}	y_i	y_{i+1}	y_{i+2}
0	29	8	5	3	1	0	1	0	1	-3
1	8	5	3	1	0	1	-1	1	-3	4
2	5	3	2	1	1	-1	2	-3	4	-7
3	3	2	1	1	-1	2	-3	4	-7	**11**
4	2	1	0	2						

Từ kết quả trên ta có x = -3, y = 11

1.3.8 Thuật toán kiểm tra tính nguyên tố

CHƯƠNG 2 – PHÂN TÍCH CƠ CHẾ HOẠT ĐỘNG CỦA HỆ MÃ VỚI KHÓA CÔNG KHAI

2.1 HỆ THỐNG MÃ VỚI KHÓA CÔNG KHAI

2.1.1 Giới thiệu chung

2.1.2 Một số quan điểm của hệ mật mã với khóa công khai

2.1.3 Nguyên lý hoạt động của hệ mật mã với khóa công khai

Đối với hệthống mã hóa khóa công khai: Mỗi người sử dụng phải tạo riêng cho mình một cặp khóa. Trong đó, một khóa công khai (public key) cùng với thuật toán mã hóa E, được công bố rộng rãi tại thư mục dùng chung cho mọi người sử dụng. Còn lại là khóa riêng (private key) cùng với thuật toán giải mã D được giữ bí mật bởi người sử dụng.

Như vậy, người A muốn gửi thông điệp R đến cho người B:

Giả sử:

Khóa công khai của B là: K_B, Khóa riêng của B là: M_B

Khóa công khai của A là: K_A , Khóa riêng của A là: M_A

Thuật toán mã hóa: E, thuật toán giải mã: D

Người A tìm khóa công khai K_B của người B trong thư mục dùng chung và tính $C = E(K_B, R)$, sau đó gửi bản mã C cho người B.

Khi nhận bản mã C người B sẽ giải mã dựa vào khóa riêng M_B của mình để tính $R = D(M_B, C)$.

2.1.4 Các yêu cầu của hệ mật mã với khóa công khai

2.2 HỆ MẬT MÃ KHÓA CÔNG KHAI RSA

2.2.1 Bài toán phân tích số nguyên

Bài toán 2.1 (Bài toán phân tích số nguyên):

Bài toán 2.2 (Bài toán RSA):

Cho số nguyên dương N, N=p*q với p và q là các số nguyên tố phân biệt, số nguyên e sao cho thỏa mãn gcd(e, (p – 1) * (q – 1)) = 1, và số nguyên c. Tìm một số nguyên m sao cho $m^e \equiv c \pmod N$.

Bài toán RSA cũng có độ khó tương tự như bài toán phân tích số nguyên, nhưng nó dễ dàng được giải nếu như biết được hai số nguyên tố p và q.

2.2.2 Quá trình tạo khoá, mã hoá và giải mã

2.2.2.1 Tạo khóa:

- Tạo hai số nguyên tố phân biệt p và q lớn, sao cho bài toán phân tích thật sự là khó giải (kích cỡ mỗi số khoảng 512 bits → 1024 bits).

- Tính N = p* q và φ(N) = (p – 1) * (q – 1).

- Chọn một số nguyên ngẫu nhiên e sao cho 1< e <φ(N) và gcd(e, φ(N)) = 1

- Sử dụng thuật toán Euclid mở rộng, để tính số nguyên d duy nhất, sao cho 0 < d <φ(N) và $e * d \equiv 1 \bmod \varphi(N)$ (d là nghịch đảo của e modulo φ(N))

- Hai số (e, N) làm khóa công khai, còn (d, N) được giữ bí mật làm khóa riêng.

Các số nguyên tố p, q sẽ bị xóa khi kết thúc quá trình tạo khóa.

2.2.2.2 Mã hóa:

Giả sử để gửi thông điệp M cho người B. Người A thực hiện như sau:

- Lấy khóa công khai của người nhận B: (e, N).

- Biến đổi thông điệp M thành những số nguyên M_i tương ứng sao cho M_i< N, (i = 1,…, k). Theo phép biến đổi sau:

- Biến đổi các ký tự trong thông điệp M thành các số nguyên tương ứng, thí dụ theo qui tắc: Dấu cách → 00, A → 01, B → 02,..., Z → 26.

- Chia thông điệp vừa biến đổi thành k nhóm có chiều dài bằng nhau, mỗi nhóm biểu diễn một số nguyên $M_i \in \{0,\ldots, N-1\}$ (với $1 \leq i \leq k$).

- Thực hiện mã hóa lần lượt cho từng số $M_i \rightarrow C_i$ bằng cách:

$C_i = M_i^e \pmod N$.

Tập các số nguyên $\{C_1, C_2,...,C_k\}$ là bản mã để gửi đến người nhận B.

2.2.2.3 Giải mã:

Người nhận B thực hiện các bước sau:

- Thực hiện giải mã lần lượt từng số nguyên $C_i \rightarrow M_i$ bằng cách:

$M_i = D(C_i) = C_i^d \pmod N$ với $0 \leq M_i < N$, (d là khoá bí mật của B).

- Thực hiện phép biến đổi ngược lại từ các số M_i thành các chuỗi ký tự tương ứng để khôi phục lại nội dung thông điệp M ban đầu.

2.2.3 Tính đúng của quá trình giải mã

Ví dụ 2.1: Minh họa của hệ mật mã RSA

+ *Tạo khóa:*

Chọn p và q là những số nguyên tố nhỏ với mục đích minh họa

- Chọn hai số nguyên tố p = 41, q = 67;

- Tính N = 41 * 67 = 2747 và φ(N) = (41- 1) * (67-1) = 2640;

- Chọn e = 59 thỏa mãn gcd(e, φ(N)) = gcd(2640, 59) = 1;

- Do gcd(2640, 59) = 1 nên áp dụng thuật toán Euclid mở rộng tìm phần tử nghịch đảo d = 179 (d là phần tử nghịch đảo của e modulo φ(N)).

- Công bố khóa công khai là cặp số(e = 59, N = 2747), khóa bí mật là (d,p,q) = (179,41,67)

+ Mã hóa:

Giả sử nội dung cần mã hoá là M = “MA HOA CONG KHAI ”

- Biến đổi các ký tự của thông điệp thành các số tương ứng như sau:

M	A	_	H	O	A	_	C	O	N	G	_
13	01	00	08	15	01	00	03	15	14	07	00
K	H	A	I								
11	08	01	09								

- Chia thông điệp thành 8 khối, mỗi khối gồm 4 chữ số biểu diễn một số nguyên $M_i<N$, với Mi $\in$ {1301;0008;1501;0003;1514;0700;1108;0109}.

- Mã hóa lần lượt từng số M_i : $C_i = M_i^{59}$ (mod 2747) (8)

Mã hóa số đầu tiên M_1 = 1301 theo cách tính (8) ta có:

$C_1 = M_1^{59} \bmod 2747 \equiv 1301^{59} \bmod 2747 = 2352$.

Tiếp tục tính các số $C_2,...,C_8$ từ các số $M_2,...,M_8$ theo (8). Ta có được kết quả ở cột C_i là bản mã để gửi đến người nhận:

Khối	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$\mathbf{M_i}$	1301	0008	1501	0003	1514	0700	1108	0109
$\mathbf{C_i=E(M_i)}$	2352	2537	1745	2733	1203	2651	0534	0454

+ Giải mã:

- Thực hiện giải mã lần lwợt từng số ở cột C_i ($1 \le i \le 8$)

$M_i = D_{kd}(C_i) \equiv C_i^{179}$ (mod 2747) (9)

Giải mã số đầu tiên C_1 = 2352 theo cách tính (9) ta có:

$M_1 = C_1^{179} \bmod 2747 = 2352^{179} \bmod 2747 = 1301$

Tiếp tục tính các số $M_2,...,M_8$ từ các số $C_2,...,C_8$theo (9) ta có bảng minh họa các số M_i được giải mã từ các số C_i như sau:

Khối	1	2	3	4	5	6	7	8
$C_i=E(M_i)$	2352	2537	1745	2733	1203	2651	0534	0454
M_i	1301	0008	1501	0003	1514	0700	1108	0109

- Thực hiện phép biến đổi ngược từ các số M_i thành các chuỗi ký tự tương ứng để khôi phục lại thông điệp gốc ban đầu M = "MA HOA CONG KHAI".

2.2.4 Chi phí thực hiện trong quá trình mã hóa và giải mã

- Chi phí cho quá trình mã hoá:

Tính $C = M^e \bmod N$, với số mũ e thường được chọn có dạng $e = 2^{16} – 1$ Như vậy, tổng chi phí của quá trình mã hóa là hàm mũ.

- Chi phí cho quá trình giải mã:

Quá trình giải mã của hệ RSA, chỉ thực hiện phép tính $M = C^d \bmod N$, với số mũ bí mật d thường rất lớn (d ≈ N) để đảm bảo độ an toàn cho dữ liệu. Vì vậy, chi phí thực hiện giải mã của hệ RSA tương đương với chi phí để thực hiện hàm mũ.

2.2.5 Đánh giá hệ mật mã khóa công khai RSA

2.2.5.1 Độ an toàn

HệRSA được thiết kế dựa trên độkhó giải bài toán phân tích ra thừa số nguyên tố. Hầu hết các phương pháp thám mã hệ RSA như tìm các thừa số p và q, tìm φ(n), hay tìm khóa riêng d… đều khó như bài toán phân tích.

Sau đây,ta có bảng chi phí thời gian cần thiết để phân tích những hệ mật mã RSA có kích cỡ số modulo N khác nhau, bằng những thuật toán phân tích tốt nhất hiện nay. Ở đây chi phí tính toán được tính bằng đơn vị MIPS-Years (đó là số các phép tính đã hoàn thành bởi một máy trong thời gian một năm, với tốc độ khoảng 10^6 phép tính trên một giây, ta có 1MIPS-Years ≈ 2^{45} phép tính).

Bảng 2.2: Bảng chi phí thời gian cần thiết để phân tích các số nguyên N

Hệ RSA	Số chữ số thập phân	Số bit	Thuật toán	Năm	Chi phí phân tích (MIPS-Years)
RSA-129	129	462	MPQS	1994	5000
RSA-130	130	430	GNFS	1996	1000
RSA-140	140	465	GNFS	1999	2000
RSA-155	155	512	GNFS	1999	8000
RSA-576	174	576	GNFS	2003	13000

2.2.5.2 Hiệu suất thực hiện và ứng dụng

2.3 HỆ MẬT MÃ RSA WITH CRT

2.3.1 Định lý đồng dư Trung Hoa

2.3.2 Thuật toán Garner

2.3.3 Các quá trình tạo khoá, mã hoá và giải mã

2.4 PHÂN TÍCH CƠ CHẾ HOẠT ĐỘNG CỦA HỆ MÃ RSA

2.4.1 Phân tích quá trình tạo khóa:

- Tạo hai số nguyên tố phân biệt p và q lớn, sao cho bài toán phân tích thật sự là khó giải.

- Tính N = p* q vàφ(N) = (p – 1) * (q – 1).

- Chọn một số nguyên ngẫu nhiên e sao cho 1< e <φ(N) thỏa mãn gcd(e, φ(N)) = 1

- Sử dụng thuật toán Euclid mở rộng, để tính số nguyên d duy nhất, sao cho 0 < d <φ(N) và e * d ≡ 1 mod φ(N) (d là nghịch đảo của e modulo N)

- Hai số (e, N) làm khóa công khai, còn (d, N) được giữ bí mật làm khóa riêng.

Các số nguyên tố p, q sẽ bị xóa khi kết thúc quá trình tạo khóa.

Như vậy, mấu chốt để tăng tính an toàn của hệ mã RSA là ta cần thực hiện được quá trình mã hóa xuất phát từ các số nguyên tố p, q lớn.

2.4.2 Phân tích quá trình mã hóa:

Giả sử để gửi thông điệp M cho người B. Người A thực hiện:

- Lấy khóa công khai của người nhận B: (e, N).

- Biến đổi thông điệp M thành những số nguyên M_i tương ứng sao cho $M_i < N$, (i = 1,…, k). Theo phép biến đổi sau:

- Biến đổi ký tự trong thông điệp M thành các số nguyên tương ứng, thí dụ theo qui tắc: Dấu cách↔00, A↔01, B ↔02,.., Z ↔ 26.

- Chia thông điệp vừa biến đổi thành k nhóm có chiều dài bằng nhau, mỗi nhóm biểu diễn một số nguyên $M_i \in \{0,\ldots, N-1\}$ (với $1 \le i \le k$).

- Thực hiện mã hóa lần lượt cho từng số $M_i \rightarrow C_i$ bằng cách:

$$C_i = M_i^e \pmod N.$$

Tập $\{C_1, C_2,...,C_k\}$ là bản mã để gửi đến người nhận B.

Ta thấy rằng quá trình mã hóa phải thực hiện liên tiếp việc mã hóa các số $M_i$theo công thức: $C_i = M_i^e \pmod N$.

Khi p và q lớn thì ta có N = p*q rất lớn.

Trên lý thuyết, số e có thể chọn chỉ cần thỏa mãn gcd(e, φ(N)) = 1, tuy nhiên để tăng tính an toàn, số e thường được sẽ là số lớn hơn Max(p,q) với Max(p,q) trả về số lớn nhất giữa p và q.

Do đó, quá trình mã hóa sẽ thực hiện với các số rất lớn nhưng vẫn phải đảm bảo thời gian thực hiện việc mã hóa là đủ tốt.

Xuất phát từ các lí do trên, ta cần tác động vào quá trình mã hóa bằng các thuật toán tốt để có thể thỏa mãn các yêu cầu trên.

2.4.3 Phân tích quá trình giải mã:

Người nhận B thực hiện các bước sau:

- Thực hiện giải mã lần lượt từng số nguyên $C_i \rightarrow M_i$ bằng cách:

$M_i = D(C_i) = C_i^d \pmod N$ với $0 \leq M_i < N$, (d là khoá bí mật của B).

- Thực hiện phép biến đổi ngược lại từ các số M_i thành các chuỗi ký tự tương ứng để khôi phục lại nội dung thông điệp M ban đầu.

Quá trình giải mã cũng phải thực hiện việc tính toán liên tiếp để tìm M_i theocông thức: $M_i = D(C_i) = C_i^d \pmod N$, quá trình này cũng thực hiện trên các số lớn vì ta có d là số lớn. Do đó, quá trình giải mã cũng cần có các tác động để đảm bảo thời gian giải mã là chấp nhận được. Điều này có ý nghĩa rất quan trọng vì hệ mã RSA có số lượng phép tính lớn, bên cạnh đó để có thể thực hiện với các bản rõ có nội dung lớn thì ta phải giải quyết được vấn đề này.

Kết luận: Trong thực tế, tốc độ mã hóa và giải mã của RSA chậm so với các hệ mã khác.Điều này dẫn đến việc RSA chủ yếu được dùng để mã hóa khóa bí mật hoặc các các bản rõ ngắn.Phần nội dung chính cần gửi sẽ được mã hóa bằng một phương pháp mã hóa khác có tốc độ thực hiện nhanh hơn (Phương pháp này thường kém an toàn hơn so với RSA). Người nhận sẽ giải mã bằng RSA để lấy khóa bí mật rồi mới tiến hành giải mã nội dung cần nhận.

2.5 KHẢ NĂNG BỊ BẺ KHÓA CỦA HỆ MÃ CÔNG KHAI RSA

2.5.1 Một số phương pháp tấn công hệ mã RSA.

2.5.2 Độ an toàn của hệ mã RSA.

2.6 HỆ MẬT MÃ KHÓA CÔNG KHAI ELGAMAL

2.6.1 Bài toán logarithm rời rạc

2.6.2 Định nghĩa các tập làm việc của hệ mật mã ElGamal

2.6.3 Quá trình tạo khoá, mã hoá, giải mã

2.6.4 Đánh giá độ an toàn và khả năng ứng dụng của hệ mật mã khóa công khai ElGamal

CHƯƠNG 3 –TỐI ƯU HÓA GIẢI THUẬT XỬ LÝ SỐ HỌC VÀ CÀI ĐẶT THỬ NGHIỆM HỆ MẬT MÃ RSA

3.1 PHÂN TÍCH CÁC THUẬT TOÁN XỬ LÝ SỐ HỌC CỦA RSA

Trong quá trình tạo khóa ta cần phải tạo ra hai số nguyên tố phân biệt p và q lớn, sao cho bài toán phân tích thật sự là khó giải.Như vậy, để đảm bảo an toàn cho hệ mã RSA, giải thuật xử lý phải thực hiện được với các số lớn hàng trăm chữ số.

3.1.1 Phép nhân, cộng và trừ

Trong quá trình tạo khóa, ta phải tính được N = p*q và φ(N) = (p–1)*(q–1). Do đó chương trình cần xửlý được phép nhân hai số p và q với p và q là các số nguyên tố giá trị lớn.

3.1.2 Phép lũy thừa

Quá trình giải mã của RSA cần tínhM = C^dmod N, với số mũ bí mật d thường rất lớn (d N) để đảm bảo độ an toàn cho dữ liệu. Vì vậy chi phí thực hiện giải mã của hệ RSA tương đương với chi phí để thực hiện phép tính lũy thừa nhanh là hàm mũ.

Do đó chương trình cần phải xử lý được các phép tính lũy thừa nhanh bằng cách: đưa phép lũy thừa về dạng phép nhân liên tiếp sử dụng các tính chất đồng dư. Vì thế, đề tài sử dụng giải thuật Fast Fourier Transform để giải quyết vấn đề này.

3.2 ỨNG DỤNG GIẢI THUẬT FAST FOURIER TRANSFORM TRONG XỬ LÝ PHÉP NHÂN

3.2.1 Giới thiệu thuật toán FFT

Cho hai số lớn X và Y với kích thước lớn nhất là n-1, chúng có thể được viết ở dạng sau:

X = P(B), Y = Q(B)

Với B là cơ số (Thông thường B=10 hoặc là lũy thừa của 10). P và Q là hai đa thức:

$$P(z) = \sum_{j=0}^{n-1} x_j z^j, \qquad Q(z) = \sum_{j=0}^{n-1} y_j z^j$$

Kết quả khi thực hiện nhân P(z) với Q(z), khi đó ta có được tích X.Y=R(B). Bằng cách sắp xếp lại theo hệ số ta thu được kết quả của phép nhân X với Y.

Dựa trên lý thuyết này, ta áp dụng đểthực hiện việc nhân hai đa thức có bậc nhỏ hơn n. Một đa thức có bậc nhỏ hơn m là duy nhất khi nó được tạo bởi các giá trị cụ thể tại m điểm. Do đó, để tính tích R(z)= P(z).Q(z) ta cần đi tính các giá trị R(w_k) tại 2n điểm phân biệt ứng với mỗi w_k, điều này cũng có nghĩa là ta cần tính các giá trị của P(w_k) và Q(w_k). Thuật toán FFT là một gợi ý phù hợp với việc lựa chọn cho w_k căn của đơn vị.

$w_k = e^{\frac{2ik\pi}{n}} = \omega^k$ là căn phức bậc n của 1, trong đó k = 0,1,…,2n-1

Với cách lựa chọn này, các giá trị w_k thỏa mãn hai thuộc tính sau:

- Tập hợp các giá trị (P(w_0),...,P(w_{2n-1})) và (Q(w_0),…,Q(w_{2n-1})) có thể tính toán được trong thời gian O(n logn).

- Từ các giá trị R(w_k), đa thức R(z) thu được trong thời gian O(n logn).

Khi đó, hệ số thứ k kí hiệu là r_k của R(z) thỏa mãn:

$$r_k = \frac{1}{2n} T(w_k), \qquad T(z) = \sum_{j=0}^{2n-1} R(w_j) z^j$$

3.2.2 Biến đổi Fourier

Cho dãy A = (a_0, a_1,…, a_{2n-1}), tìm dạng biến đổi Fourier của dãy A.

Gọi $F_{2n}(A)$ là dạng biến đổi Fourier của A, khi đó $F_{2n}(A)$ được biểu diễn như sau:

$$F_{2n}(A) = (b_0, b_1, \ldots, b_{2n-1}), \qquad b_k = \sum_{j=0}^{2n-1} a_j \omega^{jk} \qquad (*)$$

trong đó k = 0,1,2,…,2n-1

Cuối cùng ta thu được R(z) bằng dạng biến đổi Fourier ngược với các hệ số R(ω^j) được thể hiện trong công thức sau:

$$\overline{F_{2n}}(A) = (b_0, b_1, \ldots, b_{2n-1}), \qquad b_k = \sum_{j=0}^{2n-1} a_j \omega^{-jk}$$

3.2.3 Biến đổi Fourier nhanh

Thuật toán Fast Fourier Transform là phương pháp để tìm ra dạngbiến đổiFourier của dãy số A trong thời gian O(n logn). Cách này nhanh hơn với phương pháp truyền thống cần đến thời gian $O(n^2)$ với n là lũy thừa của 2.

$$b_k = \sum_{j=0}^{2n-1} a_j \omega^{jk} = \sum_{j=0}^{n-1} a_{2j} [(\omega]^2)^{jk} + \omega^k \sum_{j=0}^{n-1} a_{2j+1} [(\omega]^2)^{jk}$$

Nói cách khác, để tính toán các hệ số b_kcủa $F_{2n}(A)$, ta thực hiện các bước sau:

Định nghĩa 2 dãy con kích thước n, A_0 chứa các hệ số chẵn còn A_1 chứa các hệ số lẻ.

$A_0 = (a_0, a_2, \ldots, a_{2n-2})$, và $A_1 = (a_1, a_3, \ldots, a_{2n-1})$

Tìm các biến đổi Fourier:

$C = F_n(A_0) = (c_0, c_1, \ldots, c_{n-1})$ và $D = F_n(A_1) = (d_0, d_1, \ldots, d_{n-1})$.

Từ đó suy ra biểu diễn Fourier của $B = (b_0, b_1, \ldots, b_{2n-1}) = F_{2n}(A)$ theo các công thức sau:

$b_k = c_k + \omega^k d_k$, $b_{n+k} = c_k - \omega^k d_k$, $0 \leq k < n$

3.2.4 Ứng dụng xử lý phép nhân

3.2.4.1 giới thiệu thuật toán

Trong phần này sẽ trình bày thuật toán để thực hiện phép nhân hai số lớn với FFT.

Cho n là lũy thừa của 2 và hai số nguyên lớn X và Y có ít hơn n hệ số khi biểu diễn ở dạng đa thức cơ số B.

X và Y có dạng biểu diễn đa thức như sau:

$$X = \sum_{j=0}^{n-1} x_j B^j \qquad Y = \sum_{j=0}^{n-1} y_j B^j$$

Để tính Z=XY ta thực hiện các bước sau:

- Tìm dạng biến đổi Fourier X^* của X với kích thước 2n:

$X^* = (x_0^*, x_1^*, \ldots, x_{2n-1}^*) \equiv F_{2n}(x_0, x_1, \ldots, x_{n-1}, 0, \ldots, 0)$

- Tìm dạng biến đổi Fourier Y^* của Y với kích thước 2n:

$Y^* = (y_0^*, y_1^*, \ldots, y_{2n-1}^*) \equiv F_{2n}(y_0, y_1, \ldots, y_{n-1}, 0, \ldots, 0)$

- Nhân các phần tử tương ứng của X^* với Y^* rồi lưu kết quả trong Z^*:

$Z^* = (z_0^*, z_1^*, \ldots, z_{2n-1}^*)$ với $z_i^* = x_i^* . y_i^*$

- Biến đổi Fourier ngược để tìm Z từ Z^*:

$$Z = (z_0, z_1, \ldots, z_{2n-1}) \equiv \frac{1}{2n} F_{2n}(Z^*)$$

Đa thức Z chính là kết quả cần tìm của tích XY:

$$Z = \sum_{i=0}^{2n-1} z_i B^i$$

Chú ý rằng X^* là dạng biến đổi Fourier của của dãy $(x_0, x_1, \ldots, x_{n-1})$ với n số 0 được thêm vào sau dãy $(x_0, x_1, \ldots, x_{n-1})$ và Y^* là dạng biến đổi Fourier của của dãy $(y_0, y_1, \ldots, y_{n-1})$ với n số 0 được thêm vào sau dãy $(y_0, y_1, \ldots, y_{n-1})$.

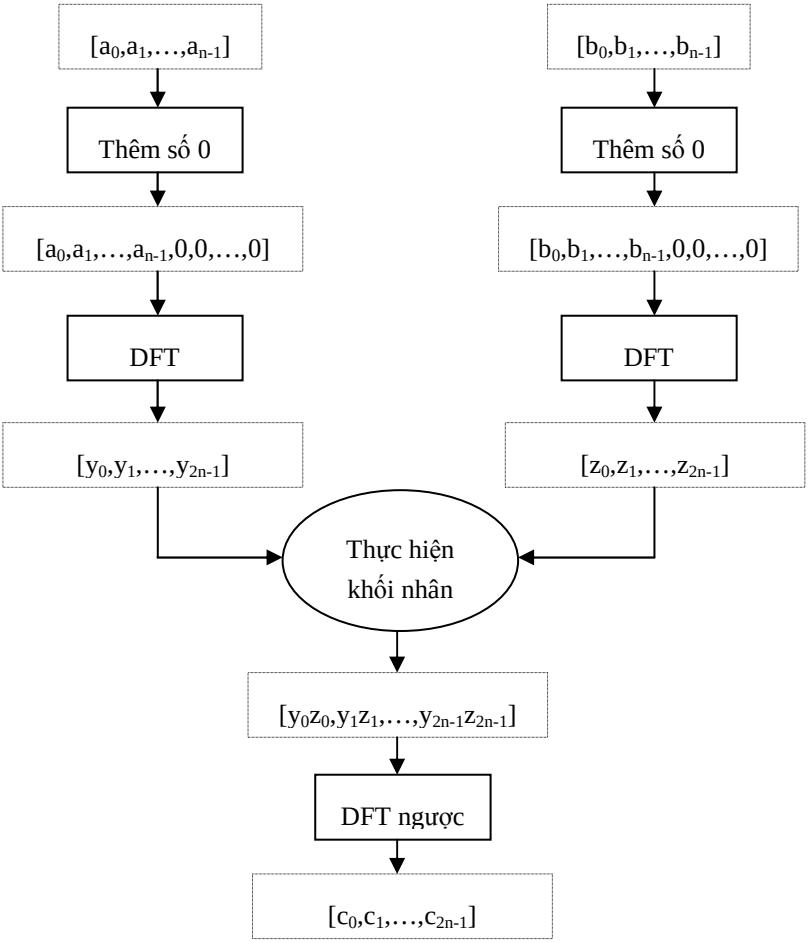
3.2.4.2 Thuật toán DFT

3.2.4.3 Mô phỏng thuật toán FFT

3.2.4.4 Thuật toán FFT

3.2.4.5 Sơ đồ thuật toán

Thực hiện việc nhân nhanh số lớn sử dụng thuật toán FFT áp dụng trong nhân nhanh hai số lớn nhằm biến đổi rời rạc Fourier trong thời gian O(nlogn), kết hợp với Giải thuật Discrete Fourier Transform (DFT):



Sơ đồ 3.1: Thuật toán nhân nhanh sử dụng DFT

3.3 CÀI ĐẶT THỬ NGHIỆM CÁC PHÉP TOÁN

Các thuật toán FFT và DFT được viết dưới dạng các module và được sử dụng trong các quá trình tạo khóa, mã hóa và giải mã trong hệ mã RSA.

3.3.1 Xử lý phép cộng –trừ

Giải thuật cộng – trừ hai số nguyên lớn có thể được thực hiện dễ dàng khi hai số đã được tổ chức dữ liệu để biểu diễn.

Ví dụ 3.1:

Giả sử ta cần cộng hai số a, b đã được biểu diễn ở dạng chuỗi là Sa và Sb. Giải thuật cộng hai số lớn có thể được thực hiện như sau:

Bước 1:

Đưa hai số a, b về cùng độ dài N (Nếu độ dài của a và b là khác nhau) bằng cách thêm các số 0 vào đầu của số có độ dài nhỏ hơn. Ta được Sa và Sb có cùng độ dài N với N =Max[Length(Sa), Length(Sb)]

Bước 2:

Gán giá trị Remem =0 (Remem là biến để nhớ số hàng chục của kết quả (Sa[k] + Sb[k]), đảo ngược Sa và Sb.

Bước 3:

Cộng từng vịtrí tương ứng của hai chuỗi Sa và Sb từ trái sang phải. Lưu kết quả mỗi bước trong chuỗi Sc với Sc[k]= (Sa[k] + Sb[k]+Remem) mod 10,Remem = (Sa[k] + Sb[k]) div 10. Gán Sc[N+1]=Remem.

Bước 4:

Đảo ngược Sc (Chỉ lấy từ vị trí 1 đến vị trí cuối cùng khác 0) ta thu được kết quả trong Sc. (Chiều dài của Sc có thể từ 0 đến N+1) Giải thuật trên có thể áp dụng tương tự cho phép trừ, bên cạnh cách

tổ chức dữ liệu bằng chuỗi (Thường gặp hạn chế vì chiều dài của chuỗi cũng có giới hạn) ta có thể tổ chức dữ liệu sử dụng Stack.

Dễ thấy độ phức tạp của thuật toán này luôn là O(n).

3.3.2 Xử lý phép nhân

Có nhiều giải thuật để thực hiện nhân nhanh hai số lớn.Ở đây ta sử dụng giải thuật nhân nhanh ứng dụng giải thuật Fast Fourier Transform (FFT). Giải thuật nhân nhanh hai số nguyên lớn với N chữ số có thể thực hiện được với độ phức tạp O(n ln(n)ln(ln(n))) khi áp dụng thuật toán FFT.

3.4 XÂY DỰNG HỆ MÃ RSA THỬ NGHIỆM

3.5 NHẬN XÉT VÀ ĐÁNH GIÁ

Thực hiện quá trình thử nghiệm mã hóa và giải mã với các tham số cho kết quả khả thi. Chương trình thử nghiệm với các bộ số p, q lớn và cho kết quả chính xác, thời gian thực hiện nhanh chóng.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

1. KẾT LUẬN

Đề tài bước đầu đưa ra giải pháp để xử lý các phép toán số học với số lớn trong các hệ mã công khai dựa trên cơ sở toán học và tính toán độ an toàn của các hệ mã công khai.

Các kết quả nghiên cứu và ứng dụng bσ¡ớc đầu đã thực hiện được mục đích của đề tài.Bằng việc tối ưu hóa các phép xử lý tính toán phức tạp trong hệ mã công khai và minh chứng trong hệ mã cụ thể RSA.Giải thuật được áp dụng để tối ưu hóa phép nhân là giải thuật xử lý có độ phức tạp nhỏ nhất được biết đến cho tới thời điểmhiện nay.

Chương trình thử nghiệm được xây dựng nhằm chứng minh tính khả thi của các kết quả nghiên cứu.Chương trình hoàn thiện cần có sự đầu tư nhiều hơn về mặt thời gian và công sức.Đề tài có thể tiếp tục phát triển để đem lại ứng dụng đáp ứng được yêu cầu thực tế.

2. HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

Các kết quả của đề tài có thể được áp dụng trong nhiều hệ mã công khai khác nhau và tiếp tục được cải tiến để có được tốc độ thực thi tốt hơn.

Các kết quả có thể được áp dụng trên nhiều hệ thống bảo mật, thực hiện trong các giao dịch trên mạng, thực hiện tạo và xác thực chữ ký điện tử.

Bên cạnh đó, cần tối ưu hơn nữa các phép toàn bằng các cài đặt được thử nghiệm với các ngôn ngữ lập trình mạnh.

Tác giả mong muốn có thể tiếp tục phát triển để đưa các kết quả đã nghiên cứu vào ứng dụng trong thực tế.