

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. TRƯỜNG 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
Lời cảm ơn	ii
Mục lục	iii
Danh mục hình ảnh	v
Lời mở đầu	1
Chương 1: KHÁI QUÁT VỀ MÃ HOÁ BẤT ĐỐI XỨNG	2
1.1. Mật mã hoá công khai	2
1.2. Mã hoá bất đối xứng RSA	2
1.3. Tiêu chuẩn về RSA	3
1.3.1. PKCS#1. Ver1.0-2.2. RSA Cryptography Standard	3
1.3.2. TCVN 7635:2007. Tiêu chuẩn mật mã – Chữ ký số	3
Chương 2: RSA	5
2.1. Khái quát đề án	5
2.2. Cơ chế hoạt động	6
2.2.1. Phân biệt mã hóa bí mật và mã hóa công khai	6
2.2.2. Cách tạo khoá	8
2.2.3. Mã hóa	10
2.2.4. Giải mã	11
2.2.5. Tính bảo mật	12
2.2.6. Quá trình tạo khóa	13
2.2.7. Tốc độ	14
Chương 3. Mã hóa RSA trong tin nhắn văn bản.	15
3.1. Giải thuật RSA	15

3.2. Mô phỏng quá trình mã hoá tin nhắn bằng RSA	15
3.2.1. Demo	15
3.2.2. Kết quả.....	17
KẾT LUẬN	19
Các vấn đề đạt được	19
Hạn chế	19
TÀI LIỆU THAM KHẢO	20

Danh mục hình ảnh

Hình 0. 1. Đối tượng đánh cắp thông tin.....	1
Hình 2. 1. 1. Mô hình trao đổi thông tin qua mạng theo cách thông thường.....	5
Hình 2. 1. 2. Mô hình trao đổi thông tin qua mạng theo phương pháp mã hóa.....	5
Hình 2. 2. 1. 1. Mô hình mã hoá bí mật	6
Hình 2. 2. 1. 2. Mô hình mã hoá công khai.....	7
Hình 3. 1. 1. Giải thuật RSA	15
Hình 3. 2. 1. 1. Giao diện mô phỏng.....	17
Hình 3. 2. 2. 1. Nhập bản rõ	17
Hình 3. 2. 2. 2. Các khoá.....	18
Hình 3. 2. 2. 3. Mã hoá.....	18
Hình 3. 2. 2. 4. Giải mã.....	18

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 những kẻ nhòm ngó để có thể lấy cắp được thông tin nhằm mục đích xấu cho bản thân.



Hình 0. 1. Đối tượng đánh cắp thông tin

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

1.1. 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.

1.2. Mã hoá bất đối xứng RSA

RSA được Rivest, Shamir và Adleman phát triển, là một thuật 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.

1.3. Tiêu chuẩn về RSA

1.3.1. PKCS#1. Ver1.0-2.2. RSA Cryptography Standard

PKCS (Tiêu chuẩn mật mã khóa công khai – Public-Key Cryptography Standard) - do Phòng thí nghiệm RSA (Mỹ) ban hành - là một tập hợp các tiêu chuẩn để hoàn thiện hệ mật mã khóa công khai, được phát triển vào năm 1991.

Trong đó, PKCS #1 là một trong những tiêu chuẩn được sử dụng nhiều nhất (thực tế) cho việc sử dụng RSA trong thực tế.

Một nâng cấp lớn cho PKCS #1, từ phiên bản 1.0 đến phiên bản 2.0 là giới thiệu các chế độ bổ sung với đối số bảo mật mạnh hơn và RSA đa nguyên tố.

Phiên bản 2.2 cập nhật thêm danh sách các thuật toán băm như SHA-224, SHA-512/224 và SHA-512/256.

1.3.2. TCVN 7635:2007. Tiêu chuẩn mật mã – Chữ ký số

TCVN 7635:2007 do Tiểu Ban kỹ thuật tiêu chuẩn TCVN/JTC 1/SC 27 “*Các kỹ thuật mật mã*” biên soạn trên cơ sở dự thảo đề nghị của Ban cơ yếu Chính phủ, Tổng cục Tiêu chuẩn Đo lường Chất lượng đề nghị, Bộ Khoa học và Công nghệ công bố.

Tiêu chuẩn này áp dụng cho các chữ ký số sử dụng trong hoạt động giao dịch điện tử của mọi tổ chức, công dân Việt Nam và tổ chức, công dân nước ngoài có quan hệ kinh tế - xã hội với tổ chức, công dân Việt Nam.

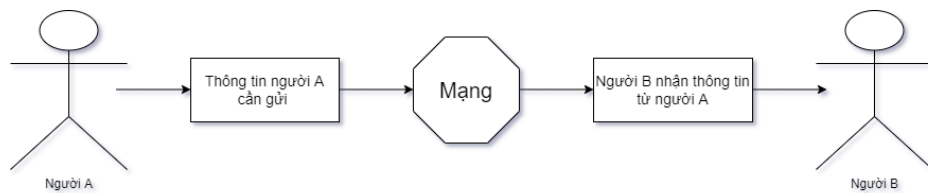
Tiêu chuẩn này quy định 3 thành phần cần thiết cho lược đồ chữ ký số:

- Thành phần thứ nhất là thuật toán chữ ký số RSA-PSS.
- Thành phần thứ hai là thuật toán hàm băm SHA-256.
- Thành phần cuối cùng là thuật toán số giả ngẫu nhiên dùng AES-128.

Chương 2: RSA

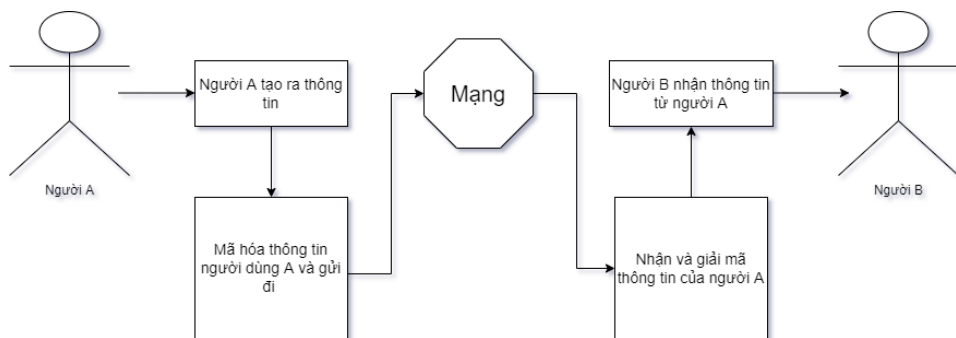
2.1. Khái quát đồ án

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 đặc biệt là mạng internet thì việc truyền tải thông tin đã dễ dàng và nhanh chóng hơn.



Hình 2. 1. 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, đặc biệt quan trọng đối với những thông tin liên quan đến chính trị, quân sự, hợp đồng kinh tế.... Vì vậy ngành 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. 1. 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.

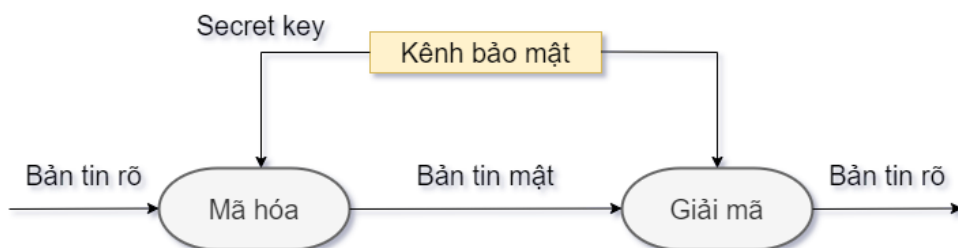
Với 2 thao tác mã hóa và giải mã ta đã đảm bảo thông tin được gửi an toàn và chính xác.

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

2.2. Cơ chế hoạt động

2.2.1. 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. 2. 1. 1. Mô hình mã hoá 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.

Private 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, sẽ 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. 2. 1. 2. Mô hình mã hoá 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.

2.2.2. Cách tạo khoá

Chúng ta cần tạo ra một cặp khóa lập mã và giải mã theo phương pháp sau:

Bước 1: Chọn số nguyên tố lớn hơn p và q với $p \neq q$, lựa chọn ngẫu nhiên và độc lập.

Bước 2: Tính: $n = p * q$.

Bước 3: Chọn một số tự nhiên e sao cho $1 < e < \phi(n)$ và là số nguyên tố cùng nhau với $\phi(n)$.

Bước 4: Tìm: d sao cho $de \equiv 1 \pmod{\phi(n)}$.

Một số lưu ý trước khi qua bước 5:

- Các số nguyên tố thường được chọn bằng phương pháp thử xác suất.
- Các bước 4 và 5 có thể được thực hiện bằng giải thuật Euclid mở rộng (xem thêm: số học môđun)

Bước 5 có thể viết cách khác: Tìm số tự nhiên x sao cho $d = \frac{x(p-1)(q-1)+1}{e}$ cũng là số tự nhiên. Khi đó sử dụng giá trị $d \bmod (p-1)(q-1)$.

Từ bước 3, PKCS#1 V2.1 sử dụng $\lambda = LCM(p-1, q-1)$ thay cho $\phi = (p-1)(q-1)$

➤ *Khóa công khai bao gồm:*

- n , môđun, và
- e , số mũ công khai (cũng gọi là số mũ mã hóa)

➤ *Khóa bí mật bao gồm:*

- n , môđun, xuất hiện cả trong khóa công khai và khóa bí mật và
- d , số mũ bí mật (cũng gọi là số mũ giải mã)

➤ *Một dạng khác của khóa bí mật bao gồm:*

- p and q , hai số nguyên tố chọn ban đầu,
- $d \bmod (p-1)$ và $d \bmod (q-1)$ (thường được gọi là d_{mp1} và d_{mq1}),
- $(1/q) \bmod p$ (thường được gọi là i_{qmp})

Dạng này cho phép thực hiện giải mã và ký nhanh hơn với việc sử dụng định lý số dư Trung Quốc (tiếng Anh: *Chinese Remainder Theorem - CRT*). Ở dạng này, tất cả thành phần của khóa bí mật phải được giữ bí mật.

Ở đây, p và q giữ vai trò rất quan trọng. Chúng là các phân tố của n và cho phép tính d khi biết e . Nếu không sử dụng dạng sau của khóa bí mật (dạng CRT) thì p và q sẽ được xóa ngay sau khi thực hiện xong quá trình tạo khóa.

Chuyển đổi thông tin:

Trước khi thực hiện mã hóa, ta phải thực hiện việc chuyển đổi thông tin (chuyển đổi từ M sang m) sao cho không có giá trị nào của M tạo ra văn bản mã không an toàn. Nếu không có quá trình này, RSA sẽ gặp phải một số vấn đề sau:

- Nếu $m = 0$ hoặc $m = 1$ sẽ tạo ra các bản mã có giá trị là 0 và 1 tương ứng
- Khi mã hóa với số mũ nhỏ (chẳng hạn $e = 3$) và m cũng có giá trị nhỏ, giá trị me cũng nhận giá trị nhỏ (so với n). Như vậy phép môđun không có tác dụng và có thể dễ dàng tìm được m bằng cách khai căn bậc e của c (bỏ qua môđun).
- RSA là phương pháp mã hóa xác định (không có thành phần ngẫu nhiên) nên kẻ tấn công có thể thực hiện tấn công lựa chọn thông tin bằng cách tạo ra một bảng tra giữa thông tin và bản mã. Khi gặp một bản mã, kẻ tấn công sử dụng bảng tra để tìm ra thông tin tương ứng.

Trên thực tế, ta thường gặp 2 vấn đề đầu khi gửi các bản tin ASCII ngắn với m là nhóm vài ký tự ASCII. Một đoạn tin chỉ có 1 ký tự NULL sẽ được gán giá trị $m=0$ và cho ra bản mã là 0 bất kể giá trị của e và N . Tương tự, một ký tự ASCII khác, SOH, có giá trị 1 sẽ luôn cho ra bản mã là 1. Với các hệ thống dùng giá trị e nhỏ thì tất cả ký tự ASCII đều cho kết quả mã hóa không an toàn vì giá trị lớn nhất của m chỉ là 255 và 2553 nhỏ hơn giá trị n chấp nhận được. Những bản mã này sẽ dễ dàng bị phá mã.

Để tránh gặp phải những vấn đề trên, RSA trên thực tế thường bao gồm một hình thức chuyển đổi ngẫu nhiên hóa m trước khi mã hóa. Quá trình chuyển đổi này phải đảm bảo rằng m không rơi vào các giá trị không an toàn. Sau khi chuyển đổi,

mỗi thông tin khi mã hóa sẽ cho ra một trong số khả năng trong tập hợp bản mã. Điều này làm giảm tính khả thi của phương pháp tấn công lựa chọn thông tin (một thông tin sẽ có thể tương ứng với nhiều bản mã tùy thuộc vào cách chuyển đổi).

Một số tiêu chuẩn, chẳng hạn như PKCS, đã được thiết kế để chuyển đổi thông tin trước khi mã hóa bằng RSA. Các phương pháp chuyển đổi này bổ sung thêm bit vào M . Các phương pháp chuyển đổi cần được thiết kế cẩn thận để tránh những dạng tấn công phức tạp tận dụng khả năng biết trước được cấu trúc của thông tin. Phiên bản ban đầu của PKCS dùng một phương pháp đặc ứng (ad-hoc) mà về sau được biết là không an toàn trước tấn công lựa chọn thông tin thích ứng (adaptive chosen ciphertext attack). Các phương pháp chuyển đổi hiện đại sử dụng các kỹ thuật như chuyển đổi mã hóa bất đối xứng tối ưu (Optimal Asymmetric Encryption Padding - OAEP) để chống lại tấn công dạng này. Tiêu chuẩn PKCS còn được bổ sung các tính năng khác để đảm bảo an toàn cho chữ ký RSA (Probabilistic Signature Scheme for RSA - RSA-PSS).

2.2.3. Mã hóa

Giả sử có đoạn thông tin M cần gửi. Đầu tiên chuyển M thành một số $m < n$ theo một hàm có thể đảo ngược (từ m có thể xác định lại M) được thỏa thuận trước. Lúc này ta có m và biết n cũng như e của người nhận. Ta sẽ tính c là bản mã hóa của m theo công thức:

$$c = m^e \mod n$$

Hàm trên có thể tính dễ dàng sử dụng phương pháp tính hàm mũ (theo môđun) bằng (thuật toán bình phương và nhân) cuối cùng ta gửi c cho đối tác.

2.2.4. Giải mã

Khi đối tác nhận c từ ta. Đối tác sử dụng khóa bí mật d tìm được m từ c theo công thức sau:

$$m = c^d \pmod{n}$$

Biết m , đối tác tìm lại M theo phương pháp đã thỏa thuận trước. Quá trình giải mã hoạt động vì ta có

$$c^d \equiv (m^e)^d \equiv m^{ed} \pmod{n}.$$

Do $ed \equiv 1 \pmod{p-1}$ và $ed \equiv 1 \pmod{q-1}$, theo định lý Fermat nhỏ nên:

$$m^{ed} \equiv m \pmod{p}$$

Và

$$m^{ed} \equiv m \pmod{q}$$

Do p và q là hai số nguyên tố cùng nhau, áp dụng định lý số dư Trung Quốc, ta có:

$$m^{ed} \equiv m \pmod{pq}$$

Và

$$c^d \equiv m \pmod{n}$$

2.2.5. Tính bảo mật

Độ an toàn của hệ thống RSA dựa trên 2 vấn đề của toán học: bài toán phân tích ra thừa số nguyên tố các số nguyên lớn và bài toán RSA. Nếu 2 bài toán trên là khó (không tìm được thuật toán hiệu quả để giải chúng) thì không thể thực hiện được việc phá mã toàn bộ đối với RSA.

Bài toán RSA là bài toán tính căn bậc e môđun n (với n là hợp số): tìm số m sao cho $m=c \bmod n$, trong đó (e, n) chính là khóa công khai và c là bản mã. Hiện nay phương pháp triển vọng nhất giải bài toán này là phân tích n ra thừa số nguyên tố. Khi thực hiện được điều này, kẻ tấn công sẽ tìm ra số mũ bí mật d từ khóa công khai và có thể giải mã theo đúng quy trình của thuật toán. Nếu kẻ tấn công tìm được 2 số nguyên tố p và q sao cho: $n = p*q$ thì có thể dễ dàng tìm được giá trị $(p-1)(q-1)$ và qua đó xác định d từ e . Chưa có một phương pháp nào được tìm ra trên máy tính để giải bài toán này trong thời gian đa thức (polynomial-time). Tuy nhiên người ta cũng chưa chứng minh được điều ngược lại (sự không tồn tại của thuật toán). Xem thêm phân tích ra thừa số nguyên tố về vấn đề này.

Tại thời điểm năm 2005, số lớn nhất có thể được phân tích ra thừa số nguyên tố có độ dài 663 bit với phương pháp phân tán trong khi khóa của RSA có độ dài từ 1024 tới 2048 bit. Một số chuyên gia cho rằng khóa 1024 bit có thể sớm bị phá vỡ (cũng có nhiều người phản đối việc này). Với khóa 4096 bit thì hầu như không có khả năng bị phá vỡ trong tương lai gần. Do đó, người ta thường cho rằng RSA đảm bảo an toàn với điều kiện n được chọn đủ lớn. Nếu n có độ dài 256 bit hoặc ngắn hơn, nó có thể bị phân tích trong vài giờ với máy tính cá nhân dùng các phần mềm có sẵn. Nếu n có độ dài 512 bit, nó có thể bị phân tích bởi vài trăm máy tính tại thời điểm năm 1999. Một thiết bị lý thuyết có tên là TWIRL do Shamir và Tromer mô tả năm 2003 đã đặt ra câu hỏi về độ an toàn của khóa 1024 bit. Vì vậy hiện nay người ta khuyến cáo sử dụng khóa có độ dài tối thiểu 2048 bit.

Năm 1993, Peter Shor công bố thuật toán Shor chỉ ra rằng: máy tính lượng tử (trên lý thuyết) có thể giải bài toán phân tích ra thừa số trong thời gian đa thức. Tuy

nhien, máy tính lượng tử vẫn chưa thể phát triển được tới mức độ này trong nhiều năm nữa.

2.2.6. Quá trình tạo khóa

Việc tìm ra 2 số nguyên tố đủ lớn p và q thường được thực hiện bằng cách thử xác suất các số ngẫu nhiên có độ lớn phù hợp (dùng phép kiểm tra nguyên tố cho phép loại bỏ hầu hết các hợp số).

p và q còn cần được chọn không quá gần nhau để phòng trường hợp phân tích n bằng phương pháp phân tích Fermat. Ngoài ra, nếu $p-1$ hoặc $q-1$ có thừa số nguyên tố nhỏ thì n cũng có thể dễ dàng bị phân tích và vì thế p và q cũng cần được thử để tránh khả năng này.

Bên cạnh đó, cần tránh sử dụng các phương pháp tìm số ngẫu nhiên mà kẻ tấn công có thể lợi dụng để biết thêm thông tin về việc lựa chọn (cần dùng các bộ tạo số ngẫu nhiên tốt). Yêu cầu ở đây là các số được lựa chọn cần đồng thời ngẫu nhiên và không dự đoán được. Đây là các yêu cầu khác nhau: một số có thể được lựa chọn ngẫu nhiên (không có kiểu mẫu trong kết quả) nhưng nếu có thể dự đoán được dù chỉ một phần thì an ninh của thuật toán cũng không được đảm bảo. Một ví dụ là bảng các số ngẫu nhiên do tập đoàn Rand xuất bản vào những năm 1950 có thể rất thực sự ngẫu nhiên nhưng kẻ tấn công cũng có bảng này. Nếu kẻ tấn công đoán được một nửa chữ số của p hay q thì chúng có thể dễ dàng tìm ra nửa còn lại (theo nghiên cứu của Donald Coppersmith vào năm 1997)

Một điểm nữa cần nhấn mạnh là khóa bí mật d phải đủ lớn. Năm 1990, Wiener chỉ ra rằng nếu giá trị của p nằm trong khoảng q và $2q$ (khá phổ biến) và $d < n^{1/4}/3$ thì có thể tìm ra được d từ n và e .

Mặc dù e đã từng có giá trị là 3 nhưng hiện nay các số mũ nhỏ không còn được sử dụng do có thể tạo nên những lỗ hổng (đã đề cập ở phần chuyển đổi văn bản rõ). Giá trị thường dùng hiện nay là 65537 vì được xem là đủ lớn và cũng không quá lớn ảnh hưởng tới việc thực hiện hàm mũ.

2.2.7. Tốc độ

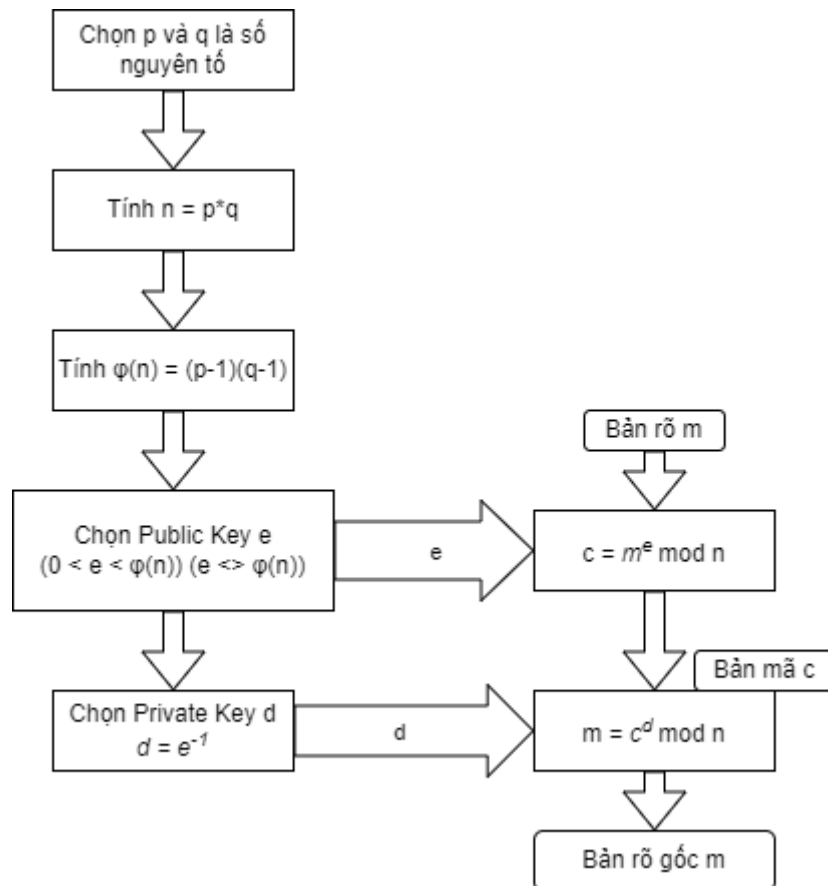
RSA có tốc độ thực hiện chậm hơn đáng kể so với DES và các thuật toán mã hóa đối xứng khác. Trên thực tế, Bob sử dụng một thuật toán mã hóa đối xứng nào đó để mã hóa văn bản cần gửi và chỉ sử dụng RSA để mã hóa khóa để giải mã (thông thường khóa ngắn hơn nhiều so với văn bản).

Phương thức này cũng tạo ra những vấn đề an ninh mới. Một ví dụ là cần phải tạo ra khóa đối xứng thật sự ngẫu nhiên. Nếu không, kẻ tấn công (thường ký hiệu là Eve) sẽ bỏ qua RSA và tập trung vào việc đoán khóa đối xứng.

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

3.1. Giải thuật RSA

Sơ đồ giải thuật:



Hình 3. 1. 1. Giải thuật RSA

3.2. Mô phỏng quá trình mã hoá tin nhắn bằng RSA

3.2.1. 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:

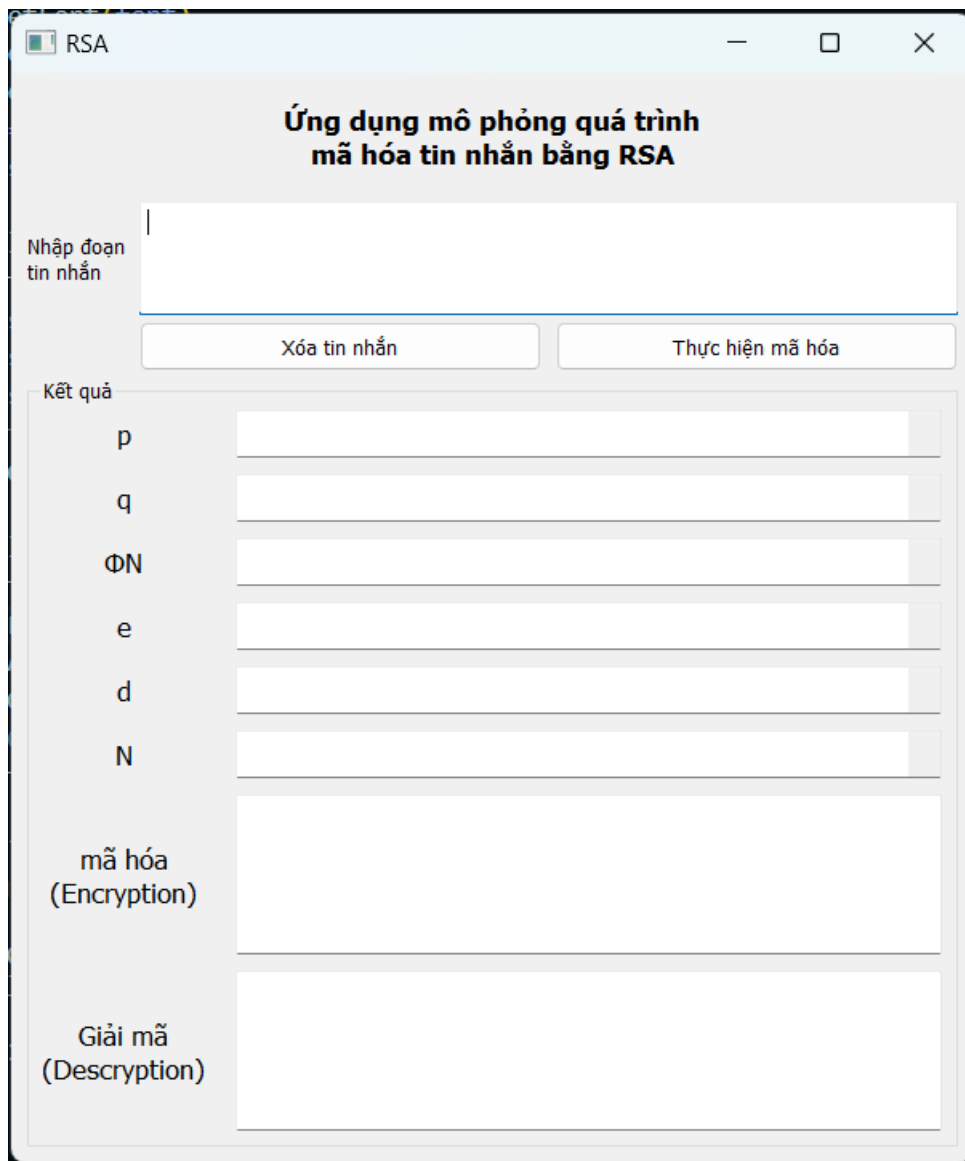
```

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:*

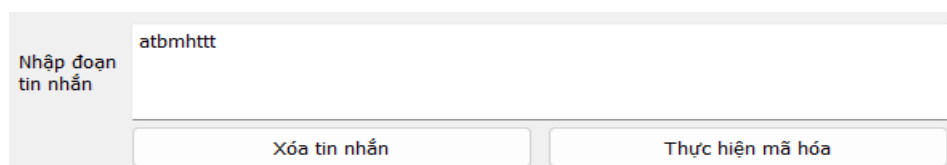


The screenshot shows a window titled "RSA" with the subtitle "Ứng dụng mô phỏng quá trình mã hóa tin nhắn bằng RSA". The interface includes a text input field for "Nhập đoạn tin nhắn" (Enter message segment), a "Xóa tin nhắn" (Clear message) button, and a "Thực hiện mã hóa" (Perform encryption) button. Below these, there is a section labeled "Kết quả" (Result) containing input fields for parameters p , q , ΦN , e , d , and N . At the bottom, there are two large text areas labeled "mã hóa (Encryption)" and "Giải mã (Decryption)".

Hình 3. 2. 1. 1. Giao diện mô phỏng

3.2.2. Kết quả

Đầu tiên ta nhập đoạn tin nhắn cần được mã hoá:



This screenshot shows the same interface as Figure 3.2.1.1, but the "Nhập đoạn tin nhắn" (Enter message segment) input field now contains the text "atbmhttt". The "Xóa tin nhắn" (Clear message) and "Thực hiện mã hóa" (Perform encryption) buttons remain visible below the input field.

Hình 3. 2. 2. 1. Nhập bản rõ

Ở đây, ta nhập đoạn tin nhắn là “atbmhttt”. Sau đó, nhấn vào “Thực hiện mã hoá”.

Kết quả	
p	199
q	233
ΦN	45936
e	133
d	25213
N	46367

Hình 3. 2. 2. 2. Các khoá

Đoạn code thực hiện lấy ngẫu nhiên các số nguyên tố cho 2 số “q” và “p” có giá trị xấp xỉ nhau, sau đó tính N theo công thức $q \cdot p = N$.

Tiếp đến tính $\varphi(n)$ bằng công thức $(p-1) \cdot (q-1) = \varphi(n)$.

Kế đến, thuật toán tìm 1 số nhiên “e” với điều kiện $1 < e_A < \varphi(n)$, $\text{gcd}(e_A, \varphi(n))=1$.

Đồng thời thực hiện tìm kiếm “d” theo thuật toán Euclide mở rộng.

mã hóa (Encryption)	17844 32951 27958 46078 34841 32951 32951 32951
------------------------	-------------------------------------------------

Hình 3. 2. 2. 3. Mã hoá

Dựa vào bảng ASCII kết hợp với “e” và “N” mà thuật toán cung cấp, chúng ta có được đoạn mã hoá như trên.

Để giải mã đoạn mã trên ta dùng “d” và “N” từ thuật toán, ta giải mã được đoạn tin nhắn ban đầu.

Giải mã (Description)	atbmhttt
--------------------------	----------

Hình 3. 2. 2. 4. Giải mã

KẾT LUẬN

Các vấn đề đạt được

Với mục tiêu đề ra ban đầu, đề án hiện tại đã đạt được các nội dung sau:

- Tìm hiểu về quá trình hình thành hệ mật mã
- Tìm hiểu thế nào là mã khoá công khai
- Tìm hiểu được mã hoá bất đối xứng RSA cũng như cách RSA hoạt động
- Nâng cao kiến thức về hệ mật mã

Hướng phát triển trong tương lai

Trong đề tài này chỉ mô phỏng quá trình mã hóa RSA cho một chuỗi tin nhắn văn bản. Tuy nhiên, nếu được áp dụng phương pháp này vào các ứng dụng thực tiễn, ta có thể tạo ra các ứng dụng tin nhắn mã hóa hoặc các hệ thống tin nhắn nhằm bảo mật thông tin người dùng cũng như bảo mật nội dung tin nhắn giữa các người dùng. Một số ứng dụng tin nhắn có độ bảo mật mã hóa cao trong thực tế có thể kể đến như Telegram, Viber, WhatsApp. Trong tương lai có thể nhóm sẽ xây dựng một ứng dụng nhắn tin mã hóa tương tự như vậy.

Hạn chế

Trong quá trình nghiên cứu, tìm hiểu và thực nghiệm nhóm có tham khảo một số tài liệu tuy chung mục tiêu nhưng cách thức triển khai lại khác nhau nên đề án có thể có nhiều sai sót không thể tránh khỏi.

TÀI LIỆU THAM KHẢO

- [1] TCVN 7635:2007 - Kỹ Thuật Mật M. - Chữ K. Số;
- [2] RSA PKCS#7 v1.5: March, 1998. Cryptographic Message Syntax Standard , RSA security inc. <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-7/pkcs-7v1-5.pdf>;
- [3] RSA PKCS#1 v2.1: June 14, 2002. RSA Cryptography Standard, RSA security inc. <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.pdf>;
- [4] RSA PKCS#11 v 2.40: Mark, 2014. RSA Cryptographic Token Interface, RSA security inc. <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/pkcs-1v2-40.pdf>;
- [5] RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. <https://www.ietf.org/rfc/rfc5280.txt>;
- [6] RFC 3125: Electronic Signature Policies <https://tools.ietf.org/html/rfc3125>;
- [7] RFC 3379: Delegated Path Validation and Delegated Path Discovery Protocol Requirements. <https://tools.ietf.org/html/rfc3379>.