TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÀI TẬP LỚN NHẬP MÔN AN TOÀN THÔNG TIN

Đề tài: Tìm hiểu về Chữ ký số

Lớp	:	132648
Mã học phần	:	IT4015
Nhóm	:	7
Giảng viên hướng dẫn	:	PGS. TS. Nguyễn Linh Giang

Danh sách thành viên nhóm:

Họ và tên	Mã số sinh viên
Nguyễn Thế Vũ	20194214
Trương Văn Hiển	20194276
Lương Thái Nam	20194126
Phạm Ngọc Dũng	20194257

Hà Nội, tháng 7 năm 2022

PHÂN CÔNG THÀNH VIÊN TRONG NHÓM

STT	Họ và tên	MSSV	Email	Công việc	Tỷ lệ % công việc làm
1	Lương Thái Nam	20194126	nam.lt194126@sis.hust.edu.vn	• Giới thiệu về chữ ký số. • Kiến trúc chữ ký số tổng quát.	25%
2	Nguyễn Thế Vũ	20194214	vu.nt194214@sis.hust.edu.vn	• Phân chia công việc trong nhóm. • Thuật toán chữ ký số RSA RABIN.	25%
3	Trương Văn Hiển	20194276	hien.tv194276@sis.hust.edu.vn	• An toàn trong chữ ký số. • Tổng hợp kết quả thành báo cáo.	25%
4	Phạm Ngọc Dũng	20194257	dung.pn194257@sis.hust.edu.vn	• Phương pháp khảo sát tính an toàn của chữ ký số.	25%

MỞ ĐẦU

Trong các hoạt động thương mại điện tử cũng như việc xây dựng một nền hành chính điện tử, không thể không tính đến mức độ chính xác, an toàn của các bản thông báo điện tử được gửi đi và đến cũng như việc xác thực đối tượng gửi bản thông báo đó. Điều này nói lên sự cần thiết của việc xác thực và chữ ký số.

Hiện nay, Bộ Thông tin và Truyền thông, Bộ Công thương, Bộ Tài chính và Ngân hàng Nhà nước Việt Nam đã được Chính phủ cho phép triển khai chữ ký số và xác thực trong thanh toán điện tử từ năm 2006. Hiện nay, Hàn Quốc cũng đang giúp ta triển khai hạ tầng cơ sở khóa công khai PKI trong Chính phủ điện tử.

Tất cả kết quả trên chủ yếu là được chuyển giao từ bên ngoài. Xét về lĩnh vực an ninh quốc gia, chúng ta sẽ đặt câu hỏi: Mức độ an toàn của chữ ký số và tính xác thực của văn bản có đảm bảo yêu cầu của chúng ta không khi mà chúng ta phải nhập ngoại hoàn toàn dây chuyện công nghệ?

Trên cơ sở đánh giá mức độ an toàn của hệ thống, nhóm chúng em đã chọn đề tài: "**Tìm hiểu về Chữ ký số**" làm đối tượng để nghiên cứu phục vụ cho Bài tập lớn môn học.

Do khả năng còn hạn chế, đặc biệt là khả năng về toán học cho nên mặc dù nhóm đã có nhiệm cố gắng nhằm hoàn thành tốt nhất nhiệm vụ của mình nhưng không tránh khỏi còn có nhiều thiết sót. Nhóm em rất mong nhận được những nhận xét thẳng thắn, chi tiết đến từ thầy để tiếp tục hoàn thiện hơn nữa. Cuối cùng, nhóm xin được gửi lời cảm ơn đến thầy **PGS.TS. Nguyễn Linh Giang** đã hướng dẫn nhóm trong suốt quá trình hoàn thiện Bài tập lớn. Xin chân thành cảm ơn thầy.

NHẬN XÉT

(Của Giảng viên hướng dẫn)

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU VỀ CHỮ KÝ SỐ

1.1. Khái niệm

Chữ kí số (Digital Signature) là những thông tin đi kèm với dữ liệu nhằm chứng thực nguồn gốc và nội dung của văn bản. Chữ kí số được dựa trên cơ sở lý thuyết hệ mã hoá công khai (public key cryptography), còn gọi là mã hóa bất đối xứng (asymetric cryptography), được tạo ra để giải quyết câu hỏi: "Làm thế nào để định nghĩa một chữ ký cho các văn bản số, với các tính chất tương tự như chữ ký viết tay ?"

Mã hóa công khai: Sử dụng 2 khóa có quan hệ toán học với nhau, khóa công khai (public key dùng để mã hóa, được công bố rộng rãi. Khóa còn lại là khóa bí mật (private key) dùng để giải mã, được giữ bí mật.

Mã hóa bất đối xứng: Cơ sở toán học của hệ mã hóa bất đối xứng là dùng những hàm một chiều, tức là những hàm để tính theo chiều thuận thì dễ còn theo chiều ngược lại thì với không khả thi với hệ thống máy tính hiện tại.

Chữ ký số bao gồm 3 thành phần:

1. Thuật toán tạo ra khóa.
2. Hàm tạo chữ ký là hàm tính toán chữ ký trên cơ sở khóa mật và dữ liệu cần ký.
3. Hàm kiểm tra chữ ký là hàm kiểm tra xem chữ ký đã cho có đúng với khóa công cộng không. (Khóa này mọi người có quyền truy cập cho nên mọi người đều có thể kiểm tra được chữ ký).

1.2. Các yêu cầu của chữ ký số

- Tính xác thực: người nhận có thể chứng minh được văn bản được ký bởi gửi.

- Tính toàn vẹn: người nhận có thể chứng minh được không có ai sửa đổi văn bản đã được ký.

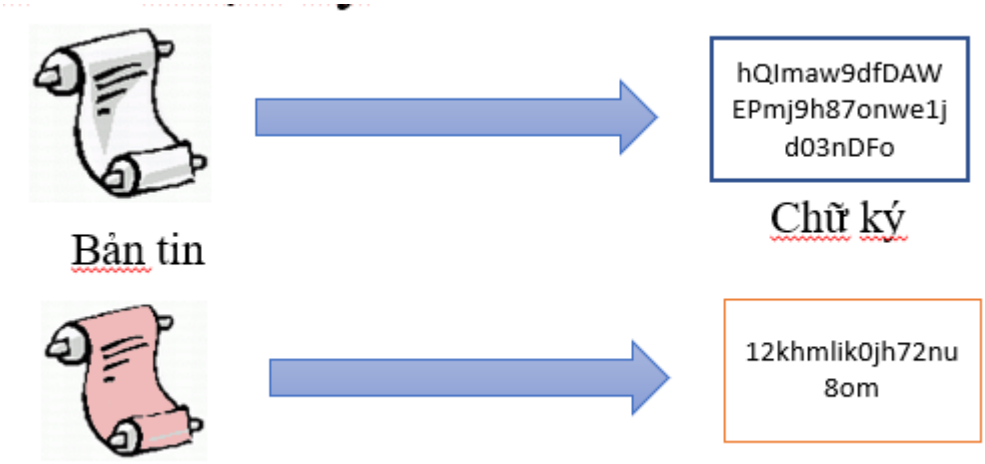
- Không thể tái sử dụng: mỗi chữ ký chỉ có giá trị trên 1 văn bản.

- Không thể giả mạo.

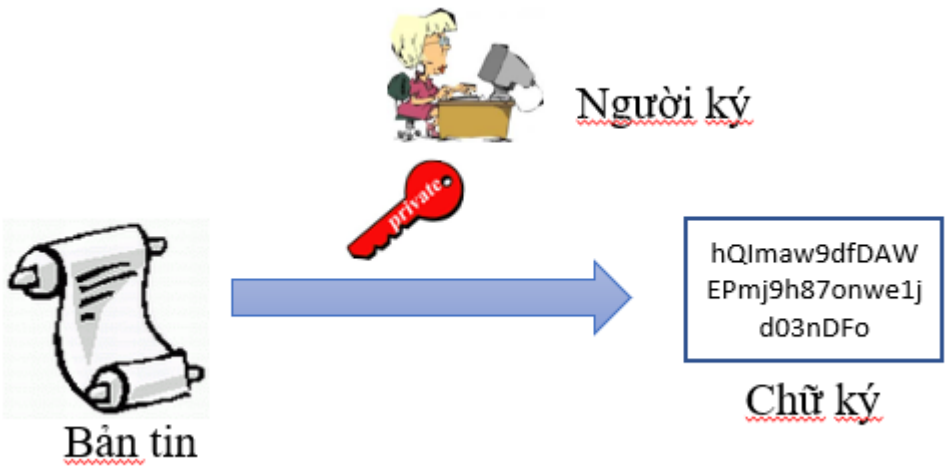
- Chống từ chối: người gửi không thể phủ nhận được hành động ký vào văn bản.

1.3. Tính chất của chữ ký số

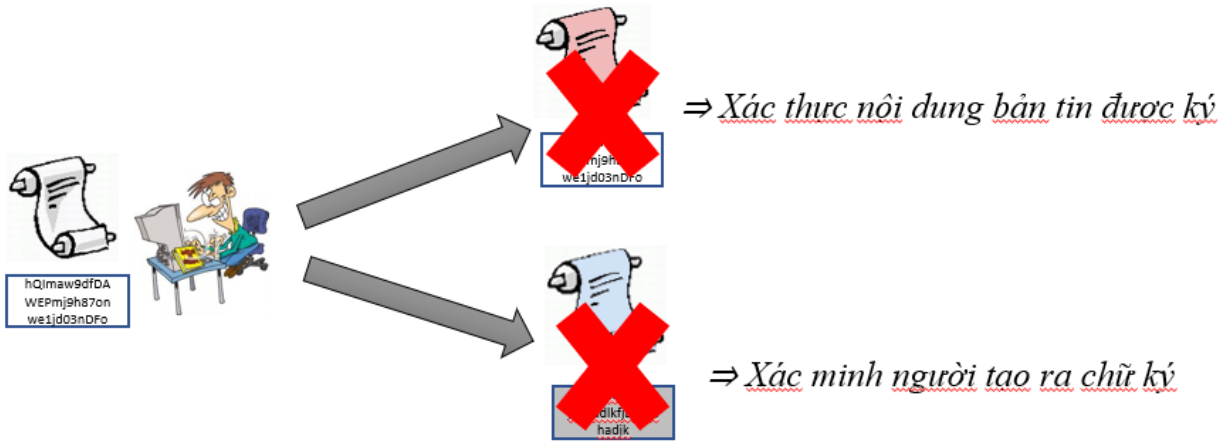
1. Là một chuỗi ký tự, có nội dung phụ thuộc vào nội dung bản tin được ký: khó thay đổi, khó dùng lại. Do đó có thể xác thực nội dung bản tin được ký



2. Sử dụng thông tin mà chỉ có người ký mới có: Khó giả mạo, khó chối từ. Do đó có thể xác minh người ký



3. Gần như không thể giả mạo chữ ký



1.4. So sánh chữ ký viết tay và chữ ký số

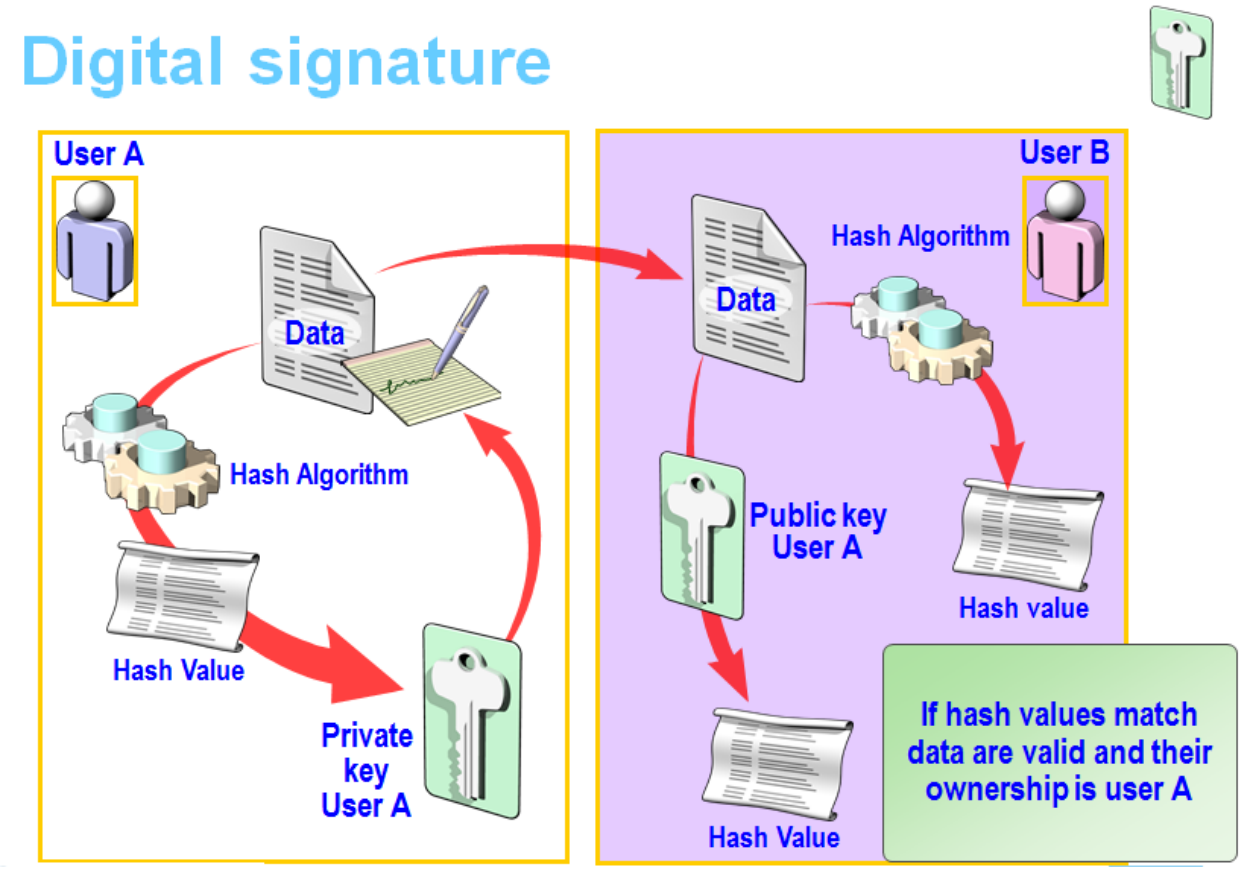
Chữ ký viết tay	Chữ ký số
Chữ ký cố định	Chữ ký thay đổi theo nội dung văn bản
Gắn liền với nội dung được ký	Có thể tách khỏi nội dung được ký

1.5. Hàm băm (Hash function)

Hàm băm là hàm toán học chuyển đổi thông điệp (message) có độ dài bất kỳ (hữu hạn) thành một dãy bít có độ dài cố định (tùy thuộc vào thuật toán băm). Dãy bít này được gọi là thông điệp rút gọn (message disgest) hay giá trị băm (hash value), đại diện cho thông điệp ban đầu.

Hàm băm SHA-1: Thuật toán SHA-1 nhận thông điệp ở đầu vào có chiều dài k<264 bit, thực hiện xử lý và đưa ra thông điệp thu gọn (message digest) có chiều dài cố định 160 bits. Quá trình tính toán cũng thực hiện theo từng khối 512 bits, nhưng bộ đệm xử lý dùng 5 thanh ghi 32-bits. Thuật toán này chạy tốt với các bộ vi xử lý có cấu trúc 32 bits.

CHƯƠNG 2: KIẾN TRÚC CHỮ KÝ SỐ TỔNG QUÁT



2.1. Quá trình ký (Bên gửi)

Tính toán chuỗi đại diện (message digest/ hash value) của thông điệp sử dụng một giải thuật băm (Hashing algorithm).

Chuỗi đại diện được ký sử dụng khóa riêng (Private key) của người gửi và 1 giải thuật tạo chữ ký (Signature/ Encryption algorithm). Kết quả chữ ký số (Digital signature) của thông điệp hay còn gọi là chuỗi đại diện được mã hóa (Encryted message digest).

Thông điệp ban đầu (message) được ghép với chữ ký số(Digital signature) tạo thành thông điệp đã được ký (Signed message).

Thông điệp đã được ký (Signed message) được gửi cho người nhận.

2.2. Quá trình kiểm tra chữ ký (Bên nhận)

Tách chữ ký số và thông điệp gốc khỏi thông điệp đã ký để xử lý riêng

Tính toán chuỗi đại diện MD1 (message digest) của thông điệp gốc sử dụng giải thuật băm (là giải thuật sử dụng trong quá trình ký).

Sử dụng khóa công khai (Public key) của người gửi để giải mã chữ ký số -> chuỗi đại diện thông điệp MD2.

So sánh MD1 và MD2:

- Nếu MD1 = MD2 -> chữ ký kiểm tra thành công. Thông điệp đảm bảo tính toàn vẹn và thực sự xuất phát từ người gửi (do khóa công khai được chứng thực).
- Nếu MD1 ≠ MD2 -> chữ ký không hợp lệ. Thông điệp có thể đã bị sửa đổi hoặc không thực sự xuất phát từ người gửi.

2.3. Nhược điểm

Bởi vì tài liệu cần ký thường có chiều dài khá dài. Một biện pháp để ký là chia tài liệu ra các đoạn nhỏ và sau đó ký lên từng đoạn và ghép lại.

Nhưng phương pháp có nhược điểm là chữ ký lớn, thứ hai là ký chậm vì hàm ký là các hàm mũ, thứ ba là chữ ký có thể bị đảo loạn các vị trí không đảm tính nguyên vẹn của tài liệu.

Chính vì điều đó mà khi ký thì người ta ký lên giá trị hàm hash của tài liệu, vì giá trị của hàm hash luôn cho chiều dài xác định.

2.4. Chức năng

Xác thực được nguồn gốc tài liệu: Tùy thuộc vào từng bản tin mà có thể thêm các thông tin nhận dạng, như tên tác giả, thời gian…

Tính toàn vẹn tài liệu: Vì khi có một sự thay bất kỳ vô tình hay cố ý lên bức điện thì gía trị hàm hash sẽ bị thay đổi và kết quả kiểm tra bức điện sẽ không đúng.

Chống từ chối bức điện: Vì chỉ có chủ của bức điện mới có khóa mật để ký bức điện.

2.5. Nguy cơ

Tội phạm có thể giả mạo chữ ký tương ứng với văn bản đã chọn.

Tội phạm thử chọn bức điện mà tương ứng với chữ ký đã cho.

Tội phạm có thể ăn trộm khóa mật và có thể ký bất kỳ một bức điện nào nó muốn giống như chủ của khóa mật.

Tội phạm có thể giả mạo ông chủ ký một bức điện nào đó.

Tội phạm có thể đổi khóa công cộng bởi khóa của mình.

CHƯƠNG 3: THUẬT TOÁN CHỮ KÝ SỐ RSA RABIN

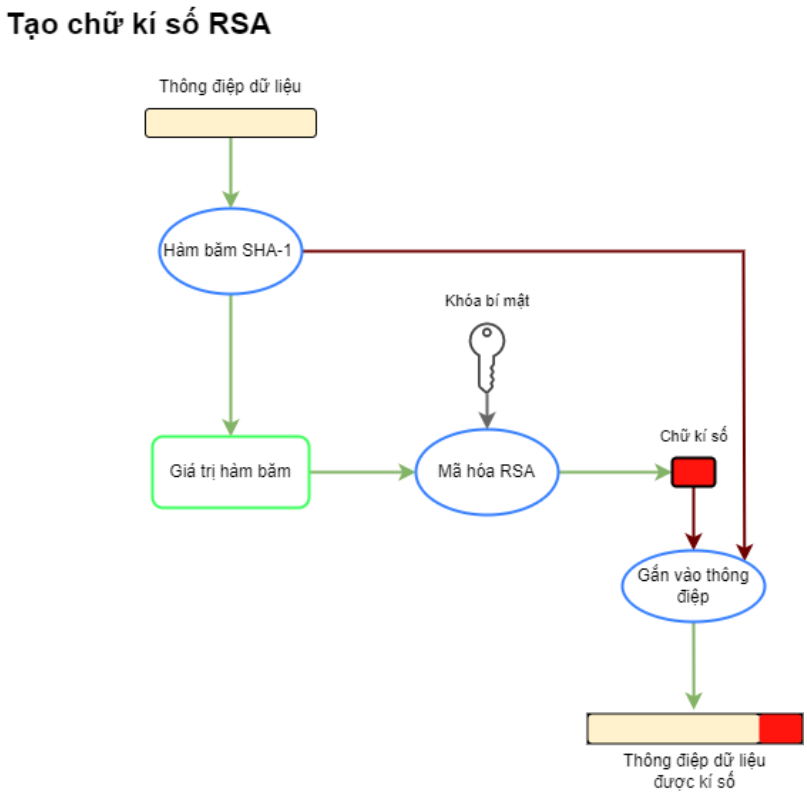
3.1. Sơ đồ thuật toán chữ ký số RSA

3.1.1. Quá trình ký (Bên gửi)

Thông điệp sẽ được ký bằng cách mã hóa văn bản bằng khóa riêng (Private key) với thuật toán mã hóa RSA. Kết quả chữ ký số (Digital signature) của thông điệp sẽ là bản mã của thông điệp. Nếu để mã hóa 1 thông điệp dài thì sẽ tốn rất nhiều thời gian trong cả việc mã hóa, giải mã và truyền tin, vì vậy trước khi đưa văn bản để mã hóa, người ta sẽ tính toán trước 1 chuỗi đại diện (hash value) của thông điệp sử dụng một giải thuật băm (Hashing Algorithm) SHA-1 hoặc MD5.

Sau đó thông điệp ban đầu sẽ được ghép với chữ ký số tạo thành thông điệp đã được ký.

Thông điệp đã được ký sẽ được gửi cho người nhận.



3.1.2. Quá trình kiểm tra (Bên nhận)

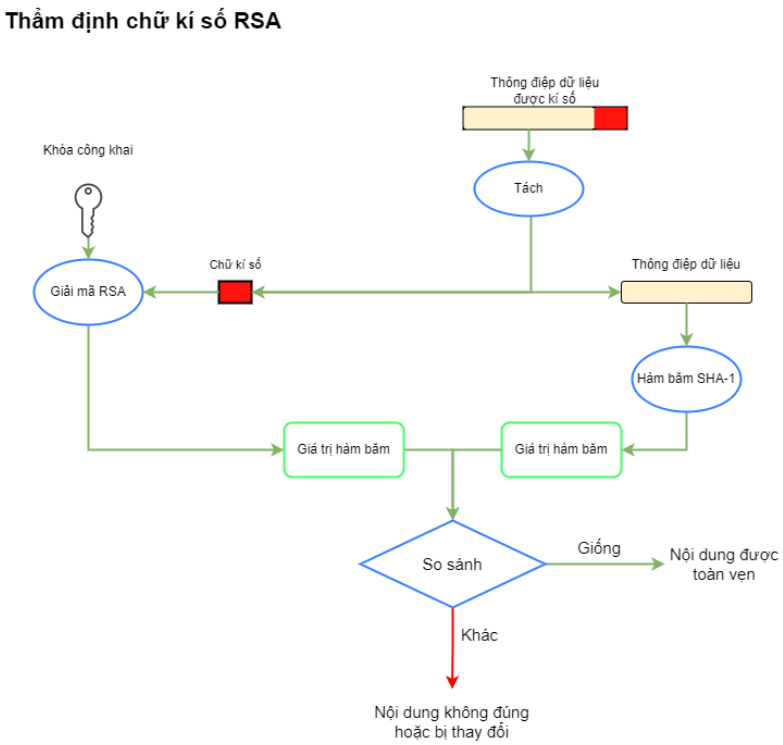
Bên nhận sẽ tách phần chữ ký số RSA và thông điệp đã ký để xử lý.

Tính toán chuỗi đại diện MD1 (Message digest) của thông điệp gốc sử dụng giải thuật băm (là giải thuật sử dụng trong quá trình ký là SHA-1).

Sử dụng khóa công khai (Public key) của người gửi để giải mã chữ ký số RSA => chuỗi đại diện thông điệp MD2.

So sánh MD1 và MD2:

- Nếu MD1 = MD2 => chữ ký kiểm tra thành công. Thông điệp đảm bảo tính toàn vẹn và thực sự xuất phát từ người gửi (do khóa công khai được chứng thực).
- Nếu MD1 ≠ MD2 => chữ ký không hợp lệ. Thông điệp có thể đã bị sửa đổi hoặc không thực sự xuất phát từ người gửi.



3.2. Thuật toán mã hóa RSA

RSA là một hệ mã hóa bất đối xứng được phát triển bởi Ron Rivest, Adi Shamir và Leonard Adleman (tên của nó cũng chính là tên viết tắt của 3 tác giả

này) và được sử dụng rộng rãi trong công tác mã hoá và công nghệ chữ ký điện tử. Trong hệ mã hóa này, public key có thể chia sẻ công khai cho tất cả mọi người. Hoạt động của RSA dựa trên 4 bước chính: sinh khóa, chia sẻ key, mã hóa và giải mã.

Thuật toán RSA có hai khóa:

- Khóa công khai (Public key): Được công bố rộng rãi cho mọi người và được dùng để mã hóa.
- Khóa bí mật (Private key): Những thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa bí mật tương ứng.

3.2.1. Quá trình sinh khóa

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

$$m^{ed} \equiv m \bmod n$$

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.

Cụ thể, khóa của RSA được sinh như sau:

1. Chọn 2 số nguyên tố p và q.
2. Tính $n = p * q$.

   Sau này, n sẽ được dùng làm modulus trong cả public key và private key.

3. Tính một số giả nguyên tố bằng phi hàm Euler như sau:

$$\Phi(n) = (p - 1) * (q - 1)$$

   Giá trị này sẽ được giữ bí mật.

4. Chọn một số ngẫu nhiên e ($0 < e < \Phi(n)$) sao cho:

$$GCD(e, \Phi(n)) = 1$$

5. Tính $d = e^{-1}$ hay $de \equiv 1 \ (mod\ \Phi(n))$ bằng cách dùng thuật toán Euclide tìm số tự nhiên x sao cho $d = \frac{x * \Phi(n) + 1}{e}$

6. Lấy (n, e) làm khóa công khai.

7. Lấy d làm khóa bí mật.

3.2.2. Quá trình mã hóa và giải mã

Trong thuật toán mã hóa RSA, người gửi sẽ mã hóa bằng khóa công khai (n, e) và người nhận sẽ giải mã bằng khóa bí mật d.

Các bước mã hóa và giải mã như sau:

1. A nhận khóa công khai của B.

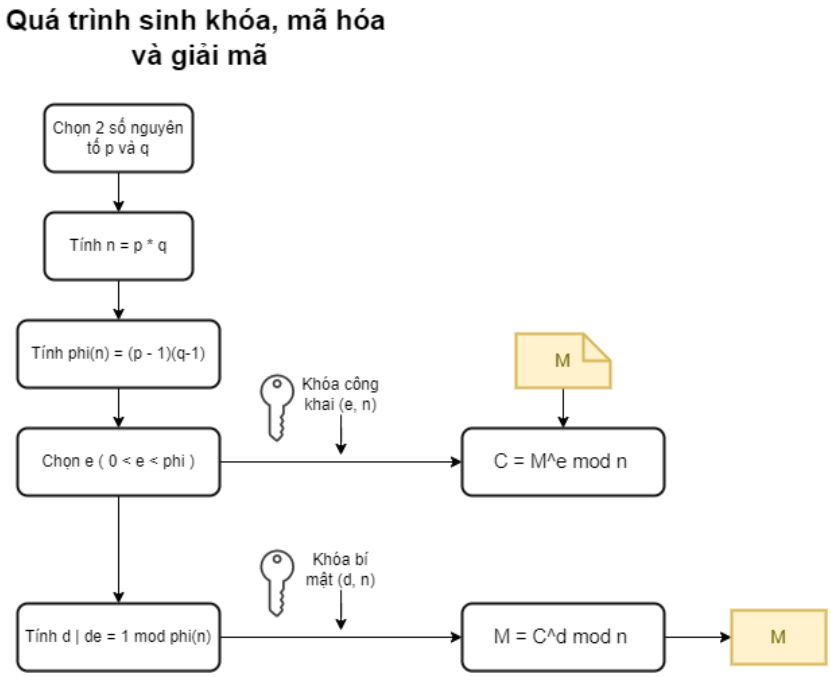
2. A biểu diễn thông tin cần giải thành số m $(0 \leq m \leq n - 1)$

3. Tính $c = m^e \ mod\ n$

4. Gửi c cho B.

5. B giải mã bằng cách tính $m = c^d \ mod\ n$

   => m là thông tin nhận được.



3.2.3. Phân tích độ an toàn

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. Phá mã một phần phải được ngăn chặn bằng các phương pháp chuyển đổi bản rõ an toàn.

Bài toán RSA là bài toán tính căn bậc e môđun n (với n là hợp số): tìm số m sao cho me=c mod 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 = pq 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).

Tại thời điểm năm 2005, số lớn nhất có thể được phân tích ra thừa số nguyên tố có độ dài 663 bít với phương pháp phân tán trong khi khóa của RSA có độ dài từ 1024 tới 2048 bít. Một số chuyên gia cho rằng khóa 1024 bít có thể sớm bị phá vỡ (cũng có nhiều người phản đối việc này). Với khóa 4096 bít 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 bít 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 bít, 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 bít. Vì vậy hiện nay người ta khuyến cáo sử dụng khóa có độ dài tối thiểu 2048 bít.

3.3. Giải thuật hàm băm SHA-1

Giải thuật băm là 1 giải thuật biến đổi chuỗi bit đầu thành 1 chuỗi bit khác tương ứng sao cho thỏa mãn các tính chất sau:

- Các hàm hash là hàm 1 chiều, vì vậy dù có được hash cũng không thể biết được bản tin gốc như thế nào.
- Độ dài hash là cố định và thường rất nhỏ, vì vậy chữ số sẽ không chiếm quá nhiều dung lượng.
- Hai chuỗi bit khác nhau sẽ có giá trị băm khác nhau. Nhờ vào tính chất này giá trị hash còn có thể dùng để kiểm tra lại bản tin nhận được có nguyên vẹn hay không?

SHA-1 là một trong những thuật toán băm mã hóa, được dùng trong việc kiểm tra tính toàn vẹn của dữ liệu ở phía người nhận. SHA-1 checksum được so sánh giữa người cung cấp và người nhận, dữ liệu được cho là toàn vẹn nếu hai chuỗi checksum là giống nhau.

SHA-1 cũng giống như MD5, nhạy cảm ở đầu vào, bất kỳ sự thay đổi bit nào cũng dẫn đến kết quả khác hoàn toàn. SHA-1 vượt trội hơn trong vấn đề bảo mật nhưng cũng vì thế cần có thời gian nhiều hơn để xử lý.

3.3.1. Các tham số

$a, b, c, \ldots, h$	Các biến là các từ w-bit được sử dụng để tính toán giá trị hàm băm $H^{(i)}$
$H^{(i)}$	Giá trị băm thứ i
$H_j^{(i)}$	Từ thứ j của giá trị hàm băm thú i
K_t	Giá trị không đổi được sử dụng cho lần lặp t của phép tính băm
k	Số số 0 được thêm vào thông điệp trong bước padding

l	Độ dài của thông điệp M $(bits)$
m	Số bits của khối thông điệp $M^{(i)}$
M	Thông điệp để băm
$M^{(i)}$	Khối thông điệp i, với m $bits$
$M_j^{(i)}$	Từ thứ j của khối thông điệp thứ i
n	Số bits được chuyển trong 1 từ với toán tử tương ứng
N	Số khối được padded
T	Từ w-bit tạm thời trong tính toán giá trị băm
w	Số bits của 1 từ
W_t	Tù thứ t của lịch thông điệp

3.3.2. Các toán tử

∧	Toán tử bitwise AND
∨	Toán tử bitwise OR
$\oplus$	Toán tử bitwise XOR
¬	Toán tử đối nghịch
+	Cộng thêm modulo 2^w
$\ll$	Toán tử dịch bits trái, VD: $x \ll n$ thực hiện xóa n bits bên trái và thêm n số 0 vào bên phải
$\gg$	Toán tử dịch bits phải, VD: $x \gg n$ thực hiện xóa n bits bên phái và thêm n số 0 vào bên trái
$\boldsymbol{ROTL^n(x}$	Toán tử quay vòng trái được định nghĩa bằng $ROTL^n(x) = (x \ll n) \vee (x \gg w - n)$.

$ROTR^n(x$ Toán tử quay vòng phải được định nghĩa bằng $ROTR^{\,n}\,(x) = (x \gg n) \vee (x \ll w - n)$.

3.3.3. Thuật toán

Tiền xử lý

1. Tạo giá trị khởi đầu

$$H_0^{(0)} = 67452301$$
$$H_1^{(0)} = efcdab89$$
$$H_2^{(0)} = 98badcfe$$
$$H_3^{(0)} = 10325476$$
$$H_4^{(0)} = c3d2e1f0$$

2. Thông điệp được padding, thêm số 0 vào đầu để đủ số bits
3. Mã giả

For $i = 1\ to\ N$:

{

1. Chuẩn bị lịch thông điệp

$$\mathrm{Wt} = \begin{cases} \mathrm{Mt}\,0 \le \mathrm{t} \le 15 \\ \mathrm{ROTL1}\,(\mathrm{Wt}-3 \oplus \mathrm{Wt}-8 \oplus \mathrm{Wt}-14 \oplus \mathrm{Wt}-16\,) 16 \le \mathrm{t} \le 79 \end{cases}$$

2. Tạo 5 biến làm việc

$a = H_0^{i-1}$
$b = H_1^{i-1}$
$c = H_2^{i-1}$
$d = H_3^{i-1}$
$e = H_4^{i-1}$

3. For $t = 0\ to\ 79$:

{

$T = ROTL^5(a) + f_t(b,c,d) + e + K_t + W_t$
$e = d$
$d = c$
$c = ROTL^{30}(b)$
$b = a$

$a = T$

}

4. Tính toán giá trị thứ i của hàm băm $H^{(i)}$:

$$H_0^{(i)} = a + H_0^{(i-1)}$$
$$H_1^{(i)} = b + H_1^{(i-1)}$$
$$H_2^{(i)} = c + H_2^{(i-1)}$$
$$H_3^{(i)} = d + H_3^{(i-1)}$$
$$H_4^{(i)} = e + H_4^{(i-1)}$$

}

Cuối cùng ghép nối các thông điệp ta có M :

$$H_0^{(N)} || H_1^{(N)} || H_2^{(N)} || H_3^{(N)} || H_4^{(N)}$$

3.4. Thuật toán chữ ký số RABIN

Trong mật mã học, thuật toán chữ ký Rabin là một phương pháp chữ ký số ban đầu được đề xuất bởi Michael O. Rabin vào năm 1978.

Thuật toán chữ ký Rabin là một trong những lược đồ chữ ký số đầu tiên được đề xuất. Bằng cách giới thiệu việc sử dụng hàm băm như một bước thiết yếu trong việc ký kết, đây là thiết kế đầu tiên đáp ứng tiêu chuẩn bảo mật hiện đại chống lại sự giả mạo, không thể tha thứ hiện sinh dưới sự tấn công tin nhắn đã chọn, giả sử các tham số được chia tỷ lệ phù hợp.

Chữ ký rabin giống với chữ ký RSA với 'số mũ e = 2', nhưng điều này dẫn đến sự khác biệt về chất lượng cho phép thực hiện hiệu quả hơn và đảm bảo bảo mật liên quan đến khó khăn của thừa số nguyên, chưa được chứng minh cho RSA.

3.4.1. Quá trình ký (Bên gửi)

Số nguyên $n = p * q$ trong đó p, q là 2 số nguyên tố khác nhau với $p, q \equiv 3 \ (mod\ 4)$ và $b \in Z_n^*$. Khóa bí mật do người ký giữ là bộ (n, p, q, b) và khóa công khai cho người xác thực chữ ký là (n, b).

Hàm băm $Hash : \{0,1\}^{\infty} \rightarrow \{0,1\}^{h}$

Hàm đổi xâu bit sang số nguyên từ biểu diễn nhị phân $Code$:

$$\{0,1\}^{\infty} \rightarrow Z$$

Thông điệp được ký như sau :

1. Lấy ngẫu nhiên xâu k bit R
2. Tính $u = Code\,(\,Hash\,(\,M \,||\, R\,)\,)$.
3. Giải phương trình : $x * (\,x + b\,) = u\,(\,mod\ n\,)$

   Nếu vô nghiệm, quay lại bước 1.

   Ngược lại, lấy s là một nghiệm của phương trình trên.
4. Trả về cặp (s, R)
5. Hợp với thông điệp thành (M, s, R) và gửi đi.

3.4.2. Quá trình kiểm tra (Bên nhận)

1. Tính $u = Code(\,Hash\,(\,M \,||\, R\,))$
2. Tính $v = x * (\,x + b\,)\ với\ x = s$
3. So sánh $u\ với\ v$.

   Nếu giống nhau thì chấp nhận chữ ký, nếu khác thì không chấp nhận chữ ký.

3.4.3. Thuật toán tạo khóa bí mật

Khóa bí mật của khóa công khai (n, b) là 1 thừa số nguyên tố lẻ p * q của n, được chọn 1 cách ngẫu nhiên trong tập số nguyên tố. Cho $d = (b/2)\ mod\ n$, $d_p = (b/2)\ mod\ p$, and $d_q = (b/2)\ mod\ q$. Để tạo chữ ký cho thông điệp m, lấy 1 chuỗi u với k-bits ngẫu nhiên và tính $u = H(m, R)$. Nếu $u + d^2$ là một modulo nonresidue bậc hai của n (nghĩa là tồn tại 1 số x, sao cho $x^2 \equiv u + d^2\ (mod\ n)$ sau đó người ký sẽ bỏ u và thử lại. Nếu không, người ký sẽ tính toán:

$$x_p = (-d_p \pm \sqrt{c + {d_p}^2}) \bmod p$$

$$x_q = (-d_q \pm \sqrt{c + {d_q}^2}) \bmod q,$$

sử dụng một thuật toán tiêu chuẩn để tính toán căn bậc hai modulo một số nguyên tố — chọn $p \equiv q \equiv 3$ làm cho nó dễ dàng nhất. Căn bậc hai không phải là duy nhất và các biến thể khác nhau của lược đồ chữ ký đưa ra các lựa chọn khác nhau về căn bậc hai; trong mọi trường hợp, người ký phải đảm bảo không tiết lộ hai gốc khác nhau cho cùng một hàm băm c. Người ký sau đó sử dụng định lý phần còn lại của Trung Quốc để giải hệ thống:

$$x \equiv x_q \ (mod\ n\ )$$

$$x \equiv x_p \ (mod\ n\ )$$

Người ký tiết lộ (x, R) .Tính đúng đắn của quy trình ký kết sau đây bằng cách đánh giá $x(x + b) - H(m, R)\ mod\ p, q\ với\ x$ như được xây dựng.

CHƯƠNG 4: AN TOÀN TRONG CHỮ KÝ SỐ

4.1. Tính an toàn của chữ ký số

4.1.1. Cơ chế bảo mật tuyệt đối

Chữ ký số được cho là có khả năng đảm bảo an toàn vô cùng cao nhờ cơ chế bảo mật tuyệt đối thông qua những đặc điểm sau đây:

4.1.1.1. Khả năng xác định nguồn gốc

Hệ thống mã hóa công khai được sử dụng trong chữ ký ố cho phép mã hóa văn bản với khóa bí mật được sở hữu bởi chủ nhân của chữ ký số đó. Một văn bản nếu muốn được ký số thì cần phải được mã hóa bằng hàm băm, có nghĩa là văn bản sẽ được chia nhỏ ra thành các chuỗi có độ dài cố định và ngắn hơn văn bản gốc, sau đó mã hóa bằng khóa bí mật của chủ sở hữu chữ ký số.

Để kiểm tra chữ ký số, phía người nhận cần giải mã bằng khóa công khai và kiểm tra với hàm băm của văn bản đã được nhận. Nếu 2 giá trị này khớp nhau thì phía người nhận hoàn toàn có thể an tâm rằng văn bản này được xuất phát từ người sờ hữu khóa bí mật.

Chính bởi đặc điểm này mà chữ ký số có khả năng xác định nguồn gốc tuyệt đối, giúp loại bỏ khả năng bị kẻ gian lợi dụng lừa đảo, giả mạo thông tin.

4.1.1.2. Tính toàn vẹn

Chữ ký số có tính toàn vẹn tuyệt đối bởi văn bản một khi đã được ký thì không thể bị sửa đổi trong quá trình gửi. Lý do là vì nếu văn bản có bất kỳ thay đổi nào dù là nhỏ nhất, thì hàm băm cũng sẽ bị thay đổi theo và bị phát hiện ngay lập tức. Chính vì vậy, văn bản khi được ký bằng chữ ký số sẽ được bảo toàn sự nguyên vẹn 100% mà không một bên thứ 3 nào có thể can thiệp vào để thay đổi nội dung của văn bản đó.

4.1.1.3. Tính không thể phủ nhận

Trong giao dịch, người nhận có thể từ chối thừa nhận một văn bản nào đó không phải do mình gửi. Để ngăn chặn trường hợp này xảy ra, người nhận có

thể yêu cầu người gửi phải ký số lên văn bản đó. Bởi khi phát sinh tranh chấp, người nhận sẽ dùng chính chữ ký số ngày của người gửi như một bằng chứng xác minh làm căn cứ để bên thứ ba xem xét và giải quyết. Vì vậy, chữ ký số được xem là công cụ an toàn nhất để chứng thực nguồn gốc của văn bản trong trường hợp cần thiết.

4.1.2. Độ an toàn của hệ thống RSA

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. Phá mã một phần phải được ngăn chặn bằng các phương pháp chuyển đổi bản rõ an toàn.

Vì vậy muốn xây dựng hệ thống RSA an toàn thì n = p*q phải là một số đủ lớn, để không có khả năng phân tích nó về mặt tính toán. Để đảm bảo an toàn nên chọn các số nguyên tố p và q từ 100 chữ số trở lên.

Dưới đây là bảng thời gian phân tích mã RSA:

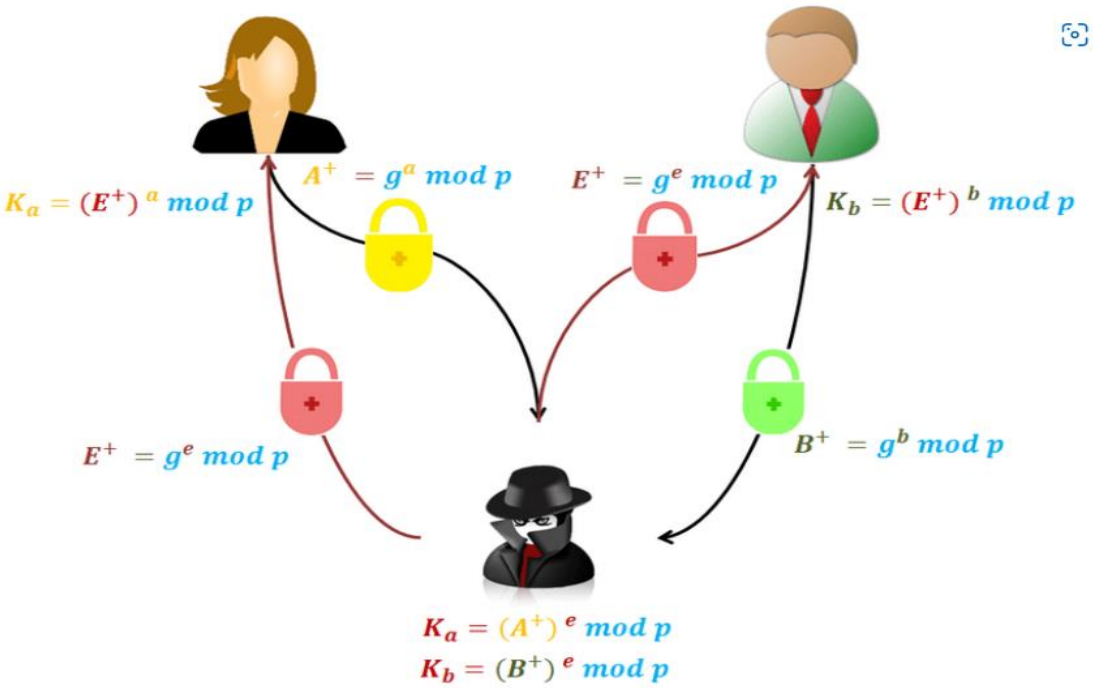
Số các chữ số trong số được phân tích	Thời gian phân tích
50	4 giờ
75	104 giờ
100	74 năm
200	4000 năm
300	500.000 năm
500	4×10^{25} năm

Cách thức phân phối khóa công khai là một trong những yếu tố quyết định đối với độ an toàn của RSA. Vấn đề này sinh ra một lỗ hổng gọi là “Man-in-the-middle attack” (MiMA - Tấn công vào giữa)

- Khi A và B trao đổi thông tin thì C có thể gửi cho A một khóa bất kì để A tin rằng đó là khóa công khai của B gửi.

- Sau đó C sẽ giải mã và đánh cắp được thông tin. Đồng thời mã hóa lại thông tin theo khóa công khai của B và gửi lại cho B.

- Về nguyên tắc, cả A và B đều không phát hiện được sự can thiệp của C.



4.2. Các dạng tấn công chữ ký số

4.2.1. Tấn công dạng 1: Tìm cách xác định khóa bí mật

4.2.1.1. Bị lộ một trong các giá trị: p, q, $\phi(n)$

Nếu trong quá trình tạo khóa mà người sử dụng vô tình để lộ nhân tử p, q hoặc $\phi(n)$ ra ngoài thì kẻ tấn công sẽ dễ dàng tính được khóa bí mật d theo công thức:

$$d \equiv e^{-1} \bmod \phi(n)$$

Biết được khóa bí mật, kẻ tấn công sẽ giả mạo chữ ký của người dùng.

➔ Giải pháp phòng tránh: Quá trình tạo lập khóa phải được tiến hành ở một nơi kýn đáo, bí mật. Sau khi thực hiện xong thì phải giữu cẩn thận khóa bí mật d, đồng thời hủy hết các giá trị trung gian (p, q, $\phi(n)$).

4.2.1.2. Tấn công dựa theo khóa công khai n và e của người ký

Kẻ tấn công sẽ tìm cách phân tích giá trị n ra hai thừa số nguyên tố p và q. Từ đó sẽ tính được $\phi(n)$ = (p-1)(q-1), cuối cùng tính được khóa bí mật d.

➔ Giải pháp phòng tránh: Nên chọn số nguyên p và q đủ lớn để việc phân tích n thành tích của hai thừa số nguyên tố là khó có thể thực hiện được trong thời gian thực. Trong thực tế, người ta thường sinh ra các số lớn (ít nhất 100 chữ số), sau đó kiểm tra tính nguyên tố của nó.

4.2.1.3. Khi nhiều người cùng sử dụng chung "modun n"

Khi có k người cùng đăng ký sử dụng chữ ký RSA, trung tâm phân phối khóa sẽ sinh ra 2 số nguyên tố p và q, rồi tính số modun n = p*q. Sau đó, sinh ra các cặp khóa mã hóa/giải mã $\{e_i, d_i\}$. Trung tâm sẽ cấp cho người đăng ký thứ i khóa bí mật d_i tương ứng, cùng với các thông tin như: n, danh sách khóa công khai $\{e_i\}$ (i = 1, …, k).

Lúc này, bất kỳ ai có thông tin công khai như trên đều có thể:

- Mã hóa văn bản M để gửi cho người đăng ký thứ i bằng cách sử dụng thuật toán mã hóa RSA với khóa mã hóa e_i:

$$Y = M^{e_i} \bmod n$$

- Người đăng ký thứ i có thể ký văn bản M bằng cách tính chữ ký:

$$S_i = M^{d_i} \bmod n$$

Bất cứ ai cũng có thể xác thực rằng M được ký bởi người đăng ký thứ i bằng cách tính $S_i^{e_i} \bmod n$ và so sánh với M.

➔ Giải pháp phòng tránh: Sử dụng giá trị modun n khác nhau cho mỗi người tham gia.

4.2.1.4. Sử dụng giá trị "modun n" nhỏ

Trong sơ đồ chữ ký RSA, công thức để tính giá trị chữ ký y trên bản rõ x như sau:

$$y = x^a \pmod n \text{ với } y \in A, x \in P, P = A = \mathbb{Z}_n$$

Lúc này, kẻ tấn công có thể tính được khóa bí mật d theo công thức:

$$d = log_x\, y \pmod n$$

do các giá trị x, y, n là công khai. Đây chính là việc giải bài toán logarit rời rạc trên vành $\mathbb{Z}_n$. Bởi vậy, nếu như giá trị modun n mà nhỏ thì bằng cách áp

dụng các thuật toán đã trình bày ở trên, kẻ tấn công có thể tìm ra được khóa bí mật d.

➔ Giải pháp phòng tránh: Nên chọn các số nguyên tố p và q đủ lớn để việc giải bài toán logarit rời rạc trên vành $\mathbb{Z}_n$ là khó có thể thực hiện được trong thời gian thực.

4.2.1.5. Sử dụng các tham số (p-1) hoặc (q-1) có các ước nguyên tố nhỏ

Nếu ta bất cẩn trong việc chọn các tham số p và q để cho (p-1) hoặc (q-1) có các ước nguyên tố nhỏ thì sơ đồ chữ ký sẽ trở nên mất an toàn.

Bởi vì khi (p-1) hoặc (q-1) có các ước nguyên tố nhỏ thì ta có thể dùng thuật toán (p-1) của Pollar để phân tích giá trị modun n thành thừa số một cách dễ dàng.

➔ Giải pháp phòng tránh: Chọn các tham số p và q sao cho (p-1) và (q-1) phải có các ước nguyên tố lớn.

4.2.2. Tấn công dạng 2: Giả mạo chữ ký (không tính trực tiếp khóa bí mật)

4.2.2.1. Người gửi G gửi tài liệu x cùng chữ ký y đến người nhận N

Sẽ có 2 cách xử lý:

- Ký trước, mã hóa sau

Người gửi G ký vào x trước bằng chữ ký $y = S_G(x)$, sau đó mã hóa x và y nhận được $z = e_G(x, y)$ rồi gửi z cho N.

Nhận được z, N giải mã z để nhận được x, y. Tiếp theo, kiểm tra chữ ký $V_N(x, y) = true$?

- Mã hóa trước, ký sau

Người gửi G mã hóa x trước bằng $u = e_G(x)$, sau đó ký vào u bằng chữ ký $v = S_G(u)$ rồi gửi (u, v) cho N.

Nhận được (u, v), N giải mã u nhận được x. Tiếp theo kiểm tra chữ ký $V_N(u, v) = true$?

4.2.2.2. Giả sử H lấy trộm được thông tin trên đường truyền từ G đến N

- Trong trường hợp ký trước, mã hóa sau thì H lấy được z. Trong trường hợp mã hóa trược, ký sau thì H lấy được (u, v).

- Để tấn công vào x trong cả hai trường hợp, H đều phải giải mã thông tin lấy được.
- Để tấn công vào chữ ký, thay bằng chữ ký giả mạo, thì xảy ra hai trường hợp:
  - Trường hợp ký trước, mã hóa sau: Để tấn công chữ ký y, H phải giải mã z, mới nhận được y.
  - Trường hợp mã hóa trước, ký sau: Để tấn công chữ ký v, H đã có sẵn v', lúc này H chỉ việc thay v bằng v'.

H thay chữ ký v trên u, bằng chữ ký của H là $v' = S_H(u)$ rồi gửi (u, v') đến N.

Khi nhận được v', N kiểm thử thấy sai, gửi phản hồi lại cho G.

G có thể chứng minh đó là chữ ký giả mạo.

G gửi chữ ký đúng v cho N, nhưng quá trình truyền tin sẽ bị chậm lại.

Như vậy, trong trường hợp mã hóa trước, ký sau thì H có thể giả mạo chữ ký mà không cần phải giải mã.

➔ Giải pháp phòng tránh: Hãy ký trước, sau đó mã hóa cả chữ ký.

CHƯƠNG 5: PHƯƠNG PHÁP KHẢO SÁT TÍNH AN TOÀN CỦA CHỮ KÝ SỐ

5.1. Phương pháp Random Oracle Model (ROM)

Cho X, Y là các tập hữu hạn. ROM có hàm băm h được chọn ngẫu nhiên từ tất cả các hàm từ X đến Y và ROM liên quan đến h. Hàm băm h có thể được coi là một bảng băm T_h xác định sự tương ứng của các phần tử trong X với các phần tử trong Y. Trong mô hình này, tất cả đều có quyền truy cập vào ROM. Khi giá trị băm của x được truy vấn, ROM trả lời giá trị y tương ứng trong T_h.

Trong hầu hết bài toán sử dụng ROM, các thuật toán tối ưu mô phỏng ROM bằng cách đưa vào các bài toán cụ thể. Đầu tiên, ta mô phỏng ROM với một bảng T ban đầu trống như sau: Khi giá trị băm của x được truy vấn, nếu tồn tại $(\tilde{x}, \tilde{y}) \in T$ sao cho $x = \tilde{x}$ thì trả về $\tilde{y}$; nếu không thì chọn tất cả $y \leftarrow Y$, thêm (x, y) vào bảng băm T và trả về y.

Ngoài ra, ta đề xuất một thuật toán RO khác để mô phỏng ROM. Ta quản lý một bảng băm T và một bảng L ban đầu trống. Bảng T thực hiện vai trò tương tự như trên, trong khi bảng L quản lý số phần tử trong X mà ánh xạ đến $y \in Y$. Ví dụ nếu tồn tại $(y, n) \in L$ thì có thể có đúng n các phần tử trong X ánh xạ tới $y \in Y$. Khi ta thêm (x, y) vào bảng băm T sao cho y chưa có trong T, ta cũng xác định số n của nghịch ảnh của y và thêm (y, n) vào bảng L.

Thuật toán RO(x):

1. Nếu tồn tại $(\tilde{x}, \tilde{y}) \in T$ sao cho $x = \tilde{x}$ thì trả về $\tilde{y}$.

2. Tính giá trị sau:

$$p = \frac{\sum_{(\tilde{y},\tilde{n})\in \mathbb{L}} (\tilde{n} - \#\mathbb{T}(\tilde{y}))}{\#X - \#\mathbb{T}}.$$

(p là xác suất để $(\tilde{x}, \tilde{y}) \in T$ với một số $\tilde{x}$.)

3. Tung một đồng xu với xác suất Pr[α = 0] = p

(Quyết định xem mô phỏng có trả về giá trị mới hay không. "α = 0" cho biết "không" và "α = 1" cho biết "có")

4. Nếu α = 0 thì chọn y theo phân phối sau và đến bước 8

$$y \leftarrow \mathcal{D},$$
$$\text{where } f_{\mathcal{D}}(y) = \frac{n - \#\mathbb{T}(y)}{\sum_{(\tilde{y},\tilde{n})\in\mathbb{L}}(\tilde{n} - \#\mathbb{T}(\tilde{y}))} \text{ for } (y,n) \in \mathbb{L}.$$

5. Nếu α ≠ 0 thì chọn tất cả y ← Y \ ∪ $_{(\tilde{y},\tilde{n})\in L}\{\tilde{y}\}$

6. Chọn n' theo phân phối nhị thức

$$n' \leftarrow \mathrm{B}(\#X - \sum_{(\tilde{y},\tilde{n})\in\mathbb{L}} \tilde{n} - 1, \frac{1}{\#Y - \#\mathbb{L}}).$$

(n' là số nghịch ảnh của y không bao gồm (x,y))

7. Đặt n = n' + 1 và thêm (y,n) vào L

8. Thêm (x,y) vào T và trả về y

5.2. ROM trong khảo sát tính an toàn của chữ ký số

Thay vì ký trực tiếp, ta yêu cầu người ký phải băm thông điệp trước khi ký. Chữ ký S thỏa mãn

$$s^e \bmod N = H \text{ (message)}$$

Giả sử A là thuật toán phá vỡ lược đồ chữ ký, nó lấy khóa công khai (N, e) và nó lấy ra bất kì (message, S').

Mục đích là thiết kế thuật toán có thể nhận một RSA (N, e, M) ngẫu nhiên và giúp ta xử lý lấy ra S sao cho $s^e \bmod N = M$.

Đầu tiên, ta sẽ gửi A một khóa công khai (N, e). Vì sử dụng ROM, A sẽ phải thực hiện hàm băm. Mỗi khi A hỏi để băm một thông điệp, ta sẽ chặn nó và gọi oracle, chọn 1 giá trị r ngẫu nhiên sao cho 0 <= r < N và trả lời truy vấn:

$$H\ (message) = M * r^e \bmod N$$

Bởi vì r ngẫu nhiên nên hàm băm cũng là ngẫu nhiên.

Ta giả sử A sẽ gửi một thông điệp giả mạo (message, S') thỏa mãn $s'^e \bmod N = H$ (message)

Nếu trường hợp này xảy ra, ta phải tìm giá trị r được dùng để tính H và in ra :

$$s = s' / r \bmod N$$

5.3. Khảo sát phương pháp trên vài biến thể của chữ ký số (ECDSA)

Với ECDSA, ECDSA không chỉ dựa vào một số ngẫu nhiên cho khóa sinh, mà còn cần một giá trị ngẫu nhiên mới k cho mỗi thông điệp đã ký. Việc tạo ra k giả ngẫu nhiên là vô cùng quan trọng với tính bảo mật của ECDSA.

Phiên bản sau của ECDSA là ECDSA* cũng vậy. Ta lấy ECDSA* với k=H'(K, M), H' là hàm băm. Một câu hỏi được đặt ra về việc thay thế k ngẫu nhiên trong ECDSA theo k.

Câu trả lời là có. Có thể nhận được nếu hàm băm H' như là ROM. Trong trường hợp đó, giá trị của k xuất hiện ngẫu nhiên cho kẻ tấn công, vì không có thông tin nào về k nếu người đó không biết toàn bộ khóa riêng tư K như một thông điệp.

Một biến thể chữ kí số khác là lược đồ chữ kí Schnorr. Nó sử dụng nhóm đường cong elliptic và xác định khoảng thời gian khóa k một cách xác định chứ không phải ngẫu nhiên. Chúng tôi gọi đó là ECSchnorr *. Việc áp dụng ROM cho H' mang lại hiệu quả với ECSchnorr * tương tự như với ECDSA và ECDSA+ .

Trong bài báo của Bellare và Rogaway đã tóm tắt một cách ngắn gọn giá trị của mô hình tiên tri ngẫu nhiên như sau: "Các mục tiêu có thể thực hiện được nhưng không thực tế trong thiết lập tiêu chuẩn trở thành thực tế trong ROM." Điều này vẫn còn như vậy, được thể hiện qua những biến thể chữ kí số khác như ECDSA *, ECDSA + và ECSchnorr *.

CHƯƠNG 6: CHƯƠNG TRÌNH THỬ NGHIỆM

Link source code : https://github.com/Vu0811/RSA-digital-signature

6.1. DEMO thuật toán mã hóa RSA

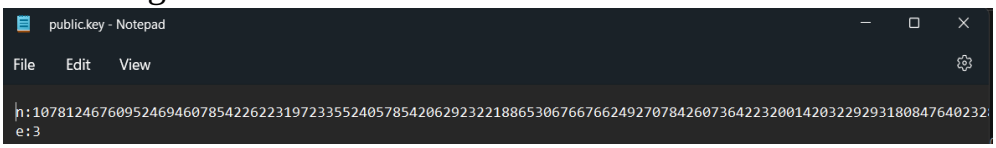
6.1.1. Tạo khóa

- Chạy câu lệnh

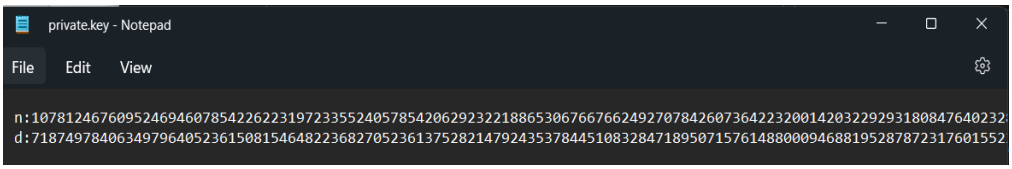
```
PS G:\Works\Code\RSA digital signature> python .\rsa\generate_keys.py
Successfully generated key: public.key
public.key path: G:\Works\Code\RSA digital signature\rsa\public.key

Successfully generated key: private.key
private.key path: G:\Works\Code\RSA digital signature\rsa\private.key
```

- Khóa công khai :



- Khóa bí mật:



6.1.2. Mã hóa bằng khóa công khai

- Chạy câu lệnh

```
PS G:\Works\Code\RSA digital signature> python .\rsa\encrypt.py
Enter message: abcdxyz
Key public.key read successfully!

n: 10781246760952469460785422622319723355240578542062923221886530676676624927078426073642232001420322929318084764023283821128478160269193332862583616833319445019516582703112194974478891640594695500540101028543779536913430101331569970595170300471054058628438228180992511623346838263913329015175166084628364146046 9
e: 3

**********
Encrypted message:
20596176539791285490870568001362072286419564020008
**********
```

6.1.3. Giải mã bằng khóa bí mật

- Chạy câu lệnh

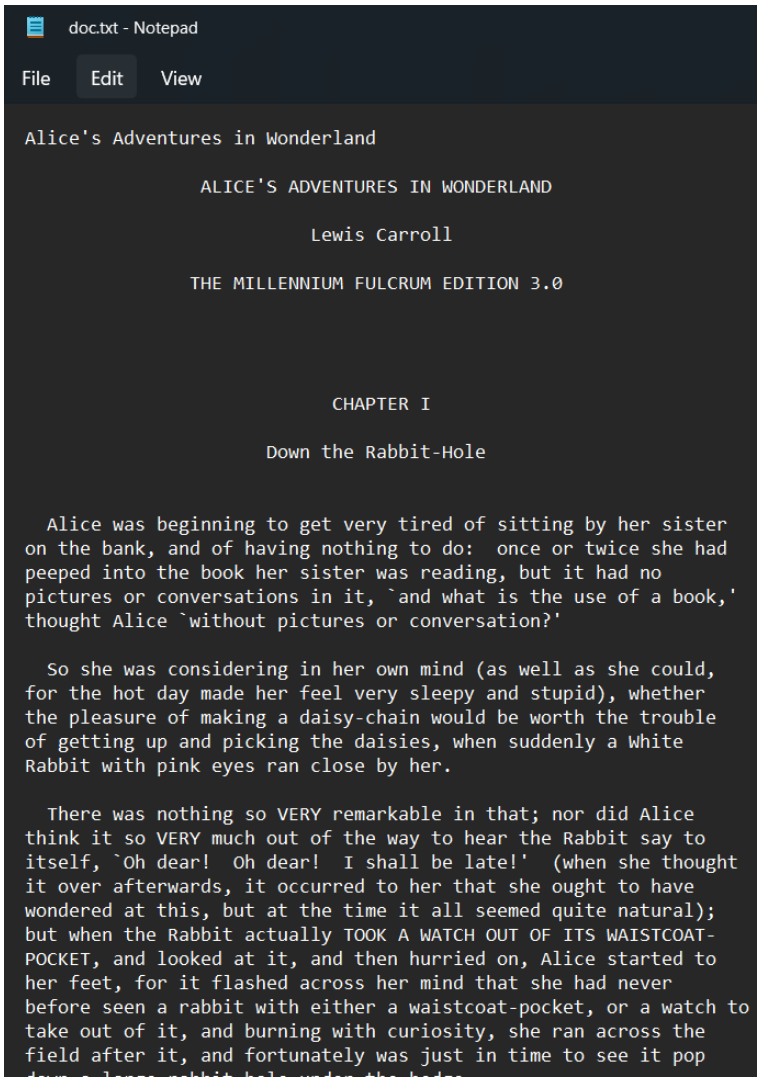
```
PS G:\Works\Code\RSA digital signature> python .\rsa\decrypt.py
Enter cipher text: 20596176539791285490870568001362072286419564020008
Key private.key read successfully!

n: 10781246760952469460785422622319723355240578542062923221886530676676624927078426073642232001420322929318084764023283821128478160269193332862583616833319445019516582703112194974478891640594695500540101028543779536913430101331569970595170300471054058628438228180992511623346838263913329015175166084628364146046 9
d: 71874978406349796405236150815464822368270523613752821479243537844510832847189507157614880009468819528787231760155225474189854401794622219083890778888796286136255136406239667116665614647725307289458741265623655240821617400113159934625060762052273835211963889281116806596921434185237751300610708753301903096307


**********
Decrypted message:
abcdxyz
**********
```

6.2. DEMO thuật toán băm SHA-1

6.2.1. Văn bản băm



6.2.2. Thực hiện hàm băm

- Chạy câu lệnh

```
PS G:\Works\Code\RSA digital signature> python sha1.py doc.txt
47ec1d4dc0c75a7a735dcaafdb48ad2fb92e46f7  doc.txt
```

6.3. DEMO thuật toán tạo chữ kí số

6.3.1. Quá trình ký

- Chạy câu lệnh

```
PS G:\Works\Code\RSA digital signature> python sign.py doc.txt
Key private.key read successfully!
Signed document
Alice's Adventures in Wonderland

                ALICE'S ADVENTURES IN WONDERLAND

                          Lewis Carroll

               THE MILLENNIUM FULCRUM EDITION 3.0




                             CHAPTER I

                       Down the Rabbit-Hole


  Alice was beginning to get very tired of sitting by her sister
on the bank, and of having nothing to do:  once or twice she had
peeped into the book her sister was reading, but it had no
pictures or conversations in it, `and what is the use of a book,'
thought Alice `without pictures or conversation?'
```

.

.

.

```
EPILOGUE
KING
The king's a beggar, now the play is done:
All is well ended, if this suit be won,
That you express content; which we will pay,
With strife to please you, day exceeding day:
Ours be your patience then, and yours our parts;
Your gentle hands lend us, and take our hearts.
Exeunt
*/*/*/*816995354400575729969668338362229169170399395605162063040108221242232408697722263101447981783736928007707853068437359953775416598212555236285220408874492523287220151693173991491327704063159315733399949355960632069423565482813380290996914132792301498173907276521669326572349194538479452366656290529100332627 31  signature_doc.txt
```

6.3.2. Quá trình kiểm tra

- Chạy câu lệnh

```
PS G:\Works\Code\RSA digital signature> python verify.py signature_doc.txt
Key public.key read successfully!
OK  signature_doc.txt
```

TÀI LIỆU THAM KHẢO

1. R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryp-tosystems".
2. Rabin, M. O. (1978), "Digitalized signatures", Foundations of Secure Computations, R. Lipton and R. DeMillo editors, Academic Press New York, 1978. [Rabin M. O. 1978]
3. H. C. Williams, "A modification of the RSA public-key procedure".
4. William Stallings, "Cryptography-network-security-5th-edition", Fifth Edition.
5. Matt Bishop, "Introduction to Computer Security".
6. Dang, Q. (2012), Secure Hash Standard (SHS), Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.FIPS.180-4 (Accessed July 20, 2022)
7. Viblo.asia: Tìm hiểu về chữ ký số và ứng dụng.
8. esign.misa.vn: Cơ chế bảo mật của phần mềm chữ ký số an toàn nhất như thế nào?
9. Bài giảng "Nhập môn An toàn thông tin" – PGS. TS. Nguyễn Linh Giang.
10. Bài giảng "Nhập môn An toàn thông tin" – TS. Trần Vĩnh Đức.
11. Bài giảng "Nhập môn An toàn thông tin" – PGS. TS. Nguyễn Khanh Văn.
12. Giáo trình "Mật mã học & An toàn thông tin" – TS. Thái Thanh Tùng.
13. Mã nguồn tham khảo:
    https://github.com/BharathKumarRavichandran/rsa.git
    https://github.com/pcaro90/Python-SHA1/blob/ae3396776773c0370607a15d92fc99542b025800/SHA1.py