**TRƯỜNG ĐẠI HỌC HỒNG ĐỨC**

**KHOA CNTT-TT**

--- 🕮 ---

**Logo

Description automatically generated**

**BÁO CÁO**

**­­­­*Đề tài: Hệ mã hóa RSA***

**GVHD: Ts.Gv Trịnh Viết Cường**

**Nhóm thực hiện:** NHÓM 4

1. Lê Xuân Quang (Nhóm trưởng) 2061030026

2. Nguyễn Văn ánh 2061030001

3. Lê Hữu Đức Hải 2061030005

4. Phạm Công Quân 2061030008

5. Cao Đăng Anh Kiệt 2061030011

*Thanh Hóa,ngày 10 tháng 04 năm 2022*

MỤC LỤC

[**PHẦN I: HỆ MÃ HÓA RSA**](#_Toc97489796)

[1. Khái niệm 4](#_Toc97489797)

[2. Hoat động của hệ mã 4](#_Toc97489798)

[3. Chi tiết các bước thực hiện 5](#_Toc97489799)

[4. Độ an toàn của hệ mã Caesar 6](#_Toc97489800)

[**PHẦN II: CÀI ĐẶT HỆ MÃ RSA**](#_Toc97489801)

[1. Mã hóa 10](#_Toc97489807)

[2. Giải mã 12](#_Toc97489810)

**LỜI NÓI ĐẦU**

Mã hóa và giải mã dữ liệu là kỹ thuật nhằm bảo mật dữ liệu, xác thực và đảm bảo tính toàn vẹn của thông tin được truyền trên các phương tiện truyền thông. Mã hóa là thao tác chuyển đổi văn bản hoặc dữ liệu ban đầu thành một hình thức mã hóa hay các bản mã bằng cách áp dụng các biến đổi toán học. Ngược lại với thao tác này là giải mã. Giải mã truy xuất dữ liệu gốc từ các bản mã sử dụng một khóa giải mã.

Khi nói đến mã hóa, chúng ta dễ liên tưởng đến những bộ phim trình chiếu những đoạn mã số dài chạy nhấp nháy trên màn hình với những thông điệp khó hiểu. Hay cuộc chiến giữa Apple và FBI về vụ thông tin bị mã hóa mà chính phủ Mỹ buộc Apple phải giải mã thông tin trên chiếc iPhone của thủ phạm trong vụ xả súng ở San Bernardino, Mỹ. Nói đơn giản, mã hóa là một kỹ thuật làm cho nội dung không thể đọc được đối với bất kỳ ai không có chìa khóa. Ta có nghe thấy những thông tin như gián điệp sử dụng mã hóa để gửi thông tin bí mật, các tướng chỉ huy quân đội gửi nội dung mã hóa để phối hợp chiến đấu cho đồng bộ, còn tội phạm sử dụng mã hóa để trao đổi thông tin và lên kế hoạch hành động, tin nhắn của người dùng được mã hóa để kể cả quản trị tối cao của hệ thống cũng không đọc được...

Các hệ thống mã hóa cũng xuất hiện trong hầu hết mọi lĩnh vực liên quan đến công nghệ, không chỉ che dấu thông tin của tội phạm, kẻ thù hay gián điệp mà còn xác thực và làm rõ những thông tin rất cơ bản, rất cá nhân.

# **PHẦN I: HỆ MÃ HÓA RSA**

## **1. Khái niệm**

## 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.

## **2. Hoạt động của hệ mã**

Thuật toán RSA có hai khóa: khóa công khai (hay khóa công cộng) và khóa bí mật (hay khóa cá nhân). Mỗi khóa là những số cố định sử dụng trong quá trình mã hóa và giải mã. Khóa công khai được công bố rộng rãi cho mọi người và được dùng để mã hóa. Những thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa bí mật tương ứng. Nói cách khác, mọi người đều có thể mã hóa nhưng chỉ có người biết khóa cá nhân (bí mật) mới có thể giải mã được.

Mấu chốt cơ bản của việc sinh khóa trong RSA là tìm được bộ 3 số tự nhiên e, d và n sao cho:

med = m mod n

và một điểm không thể bỏ qua là cần bảo mật cho d sao cho dù biết e, n hay thậm chí cả m cũng không thể tìm ra d được.

**3. Chi tiết các bước thực hiện**

Cụ thể, khóa của RSA được sinh như sau:

* + Chọn 2 số nguyên tố p và q
  + Tính n = pq. Sau này, n sẽ được dùng làm modulus trong cả public key và private key.
  + Tính một số giả nguyên tố bằng phi hàm carmichael như sau: λ(n) = BCNN(λ(p), λ(q)) = BCNN(p − 1, q − 1). Giá trị này sẽ được giữ bí mật.
  + Chọn một số tự nhiên e trong khoảng (1, λ(n)) sao cho ƯCLN(e, λ(n)) = 1, tức là e và λ(n) nguyên tố cùng nhau.
  + Tính toán số d sao cho d ≡ 1/e (mod λ(n)) hay viết dễ hiểu hơn thì de ≡ 1 (mod λ(n)). Số d được gọi là số nghịch đảo modulo của e (theo modulo mod λ(n))
  + Public key sẽ là bộ số (n, e), và private key sẽ là bộ số (n, d). Chúng ta cần giữ private key thật cẩn thận cũng như các số nguyên tố p và q vì từ đó có thể tính toán các khóa rất dễ dàng.

Public key sẽ là bộ số (n, e), và private key sẽ là bộ số (n, d). Chúng ta cần giữ private key thật cẩn thận cũng như các số nguyên tố p và q vì từ đó có thể tính toán các khóa rất dễ dàng.

Trong thực hành, chúng ta thường chọn e tương đối nhỏ để việc mã hóa và giải mã nhanh chóng hơn. Giá trị thường được sử dụng là e = 65537. Ngoài ra, chúng ta có thể tính số giả nguyên tố bằng phi hàm Euler φ(n) = (p − 1)(q − 1) và dùng nó như λ(n). Vì φ(n) là bội của λ(n) nên các số d thỏa mãn de ≡ 1 (mod φ(n)) cũng sẽ thỏa mãn d ≡ 1/e (mod λ(n)). Tuy nhiên, một số tác dụng phụ của việc này là d thường sẽ trở nên lớn hơn mức cần thiết.

**PHẦN III: CÀI ĐẶT**

1 **Hàm số nguyên tố**

int prime(long int pr)

{

int i;

// vi du can 20 = 4 , xet gia tri tu 2 -> 4 , 20 chia het cho 2 va 4 nen 20 khong la ngto

j = sqrt(pr);

for (i = 2; i <= j; i++)

{

if (pr % i == 0)

return 0;

}

return 1;

}

2.**Hàm nhập và kiểm tra input**

cout << "\nENTER FIRST PRIME NUMBER \n";

cin >> p;

flag = prime(p);

if (flag == 0)

{

cout << "\nWRONG INPUT\n";

exit(1);

}

cout << "\nENTER ANOTHER PRIME NUMBER\n";

cin >> q;

flag = prime(q);

if (flag == 0 || p == q)

{

cout << "\nWRONG INPUT\n";

exit(1);

}

cout << "\nENTER MESSAGE\n";

fflush(stdin);

cin >> msg;

for (i = 0; msg[i] != '\0'; i++) // luu cac gia tri cua ki tu duoi dang so , '0' la ki tu cuoi ,

m[i] = msg[i];

n = p \* q;

t = (p - 1) \* (q - 1);

ce();

cout << "\nPOSSIBLE VALUES OF e AND d ARE\n";

for (i = 0; i < j - 1; i++)

cout << e[i] << "\t" << d[i] << "\n";

encrypt();

decrypt();

return 0;

}

3**. Hàm đưa các số nguyên tố từ 2 đến t vào mảng**

void ce()

{

int k;

k = 0;

for (i = 2; i < t; i++)

{

if (t % i == 0)

continue;

flag = prime(i);

if (flag == 1 && i != p && i != q)

{

e[k] = i;

flag = cd(e[k]);

if (flag > 0)

{

d[k] = flag;

k++;

}

if (k == 99)

break;

}

}

}

long int cd(long int x)

{

long int k = 1;

while (1)

{

k = k + t;

if (k % x == 0)

return (k / x);

}

}

**4 Hàm mã hóa**

void encrypt()

{

long int pt, ct, key = e[0], k, len;

i = 0;

len = strlen(msg);

while (i != len)

{

pt = m[i];

pt = pt - 96;

k = 1;

for (j = 0; j < key; j++)

{

k = k \* pt;

k = k % n;

}

temp[i] = k;

ct = k + 96;

en[i] = ct;

i++;

}

en[i] = -1;

cout << "\nTHE ENCRYPTED MESSAGE IS\n";

for (i = 0; en[i] != -1; i++)

printf("%c", en[i]);

}

**5 Hàm giải mã**

void decrypt()

{

long int pt, ct, key = d[0], k;

i = 0;

while (en[i] != -1)

{

ct = temp[i];

k = 1;

for (j = 0; j < key; j++)

{

k = k \* ct;

k = k % n;

}

pt = k + 96;

m[i] = pt;

i++;

}

m[i] = -1;

cout << "\nTHE DECRYPTED MESSAGE IS\n";

for (i = 0; m[i] != -1; i++)

printf("%c", m[i]);

}

**Danh mục tài liệu tham khảo**

1. https://vi.wikipedia.org/wiki/RSA\_(m%C3%A3\_h%C3%B3a)

2. https://www.sanfoundry.com/cpp-program-implement-rsa-algorithm/