**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

****



**AN TOÀN VÀ BẢO MẬT DỮ LIỆU TRONG HỆ THỐNG THÔNG TIN**

**tÊN ĐỀ TÀI: MÃ HOÁ RSA TRONG TIN NHẮN VĂN BẢN**

Nhóm 5:

Trần Nguyên Lộc – 3120410297

Võ Đăng Quang – 3120410429

Phạm Nhật Tân – 3120410465

Phạm Minh Quân – 3120410438

Trịnh Hùng Thái – 3120410471

**Giảng viên phụ trách:**

**ThS. TRƯƠNG TẤN KHOA**

**Thành phố Hồ Chí Minh, tháng 12 năm 2023**

# **Danh sách nhóm 5**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thành viên | MSSV | Mức độ tham gia |
| 1 | Trần Nguyên Lộc | 3120410297 | 100% |
| 2 | Võ Đăng Quang | 3120410429 | 100% |
| 3 | Phạm Nhật Tân | 3120410465 | 100% |
| 4 | Phạm Minh Quân | 3120410438 | 100% |
| 5 | Trịnh Hùng Thái | 3120410471 | 100% |

# **Lời cảm ơn**

Trước hết em xin gửi đến lời cảm ơn chân thành và sâu sắc nhất đến thầy ThS. Trương Tấn Khoa, trong quá trình tìm hiểu về RSA, tuy thời gian không nhiều nhưng với sự hướng dẫn và giúp đỡ của thầy, nhóm em đã hoàn thành đồ án.

Tiếp đến em xin giành lời cảm ơn đến quý thầy cô Trường Đại học Sài Gòn – khoa Công nghệ thông tin đã truyền đạt cho em những kiến thức vô cùng quý báu và bổ ích trong suốt quá trình nghiên cứu và học tập tại trường.

Xin chân thành cảm ơn tới những người bạn đã luôn sát cánh cùng em, những lời động viên, những lần hỗ trợ những lúc cần thiết đã phần nào giúp em hoàn thành đồ án này. Do thời gian hạn chế nên phạm vi nghiên cứu và vẫn còn một số vấn đề chưa được giải quyết triệt để. Nhóm em mong nhận được sự đóng góp của thầy và các bạn, nhóm em xin chân thành cảm ơn thầy và các bạn!

Cuối cùng, em xin cảm ơn đến ba mẹ và người thân trong gia đình đã hỗ trợ và tạo điều kiện thuận lợi cho em trong suốt thời gian học tập và nghiên cứu tại Đại học Sài Gòn.

# **Mục lục**

[**Danh sách nhóm 5** i](#_Toc151285413)

[**Lời cảm ơn** ii](#_Toc151285414)

[**Mục lục** iii](#_Toc151285415)

[**Danh mục hình ảnh** iv](#_Toc151285416)

[**Danh mục bảng biểu** v](#_Toc151285417)

[**Lời mở đầu** 1](#_Toc151285418)

[**Chương 1: KHÁI QUÁT VỀ MÃ HOÁ BẤT ĐỐI XỨNG** 2](#_Toc151285419)

[**1.1.** **Mật mã hoá công khai** 2](#_Toc151285420)

[**1.2.** **Mã hoá bất đối xứng RSA** 2](#_Toc151285421)

[**Chương 2: RSA** 4](#_Toc151285422)

[**2.1.** **Khái quát đồ án** 4](#_Toc151285423)

[**2.2.** **Cơ chế hoạt động** 5](#_Toc151285424)

[**2.2.1.** **Cách tạo khoá** 7](#_Toc151285425)

[**2.2.2.** **Mã hóa** 7](#_Toc151285426)

[**2.2.3.** **Mã hoá và giải mã** 7](#_Toc151285427)

[**2.2.4.** **Tính bảo mật** 7](#_Toc151285428)

[**2.2.5.** **Quá trình tạo khóa** 7](#_Toc151285429)

[**2.2.6.** **Tốc độ** 7](#_Toc151285430)

[**Chương 3. Mã hóa RSA trong tin nhắn văn bản.** 8](#_Toc151285431)

[**3.1.** **Giải thuật RSA** 8](#_Toc151285432)

[**3.2.** **Demo và kết quả thực nghiệm** 8](#_Toc151285433)

[**3.2.1.** **Demo** 8](#_Toc151285434)

[**3.2.2.** **Kết quả** 11](#_Toc151285435)

[**KẾT LUẬN** 12](#_Toc151285436)

[**Tài liệu tham khảo** 13](#_Toc151285437)

# **Danh mục hình ảnh**

# **Danh mục bảng biểu**

# **Lời mở đầu**

Cùng với sự phát triển của công nghệ thông tin, công nghệ mạng máy tính và sự phát triển của mạng internet ngày càng phát triển đa dạng và phong phú. Các dịch vụ trên mạng đã thâm nhập vào hầu hết các lĩnh vực trong đời sống xã hội. Các thông tin trên Internet cũng đa dạng về nội dung và hình thức, trong đó có rất nhiều thông tin cần được bảo mật cao hơn bởi tính kinh tế, tính chính xác và tính tin cậy của nó.

Trước đây khi công nghệ máy tính chưa phát triển, khi nói đến vấn đề an toàn bảo mật thông tin, chúng ta thường hay nghĩ đến các biện pháp nhằm đảm bảo cho thông tin được trao đổi hay cất giữ một cách an toàn và bí mật, chẳng hạn là các biện pháp như: Đóng dấu và ký niêm phong một bức thư để biết rằng lá thư có được chuyển nguyên vẹn đến người nhận hay không, dùng mật mã mã hóa thông điệp để chỉ có người gửi và người nhận hiểu được thông điệp, lưu giữ tài liệu trong các két sắt có khóa tại nơi được bảo vệ nghiêm ngặt.

Bên cạnh đó, thì luôn có nhưng kẻ nhòm ngó để có thể lấy cắp được thông tin nhằm mục đích xấu cho bản thân.



Vậy nên đã có rất nhiều người chọn cách mã hóa tin nhắn để tăng tính bảo mật. Nhưng nói đến tin nhắn thì sẽ tồn tại người gửi và người nhận, vậy làm thế nào có thể gửi cho ai đó một tin nhắn được mã hóa mà không có cô hội trước đó? Đây cũng là lí do nhóm chọn đề tài:”Mã hoá RSA trong tin nhắn văn bản”.

# **Chương 1: KHÁI QUÁT VỀ MÃ HOÁ BẤT ĐỐI XỨNG**

## **Mật mã hoá công khai**

Mật mã khóa công khai (Public Key Certificate – PKC), còn được gọi là mật mã hóa bất đối xứng, là một cơ cấu sử dụng cả chìa khóa cá nhân và chìa khóa công khai, trái ngược với chìa khóa đơn được sử dụng trong mật mã hóa đối xứng. Việc sử dụng các cặp chìa khóa khiến cho PKC có một bộ các đặc điểm và khả năng độc đáo có thể được sử dụng để giải quyết các thách thức tồn tại cố hữu trong các kỹ thuật mã hóa khác. Hình thức mật mã này đã trở thành một yếu tố quan trọng trong bảo mật hiện nay.

Trong sơ đồ PKC, Public Key được người gửi sử dụng để mã hóa thông tin, trong khi Private Key được người nhận sử dụng để giải mã. Cả hai Key là khác nhau, trong đó Public Key có thể được chia sẻ an toàn mà không ảnh hưởng đến tính bảo mật của Private Key. Mỗi cặp Key bất đối xứng là duy nhất, đảm bảo rằng một thông điệp được mã hóa bằng Public Key chỉ có thể được đọc bởi người sở hữu Private Key tương ứng. Một trong những thuật toán thông dụng nhất cho mã hóa bất đối xứng được sử dụng ngày nay có tên là RSA.

## **Mã hoá bất đối xứng RSA**

RSA được Rivest, **S**hamir và **A**dleman phát triển, là một thuận toán mật mã hóa khóa công khai. Nó đánh dấu một sự tiến hóa vượt bậc của lĩnh vực mật mã học trong việc sử dụng khóa công khai. 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.

Thuật toán được Ron Rivest, Adi Shamir và Len Adleman mô tả lần đầu tiên vào năm 1977 tại Học viện Công nghệ Massachusetts (MIT). Tên của thuật toán lấy từ 3 chữ cái đầu của tên 3 tác giả.

Trước đó, vào năm 1973, Clifford Cocks, một nhà toán học người Anh làm việc tại GCHQ, đã mô tả một thuật toán tương tự. Với khả năng tính toán tại thời điểm đó thì thuật toán này không khả thi và chưa bao giờ được thực nghiệm. Tuy nhiên, phát minh này chỉ được công bố vào năm 1997 vì được xếp vào loại tuyệt mật.

RSA là một thí dụ điển hình về một đề tài toán học trừu tượng lại có thể áp dụng thực tiễn vào đời sống thường nhật . Khi nghiên cứu về các số nguyên tố, ít có ai nghĩ rằng khái niệm số nguyên tố lại có thể hữu dụng vào lãnh vực truyền thông.

# **Chương 2: RSA**

## **Khái quát đồ án**

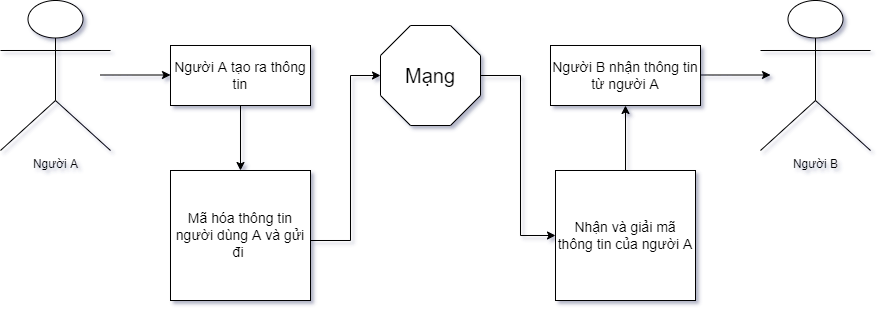
Trong mọi lĩnh vực kinh tế, chính trị, xã hội, quân sự... luôn có nhu cầu trao đổi thông tin giữa các cá nhân, các công ty, tổ chức, hoặc giữa các quốc gia với nhau. Ngày nay, với sự phát triển của công nghệ thông tin đặt biệt là mạng internet thì việc truyền tải thông tin đã dễ dàng và nhanh chóng hơn.

A white hexagon with black text

Description automatically generated

Hình 2. 1. Mô hình trao đổi thông tin qua mạng theo cách thông thường

Và vấn đề đặt ra là tính bảo mật trong quá trình truyền tải thông tin, đặt biệt quan trọng đối với những thông tin liên quan đến chính trị, quân sự, hợp đồng kinh tế.... Vì vậy nghành khoa học nghiên cứu về mã hóa thông tin được phát triển. Việc mã hóa là làm cho thông tin biến sang một dạng khác khi đó chỉ có bên gửi và bên nhận mới đọc được, còn người ngoài dù nhận được thông tin nhưng cũng không thể hiểu được nội dung.



Hình 2.2. Mô hình trao đổi thông tin qua mạng theo phương pháp mã hóa

Như chúng ta thấy ở mô hình 1.1: Việc trao đổi thông tin được thực hiện qua các bước sau:

* Tạo ra thông tin cần gửi đi.
* Gửi thông tin này cho đối tác.

Ở mô hình 1.2: Việc trao đổi thông tin được thực hiện:

* Tạo thông tin cần gửi
* Mã hóa và gửi thông tin đã được mã hóa đi. Đối tác nhận và giải mã thông tin
* Đối tác có được thông tin ban đầu của người gửi.

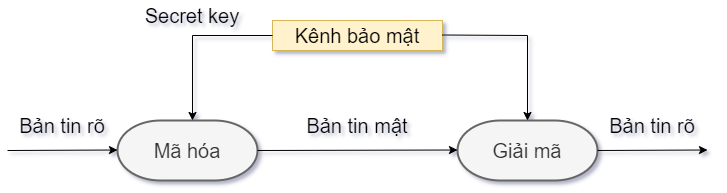
Với 2 thao tác mã hóa và giải mã ta đã đảm bảo thông tin được gửi an toàn và chính xác.

Chúng ta có nhiều phương pháp để mã hóa thông tin: Ở đây ta tìm hiểu về hệ mã hóa công khai RSA.

## **Cơ chế hoạt động**

* *Phân biệt mã hóa bí mật và mã hóa công khai*

Mã hóa bí mật: thông tin sẽ được mã hóa theo một phương pháp ứng với một key, key này dùng để lập mã và đồng thời cũng để giải mã. Vì vậy key phải được giữ bí mật, chỉ có người lập mã và người nhận biết được, nếu key bị lộ thì người ngoài sẽ dễ dàng giải mã và đọc được thông tin.



Hình 2.3. Mô hình mã hóa bí mật

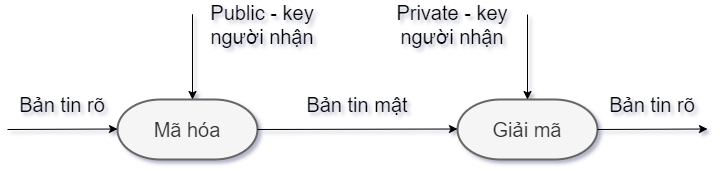
Mã hóa công khai: sử dụng 2 key public key private key.

Public key: Được sử dụng để mã hoá những thông tin mà ta muốn chia sẻ với bất cứ ai. Chính vì vậy ta có thể tự do phân phát nó cho bất cứ ai mà ta cần chia sẻ thông tin ở dạng mã hoá.

Privite key: Đúng như cái tên, Key này thuộc sở hữu riêng tư của bạn(ứng với public key) và nó được sử dụng để giải mã thông tin. Chỉ mình bạn sở hữu nó, Key này không được phép và không lên phân phát cho bất cứ ai.

Nghĩa là mỗi người sẽ giữ 2 key 1 dùng để mã hóa, key này được công bố rộng rãi, 1 dùng để giải mã, key này giữ kín.

Khi ai đó có nhu cầu trao đổi thông tin với bạn, sẻ dùng public key mà bạn công bố để mã hóa thông tin và gửi cho bạn, khi nhận được bạn dùng private key để giải mã. Những người khác dù có nhận được thông tin nhưng không biết được private key thì cũng không thể giải mã và đọc được thông tin.



Hình 2.4. Mô hình mã hóa công khai

Hàm một phía.

Một hàm một phía là hàm mà dễ dàng tính toán ra quan hệ một chiều nhưng rất khó để tính ngược lại. Ví như : biết giả thiết x thì có thể dễ dàng tính ra f(x), nhưng nếu biết f(x) thì rất khó tính ra được x. Trong trường hợp này “khó” có nghĩa là để tính ra được kết quả thì phải mất hàng triệu năm để tính toán, thậm chí tất cả máy tính trên thế giới này đều tính toán công việc đó.

Vậy thì hàm một phía tốt ở những gì ? Chúng ta không thể sử dụng chúng cho sự mã hoá. Một thông báo mã hoá với hàm một phía là không hữu ích, bất kỳ ai cũng không giải mã được. Đối với mã hoá chúng ta cần một vài điều gọi là cửa sập hàm một phía.(khóa)

Hộp thư là một ví dụ rất tuyệt về hàm một phía cũng như hình thức mã hóa này. Bất kỳ ai cũng có thể bỏ thư vào thùng. Bỏ thư vào thùng là một hành động công cộng. Mở thùng thư không phải là hành động công cộng. Nó là việc khó khăn, khi bạn không có chìa khóa ứng với thùng thư. Hơn nữa nếu bạn có điều bí mật (chìa khóa), nó thật dễ dàng mở hộp thư. Hệ mã hóa công khai có rất nhiều điều giống nhau như vậy.

### **Cách tạo khoá**

### **Mã hóa**

### **Mã hoá và giải mã**

### **Tính bảo mật**

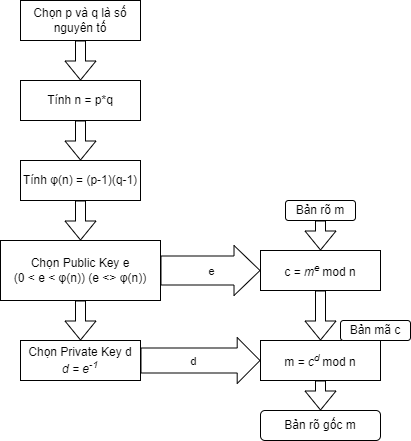
### **Quá trình tạo khóa**

### **Tốc độ**

# **Chương 3. Mã hóa RSA trong tin nhắn văn bản.**

## **Giải thuật RSA**

Sơ đồ giải thuật:



## **Demo và kết quả thực nghiệm**

### **Demo**

Cụ thể code RSA:

* *Tạo khoá:*

def *generateKeys*(keysize=1024):

    e = d = N = 0

    # *get prime nums, p & q*

    p = generateLargePrime(keysize)

    q = generateLargePrime(keysize)

    # *print(f"p: {p}")*

    # *print(f"q: {q}")*

    N = p \* q  # *RSA Modulus*

    phiN = (p - 1) \* (q - 1)  # *totient*

    # *choose e*

    # *e is coprime with phiN & 1 < e <= phiN*

*while* True:

        e = random.randrange(2 \*\* (keysize - 1), 2 \*\* keysize - 1)

*if* (isCoPrime(e, phiN)):

*break*

    # *choose d*

    # *d is mod inv of e with respect to phiN, e \* d (mod phiN) = 1*

    d = modularInv(e, phiN)

*return* p, q, e, d, N

* *Mã hoá:*

    def *encrypt*(self, msg):

        cipher = ""

*for* c *in* msg:

            m = ord(c)

            cipher += str(pow(m, self.e, self.N)) + " "

*return* cipher

* *Giải mã:*

    def *decrypt*(self, cipher):

        msg = ""

        parts = cipher.split()

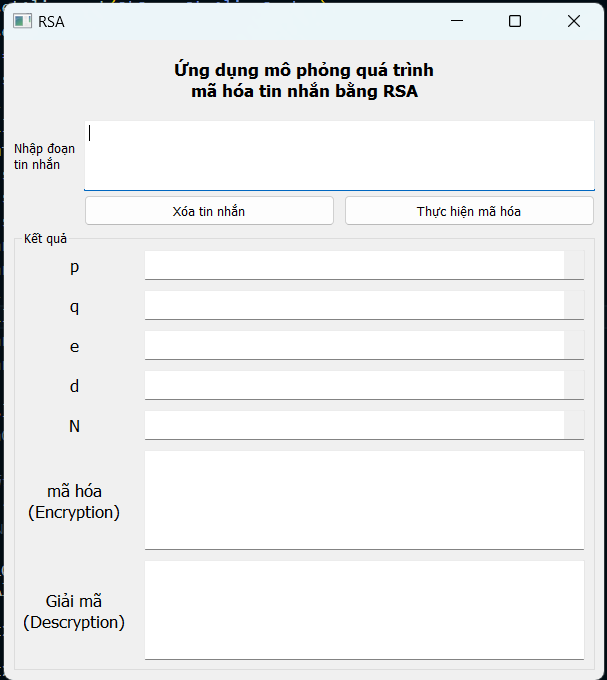
*for* part *in* parts:

*if* part:

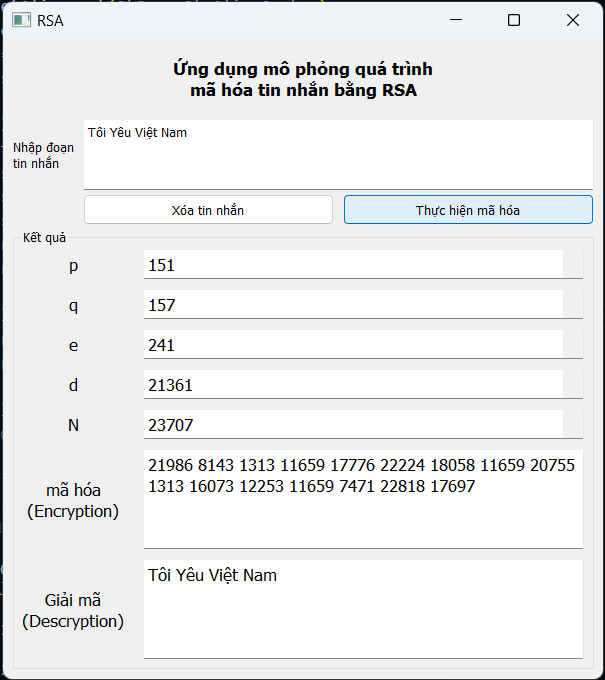
                c = int(part)

                msg += chr(pow(c, self.d, self.N))

* *Giao diện demo:*



### **Kết quả**



# **KẾT LUẬN**

# **Tài liệu tham khảo**

1. Giáo trình An Toàn Bảo Mật Hệ Thống Thông Tin\_TS Hoàng Xuân Dậu\_HVCNBCVT.(Chương 4: Các kỹ thuật mã hóa thông tin)

2. <http://vi.wikipedia.org/wiki/RSA_(m%C3%A3_h%C3%B3a)>

3. <http://en.wikipedia.org/wiki/RSA>

4. <http://vi.wikipedia.org/wiki/Ch%E1%BB%AF_k%C3%BD_s%E1%BB%91>

5. <http://en.wikipedia.org/wiki/Digital_signature>

6. <http://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html>

7. <http://vi.wikipedia.org/wiki/SHA>

8. <http://www.java2s.com/Tutorial/Java/0490__Security/UseSHA1.html>