HỌC VIỆN KỸ THUẬT QUÂN SỰ
KHOA CÔNG NGHỆ THÔNG TIN
****

ĐỒ ÁN MÔN HỌC: AN TOÀN VÀ BẢO MẬT HỆ THỐNG THÔNG TIN

ĐỀ TÀI: MÃ HÓA CÔNG KHAI – MÃ HÓA RSA

Giáo viên : Tống Minh Đức

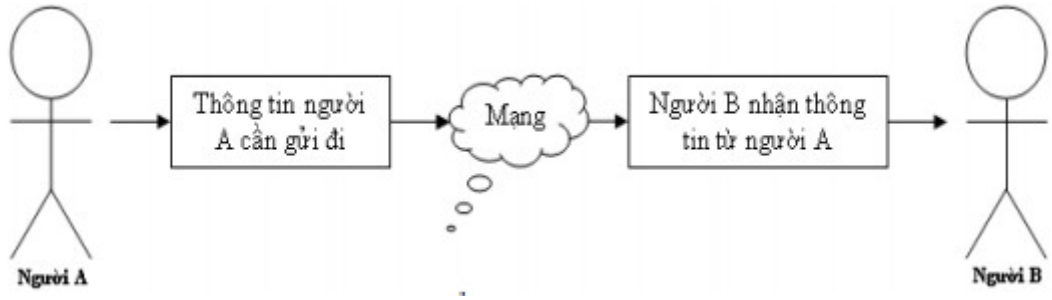
Sinh viên thực hiện : Đồng Tố Trung

MỤC LỤC

Đề tài: Mã hóa công khai – Mã hóa RSA

TÌM HIỂU CHUNG VỀ HỆ MÃ HÓA

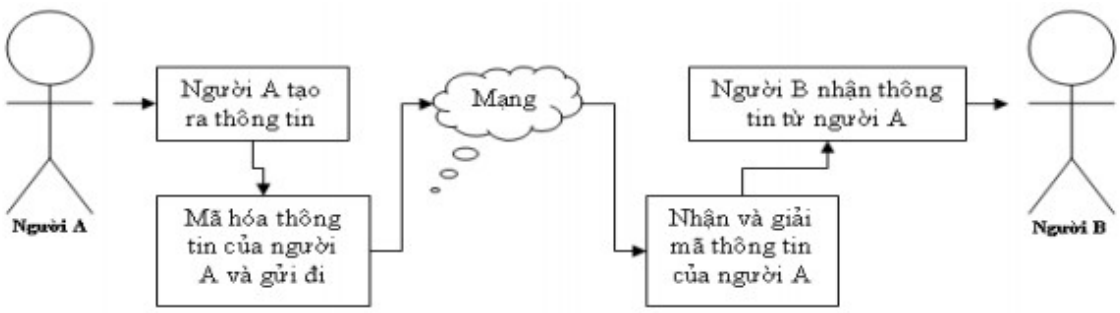
Trong mọi lĩnh vực kinh tế, chính trị, xã hội, quân sự… luôn có nhu cầu trao đổi thông tin giữa các cá nhân, các công ty, tổ chức, hoặc giữa các quốc gia với nhau. Ngày nay, với sự phát triển của công nghệ thông tin đặt biệt là mạng internet thì việc truyền tải thông tin đã dễ dàng và nhanh chóng hơn.



- Hình 1: Việc trao đổi thông tin được thực hiện qua các bước sau

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

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 nghành khoa học nghiên cứu về mã hóa thông tin được phát triển. Việc mã hóa 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: 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.

HỆ MÃ HÓA CÔNG KHAI

I. PHÂN BIỆT HỆ MÃ HÓA BÍ MẬT VÀ HỆ MÃ HÓA CÔNG KHAI.

Mã hóa bí mật: Thông tin sẻ 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.

Mã hóa bí mật

Mã hóa công khai: sử dụng 2 key là *public key* và *private key*

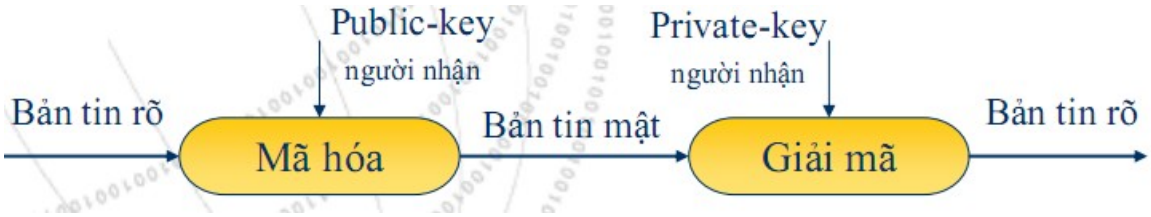
Public key: Được sử dụng để mã hoá những thông tin mà ta muốn chia sẻ với bất cứ ai. Chính vì vậy ta có thể tự do phân phát nó cho bất cứ ai mà ta cần chia sẻ thông tin ở dạng mã hoá.

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

=> Nghĩa là mỗi người sẽ giữ 2 key: + Một dùng để mã hóa, key này được công bố rộng rãi

+ Một 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.



Mô hình mã hóa công khai

II. NGUYÊN TẮC CẤU TẠO CỦA HỆ MÃ HÓA CÔNG KHAI.

Hệ mã khóa công khai được xây dựng dựa trên các hàm được gọi là hàm 1 phía hay hàm 1 chiều (one – way functions).

Hàm một chiều f: $X \rightarrow Y$ là một hàm mà nếu biết $x \in X$ ta có thể dễ dàng tính được

$y = f(x)$. Nhưng với y bất kỳ $\in Y$ việc tìm $x \in X$ sao cho $y = f(x)$ là khó. Có nghĩa là việc tìm hàm ngược f^{-1} là rất khó.

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 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 khoá), nó thật dễ dàng mở hộp thư. Hệ mã hoá công khai cũng tương tự như vậy.

TÌM HIỂU VỀ MÃ HÓA RSA

I. LỊCH SỬ RA ĐỜI CỦA THUẬT TOÁN RSA.

RSA được **Rivest**, **Shamir** và **Adleman** phát triển, là một thuận toán mật mã hóa khóa công khai. Đây là thuật toán đầu tiền phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa. 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 lĩnh vực truyền thông.

II. MÔ HÌNH THỰC HIỆN

1. Mô tả sơ lược

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.

Có thể mô phỏng trực quan một hệ mật mã khoá công khai như sau : Bob muốn gửi cho Alice một thông tin mật mà Bob muốn duy nhất Alice có thể đọc được. Để làm được điều này, Alice gửi cho Bob một chiếc hộp có khóa đã mở sẵn và giữ lại chìa khóa. Bob nhận chiếc hộp, cho vào đó một tờ giấy viết thư bình thường và khóa lại (như loại khoá thông thường chỉ cần sập chốt lại, sau khi sập chốt khóa ngay cả Bob cũng không thể mở lại được-không đọc lại hay sửa thông tin trong thư được nữa). Sau đó Bob gửi chiếc hộp lại cho Alice. Alice mở hộp với chìa khóa của mình và đọc thông tin trong thư. Trong ví dụ này, chiếc hộp với khóa mở đóng vai trò khóa công khai, chiếc chìa khóa chính là khóa bí mật.

2. Tạo khóa

Giả sử Alice và Bob cần trao đổi thông tin bí mật thông qua một kênh không an toàn (ví dụ như Internet). Với thuật toán RSA, Alice đầu tiên cần tạo ra cho mình cặp khóa gồm khóa công khai và khóa bí mật theo các bước sau:

1. Chọn 2 số nguyên tố lớn p và q với $p \neq q$, lựa chọn ngẫu nhiên và độc lập.
2. Tính: $n = pq$.
3. Tính: giá trị hàm số Ơle $\phi(n) = (p-1)(q-1)$.

4. Chọn một số tự nhiên *e* sao cho $1 < e < \phi(n)$ và là số nguyên tố cùng nhau với $\phi(n)$.
5. Tính: *d* sao cho $de \equiv 1 \pmod{\phi(n)}$.

Một số lưu ý:

- 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 $d \mod (p-1)(q-1)$.
- 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 mod (p-1)* và *d mod (q-1)* (thường được gọi là *dmp1* và *dmq1*),
- *(1/q) mod p* (thường được gọi là *iqmp*)

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.

Alice gửi khóa công khai cho Bob, và giữ bí mật khóa cá nhân của mình. 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.

3. Mã hóa

Giả sử Bob muốn gửi đoạn thông tin *M* cho Alice. Đầu tiên Bob 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 Bob có *m* và biết *n* cũng như *e* do Alice gửi. Bob 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 Bob gửi *c* cho Alice.

4. Giải mã

Alice nhận *c* từ Bob và biết khóa bí mật *d*. Alice có thể tìm được *m* từ *c* theo công thức sau:

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

Biết *m*, Alice 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:

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

hay:

$$c^d \equiv m \pmod n.$$

5. Ví dụ

Sau đây là một ví dụ với những số cụ thể. Ở đây chúng ta sử dụng những số nhỏ để tiện tính toán còn trong thực tế phải dùng các số có giá trị đủ lớn.

Lấy:

$p = 61$	— số nguyên tố thứ nhất (giữ bí mật hoặc hủy sau khi tạo khóa)
$q = 53$	— số nguyên tố thứ hai (giữ bí mật hoặc hủy sau khi tạo khóa)
$n = pq =$	— môđun (công bố công khai)

3233
e = 17 — số mũ công khai
d = 2753 — số mũ bí mật

Khóa công khai là cặp (e, n). Khóa bí mật là d. Hàm mã hóa là:

$$\text{encrypt}(m) = m^e \bmod n = m^{17} \bmod 3233$$

với m là văn bản rõ. Hàm giải mã là:

$$\text{decrypt}(c) = c^d \bmod n = c^{2753} \bmod 3233$$

với c là văn bản mã.

Để mã hóa văn bản có giá trị 123, ta thực hiện phép tính:

$$\text{encrypt}(123) = 123^{17} \bmod 3233 = 855$$

Để giải mã văn bản có giá trị 855, ta thực hiện phép tính:

$$\text{decrypt}(855) = 855^{2753} \bmod 3233 = 123$$

Cả hai phép tính trên đều có thể được thực hiện hiệu quả nhờ giải thuật bình phương và nhân.

6. Chuyển đổi văn bản rõ

Trước khi thực hiện mã hóa, ta phải thực hiện việc chuyển đổi văn bản rõ (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 m^e 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 bản rõ bằng cách tạo ra một bảng tra giữa bản rõ và bản mã. Khi gặp một bản mã, kẻ tấn công sử dụng bảng tra để tìm ra bản rõ 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 NUL 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 255^3 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 bản rõ 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 bản rõ (một bản rõ sẽ có thể tương ứng với nhiều bản mã tuỳ thuộc vào cách chuyển đổi).

Một số tiêu chuẩn, chẳng hạn như PKCS, đã được thiết kế để chuyển đổi bản rõ trước khi mã hóa bằng RSA. Các phương pháp chuyển đổi này bổ sung thêm bít 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 bản rõ. 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 bản rõ 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).

7. Tạo chữ ký số cho đoạn văn bản

Thuật toán RSA còn được dùng để tạo chữ ký số cho văn bản. Giả sử Alice muốn gửi cho Bob một văn bản có chữ ký của mình. Để làm việc này, Alice tạo ra một giá trị băm (hash value) của văn bản cần ký và tính giá trị mũ *d* mod *n* của nó (giống như khi Alice thực hiện giải mã). Giá trị cuối cùng chính là chữ ký điện tử của văn bản đang xét. Khi Bob nhận được văn bản cùng với chữ ký điện tử, anh ta tính giá trị mũ *e* mod *n* của chữ ký đồng thời với việc tính giá trị băm của văn bản. Nếu 2 giá trị này như nhau thì Bob biết rằng người tạo ra chữ ký biết khóa bí mật của Alice và văn bản đã không bị thay đổi sau khi ký.

Cần chú ý rằng các phương pháp chuyển đổi bản rõ (như RSA-PSS) giữ vai trò quan trọng đối với quá trình mã hóa cũng như chữ ký điện tử và không được dùng khóa chung cho đồng thời cho cả hai mục đích trên.

PHÂN TÍCH THỜI GIAN PHÁ MÃ

Phá mã là nỗ lực giải mã văn bản đã được mã hóa không biết trước khóa bí mật.

Các phương pháp phá mã RSA:

- Vét cạn : thử tất cả các mã có thể.
- Phân tích toán học
  + Phân tích n thành 2 số nguyên tố p và q.
  + Xác định trực tiếp $\phi(n)$ không thông qua p và q.
  + Xác định trực tiếp d không thông qua $\phi(n)$
- Phân tích thời gian
  + Dựa trên việc đo thời gian giải mã
  + có thể ngăn ngừa bằng cách làm nhiễu

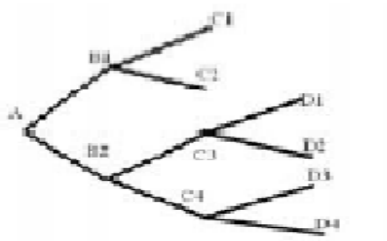
1. Phương pháp vét cạn

Thử tất cả các khóa có thể cho đến khi xác định được nguyên bản từ bản mã.
Ưu điểm : thử qua tất cả các trường hợp
Nhược điểm :

- Tốn thời gian, nhiều động tác thừa, tốn không gian nhớ
- Không thể hiện tư duy khoa học

Ví dụ:

- Cho sơ đồ như hình bên
- Yêu cầu :
  - Tìm tất cả các đường đi từ A đến D4
- Sử dụng PP vét cạn như sau:



Hoạt động tìm đường
Đi theo thứ tự từ trên xuống ta có :
+ Tại A => có 2 đường đi đến B1 và B2 => đi đến điểm B1
+ Tại B1 có 2 đường đi => đi đến C1
+ Tại C1 không có đường đi => quay lại B1
+ Tại B1 có 2 đường đi, điểm C1 đã đi qua nên đi tiếp =>C2
  - Tại B1 có 2 đường đi nhưng cả 2 đều đã đi qua => quay lại điểm A

- Tại A có 2 đường đi, đường qua B1 đã đi => B2
- ….

Cứ thế cho đến khi tìm được đường đi từ A -> D4

2. Phương pháp phân tích toán học

Chọn 2 số nguyên tố lơn p và q
Tính n= p*q
Tính φ(n) = (p-1) * (q-1)
Có thể dùng định lý Trung Hoa để giảm bớt phần tính toán
Chọn ngẫu nhiên khóa mã e sao cho USCLN(e, φ(n)) = 1 với 1 < e < φ(n)
Giải phương trình sau để tìm ra khóa giải mã d sao cho ed ≡ 1 mod (φ(n))
Vấn đề chọn p và q :

- p và q phải là những số nguyên tố lớn, ít nhất là cỡ 100 chữ số.
- p và q phải lớn xấp xỉ nhau (về độ dài cùng 100 chữ số chẳng hạn).

Ví dụ: Cho các số nguyên tố p=17 & q=11.

Tính $n = pq$, n = 17×11=187
Tính Φ(n)=(p–1)(q-1)=16×10=160
Chọn e *:* gcd(e,160)=1; Lây e=7
Xác định d*:* de=1 mod 160 và d < 160
Giá trị cần tìm là d=23, vì 23×7=161= 10×160+1

3. Phương pháp phân tích thời gian

So với DES thì RSA có tốc độ chậm hơn rất nhiều và kích thước của khóa mật lớn hơn rất nhiều. nếu p và q cỡ 300 bits thì n cỡ 600 bits. Phép nâng lên lũy thừa là khá chậm hơn so với n lớn, đặc biệt là nếu sử dụng phần mềm. Tốc độ hiện thời :

- Sử dụng phần cứng đặc chủng : n cỡ 507 bits thì đạt được tốc độ khoảng 200kb/s
- Phần mềm : n cỡ 512 bits thì đạt được tốc độ khoảng 11kb/s.

Giải thuật tốt nhất vẫn là phương pháp sàng số. Một ước lượng về thời gian thực hiện của giải thuật là:

$$L(n) \approx 10^{9.7+\frac{1}{50}\log n}{}_2$$

Log_2^n cho biết số bit cần biểu diễn n, số cần phân tích thành thừa số nguyên tố. Người ta đã ước lượng thấy, với n = 200, L(n) ≈ 55 ngàn năm.

Đối với khả năng thực hiện bằng xử lý song song, một trong các kết quả tốt nhất về phân tích một số có 129 chữ số, phân bố tính toán trên toàn mạng Internet và mất trọn 3 tháng. Ngày nay, với những ứng dụng có độ đòi hỏi an toàn đặc biệt cao người ta sử dụng đại lượng Modulo của RSA này lên đến 1024 bits và thậm chí 2048 bits. Trong các bài toán mã hoá công khai, chúng ta sử dụng nhiều phép toán lũy thừa với số mũ lớn. Như vậy cần có thuật toán nhanh hiệu quả đối với phép toán này. Trước hết ra phân tích số mũ cơ số 2, xét biểu diễn nhị phân của số mũ, sau đó sử dụng thuật toán bình phương và nhân. Khái niệm được xây dựng trên phép lặp cơ sở bình phương và nhân để nhận được kết quả mong muốn. Độ phức tạp của thuật toán là O(log2 n) phép nhân với số mũ n.

Ví dụ:

75 = 74.71 = 3.7 = 10 mod 11

vì 72 = 7.7 = 49 = 5 mod 11

74 = 72.72 = 5.5 = 3 mod 11

3129 = 3128.31 = 5.3 = 4 mod 11

Phân tích số mũ theo cơ số 2:

Trước hết ta chuyển số mũ từ cơ số 10 sang cơ số 2: (11)10 = (1011)2. Sau đó tính toán như sau:

M11 = M1.2^3 + 0.2^2+ 1.2^1 + 1.2^0
= (M1.2^2 + 0.2^1+ 1.2^0)2M
= (M1.2^1 + 0.2^0)2M)2M
= ((M2)2 M)2M

Mã sử dụng lũy thừa của khoá công khai e, nếu giá trị của e nhỏ thì tính toán sẽ nhanh, nhưng dễ bị tấn công. Thường chọn e nhỏ hơn hoặc bằng 65537 (2^{16}-1), tức là độ dài khoá công khai là 16 bit. Chẳng hạn trong ví dụ trên ta có thể chọn e = 23 hoặc e = 7. Ta có thể tính mã hoá nhanh, nếu biết n=pq và sử dụng định lý phần dư Trung Hoa với các mẩu tin M theo các Modulo p và q khác nhau. Nếu khóa công khai e cố định thì cần tin tưởng rằng khi chọn n ta luôn có gcd(e,Φ(n)) = 1 Loại bỏ mọi p, q mà làm cho Φ(n) không nguyên tố cùng nhau với e. Có thể sử dụng định lý phần dư Trung Hoa để tính theo mod p và q, sau đó kết hợp lại để tìm ra bản rõ. Vì ở đây, người ta sử dụng khóa riêng biết được p và q, do đó có thể sử dụng kỹ thuật này. Nếu sử dụng định lý phần dư Trung Hoa để giải mã thì hiệu quả là nhanh gấp 4 lần so với giải mã tính trực tiếp.

ỨNG DỤNG CỦA RSA

1. Chữ ký điện tử (Digital Signature)

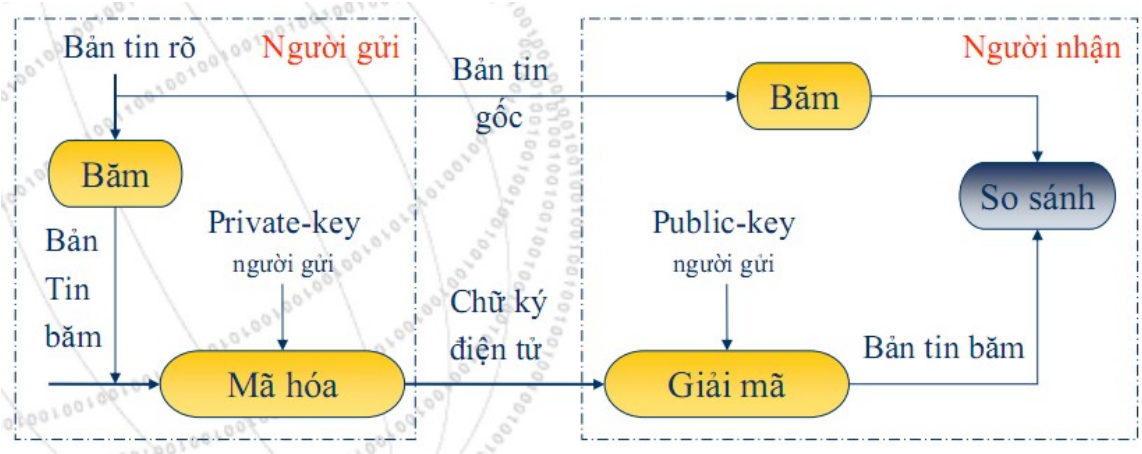
Hệ mã RSA có tính an toàn rất cao. Nhưng nhược điểm lớn là tốc độ mã hóa chậm (nhất là so với các hệ mã đối xứng có cùng độ an toàn). Bởi vậy nó chỉ được sử dụng với các văn bản ngắn, và thường dùng trong giao thức xác nhận chủ thể (chữ kí điện tử).
Chữ kí điện tử đảm bảo khi người nhận có được mật thư thì biết chắc chắn ai là tác giả bức thư đó. Và cũng đảm bảo việc không ai có thể mạo danh người khác để gửi thư.

Chữ kí điện tử và chữ kí tay có chung đặc điểm là rất khó xảy ra trường hợp trùng.
Để làm được điều đó, khi anh U muốn gửi bức thư P cho anh V . Đầu tiên anh U dùng khóa lập mã Ev (được công khai của anh V) để mã hóa P thu được Ev (P). Sau đó dùng khóa giải mã của mình là Du (bí mật) để tính Du (Ev (P)) = C và gửi đi.

Sau khi nhận được mật thư C , anh V sẽ dùng khóa lập mã của anh U là Eu (công khai) đế tính Eu **(**C) = Eu (Du (Ev (P))) = Ev (P) (do Eu là hàm ngược của Du) Cuối cùng dùng khóa giải mã bí mật Dv để tính ra Dv (Ev (P)) = P chính là bức thư ban đầu .

Rõ ràng với cách này chúng ta chỉ có thể áp dụng với những văn bản ngắn, còn khi văn bản dài chúng ta phải áp dụng một phương pháp biến thể từ phương pháp trên, đó là sử dụng hàm băm.

Đó là một hàm được công khai trên toàn hệ thống. Khi cần gửi văn bản, người ta sẽ gửi kèm theo bản giá trị băm của văn bản, sau khi nhận được người ta sẽ băm văn bản lại lần nữa và so sánh với giá trị băm của bên gửi, nếu trùng khớp thì có thể khẳng định văn bản đã không bị thay đổi trên đường đi …



Mô hình chữ kí điện tử

2. SSL (Secure Socket Layer)

SSL (Secure Socket Layer) là giao thức đa mục đích được thiết kế nhằm mã hóa toàn bộ thông tin đến/ đi giữa hai chương trình ứng dụng trên một cổng định trước (socket 443). Giao thức SSL được hình thành và phát triển đầu tiên năm 1994 bởi nhóm nghiên cứu Netscape và ngày nay trở thành chuẩn bảo mật thực hành trên mạng Internet.

Giao thức SSL được hình thành và phát triển đầu tiên năm 1994 bởi nhóm nghiên cứu Netscape và ngày nay trở thành chuẩn bảo mật thực hành trên mạng Internet. Phiên bản hiện nay là SSL 3.0 và đang tiếp tục được bổ sung hoàn thiện.