BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

ĐỒ ÁN TỐT NGHIỆP

NGÀNH : CÔNG NGHỆ THÔNG TIN

Sinh viên : Phạm Nhật Hoàng

Giảng viên hướng dẫn: TS. Hồ Văn Canh

HẢI PHÒNG – 2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

ỨNG DỤNG HỆ MẬT MÃ RSA TRONG CHỮ KÝ ĐIỆN TỬ

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

NGÀNH: CÔNG NGHỆ THÔNG TIN

Sinh viên : Phạm Nhật Hoàng

Giảng viên hướng dẫn: TS. Hồ Văn Canh

HẢI PHÒNG – 2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên: Phạm Nhật Hoàng **Mã SV**: 1812402016

Lớp : CT2301C

Ngành : Công nghệ thông tin

Tên đề tài: Ứng dụng Hệ mật mã RSA trong chữ ký điện tử

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC CHỮ VIẾT TẮT

Chữ viết tắt	Tên tiếng anh	Nghĩa tiếng việt
RSA		Viết theo 3 chữ cái đầu của tên ba tác giả Rivest – Shamir - Adleman
DES	Data Encryption Standard	Tiêu chuẩn mã hóa dữ liệu
MD1	Message – Digest Algorithm 1	Giải thuật tiêu hóa thông tin 1
MD2	Message – Digest Algorithm 2	Giải thuật tiêu hóa thông tin 2
MD5	Message – Digest Algorithm 5	Giải thuật tiêu hóa thông tin 5
MIT	Massachusetts Institute of Technology	Viện công nghệ Massachusetts
SHA	Secure Hash Algorithm	Thuật giải băm an toàn
UCLN		Ước chung lớn nhất

LỜI CẢM ƠN

Để hoàn thành tốt được Đồ án tốt nghiệp, trước hết em xin gửi tới các Thầy Cô khoa Công nghệ thông tin trường Đại học Quản lý và Công nghệ Hải Phòng lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của Thầy Cô đã tạo điều kiện tốt nhất cho em để em hoàn thành đề tài đúng dự kiến, đến nay em đã có thể hoàn thành đồ án tốt nghiệp với đề tài *"Ứng dụng Hệ mật mã RSA trong chữ ký điện tử"*. Đặc biệt em xin gửi lời cảm ơn chân thành nhất tới thầy giáo – TS. Hồ Văn Canh đã quan tâm giúp đỡ, hướng dẫn em hoàn thành tốt đề tài này trong thời gian qua.

Em xin bày tỏ lòng biết ơn đến lãnh đạo Trường Đại học Quản lý và Công nghệ Hải Phòng, Khoa Công nghệ thông tin, các Phòng ban chức năng đã trực tiếp, gián tiếp giúp đỡ em khi em còn ngồi trên ghế nhà trường và trong suốt quá trình học tập, nghiên cứu đề tài.

Trong quá trình làm đồ án tốt nghiệp với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án tốt nghiệp này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, học được thêm nhiều kinh nghiệm và kiến thức để có thể phục vụ tốt hơn cho công việc thực tế sau này.

Em xin chân thành cảm ơn!

Hải Phòng, ngày… tháng… năm 2023

Sinh viên

(Ký và ghi rõ họ tên)

LỜI NÓI ĐẦU

Với sự phát triển ngày càng nhanh chóng của Internet và các ứng dụng giao dịch điện tử trên mạng, nhu cầu bảo vệ thông tin trong các hệ thống và ứng dụng điện tử ngày càng được quan tâm và có ý nghĩa hết sức quan trọng. Các kết quả của khoa học mật mã ngày càng được triển khai trong nhiều lĩnh vực khác nhau của đời sống – xã hội, trong đó phải kể đến rất nhiều những ứng dụng đa dạng trong lĩnh vực dân sự, thương mại...Các ứng dụng mã hóa thông tin cá nhân, trao đổi thông tin kinh doanh, thực hiện các giao dịch điện tử qua mạng... đã trở nên gần gũi và quen thuộc với mọi người.

Cùng với sự phát triển của khoa học máy tính, các nghiên cứu và ứng dụng của mật mã học ngày càng trở nên đa dạng hơn, mở ra nhiều hướng nghiên cứu chuyên sâu vào từng lĩnh vực ứng dụng đặc thù với những đặc trưng riêng. Ứng dụng của khoa học mật mã không chỉ đơn thuần là mã hóa và giải mã thông tin mà còn bao gồm nhiều vấn đề khác nhau cần được nghiên cứu và giải quyết, ví dụ như chứng thực nguồn gốc nội dung thông tin (kỹ thuật chữ ký điện tử), chứng nhận tính xác thực về người sở hữu mã khóa (chứng nhận khóa công cộng), các quy trình giúp trao đổi thông tin và thực hiện giao dịch điện tử an toàn trên mạng...

Và hệ mật RSA là một trong những hệ mật mã được sử dụng cho đến tận ngày nay nhằm đáp ứng những ví dụ trên. RSA được công bố lần đầu vào tháng 8 năm 1977 trên tạp chí khoa học Mỹ. Hệ mật sử dụng trong lĩnh vự đảm bảo tính riêng tư và cung cấp cơ chế xác thực của dữ liệu số. Ngày nay, RSA đã được phát triển ứng dụng rộng rãi trong thương mại điện tử. Nó được sử dụng trên Web servers và trên các Browsers nhằm đảm bảo an ninh đường truyền, được sử dụng trong việc tạo khóa và xác thực của mail, trong truy cập từ xa…, và đặc biệt nó là hạt nhận của hệ thống thanh toán điện tử. Tóm lại, RSA được ứng dụng rộng rãi trong các lĩnh vực nơi mà an ninh an toàn thông tin được đòi hỏi.

Qua quá trình học tập và nghiên cứu tại trường, dưới góc độ là sinh viên năm cuối và kết hợp sự định hướng hướng dẫn của thầy giáo Hồ Văn Canh em đã quyết định chọn đề tài: "*Ứng dụng hệ mật mã RSA trong chữ ký điện tử*" làm đồ án tốt nghiệp của mình.

CHƯƠNG 1: TỔNG QUAN VỀ HỆ MẬT MÃ KHOÁ CÔNG KHAI

1.1 Khái niệm về mật mã và mật mã khoá công khai

1.1.1 Mật mã

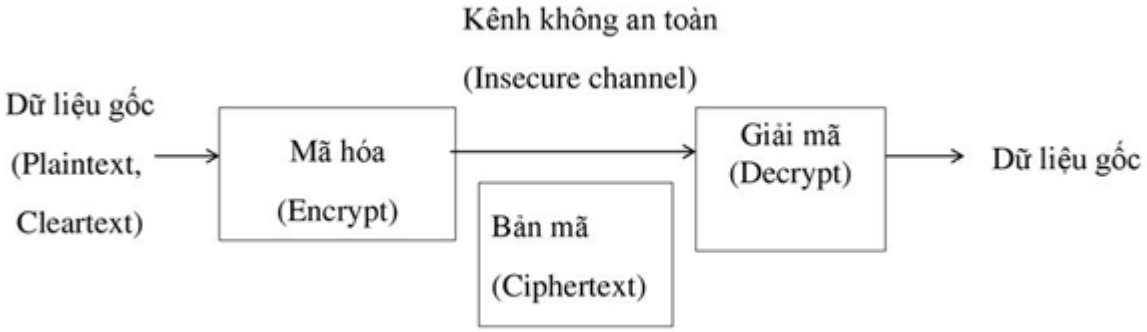
Mật mã là những quy tắc, quy ước riêng dùng để thay đổi hình thức biểu hiện thông tin nhằm bảo đảm bí mật, xác thực, toàn vẹn của nội dung thông tin.

Là một lĩnh vực liên quan đến các kỹ thuật ngôn ngữ và toán học để đảm bảo an toàn thông tin, cụ thể là trong thông tin liên lạc. Trong lịch sử, mật mã học gắn liền với qua trình mã hoá. Điều này có nghĩa là nó gắn với các cách thức để chuyển đổi thông tin từ dạng này sang dạng khác nhưng ở đây là dạng thông tin có thể nhận thức được thành dạng không nhận thức được, làm cho thông tin trở thành dạng không thể đọc được nếu như không có các kiến thức bí mật. Quá trình mã hoá được sử dụng chủ yếu để đảm bảo tính bí mật của các thông tin quan trọng, chẳng hạn trong công tác tình báo, quân sư hay ngoại giao cũng như các bí mật về kinh tế, thương mại.

Hình 1.1: Máy Enigma từ thế chiến thứ 2 được sử dụng để mã hoá, bảo vệ các thông tin tính toán học nhạy cảm.

Trong những năm gần đây, lĩnh vực hoạt động của mật mã hoá đã được mở rộng: mật mã hoá hiện đại cung cấp cơ chế cho nhiều hoạt động hơn là chỉ duy nhất việc giữ bí mật và có một loạt các ứng dụng như: chứng thực khoá công khai, chữ ký số, bầu cử điện tử hay tiền điện tử. Ngoài ra, những người không có nhu cầu thiết yếu đặc biệt về tính bí mật cũng sử dụng các công nghệ mật mã hoá, thông thường được thiết kế và tạo lập sẵn trong các cơ sở hạ tầng của công nghệ tính toán và liên lạc viễn thông.

Mật mã hóa cũng được coi là một nhánh của công nghệ, nhưng nó được coi là không bình thường vì nó liên quan đến các sự chống đối ngầm (xem công nghệ mật mã hóa và công nghệ an ninh). Mật mã hóa là công cụ được sử dụng trong an ninh máy tính và mạng.



Hình 1.2: Sơ đồ mã hoá và giải mã

1.1.2 Mật mã khoá công khai

Mật mã hóa khóa công khai là một dạng mật mã hóa cho phép người sử dụng trao đổi các thông tin mật mà không cần phải trao đổi các khóa chung bí mật trước đó. Điều này được thực hiện bằng cách sử dụng một cặp khóa có quan hệ toán học với nhau là khóa công khai và khóa cá nhân (hay khóa bí mật).

Thuật ngữ mật mã hóa khóa bất đối xứng thường được dùng đồng nghĩa với mật mã hóa khóa công khai mặc dù hai khái niệm không hoàn toàn tương

đương. Có những thuật toán mật mã khóa bất đối xứng không có tính chất khóa công khai và bí mật như đề cập ở trên mà cả hai khóa (cho mã hóa và giải mã) đều cần phải giữ bí mật.

Trong mật mã hóa khóa công khai, khóa cá nhân phải được giữ bí mật trong khi khóa công khai được phổ biến công khai. Trong 2 khóa, một dùng để mã hóa và khóa còn lại dùng để giải mã. Điều quan trọng đối với hệ thống là không thể tìm ra khóa bí mật nếu chỉ biết khóa công khai.

Hệ thống mật mã hóa khóa công khai có thể sử dụng với các mục đích:

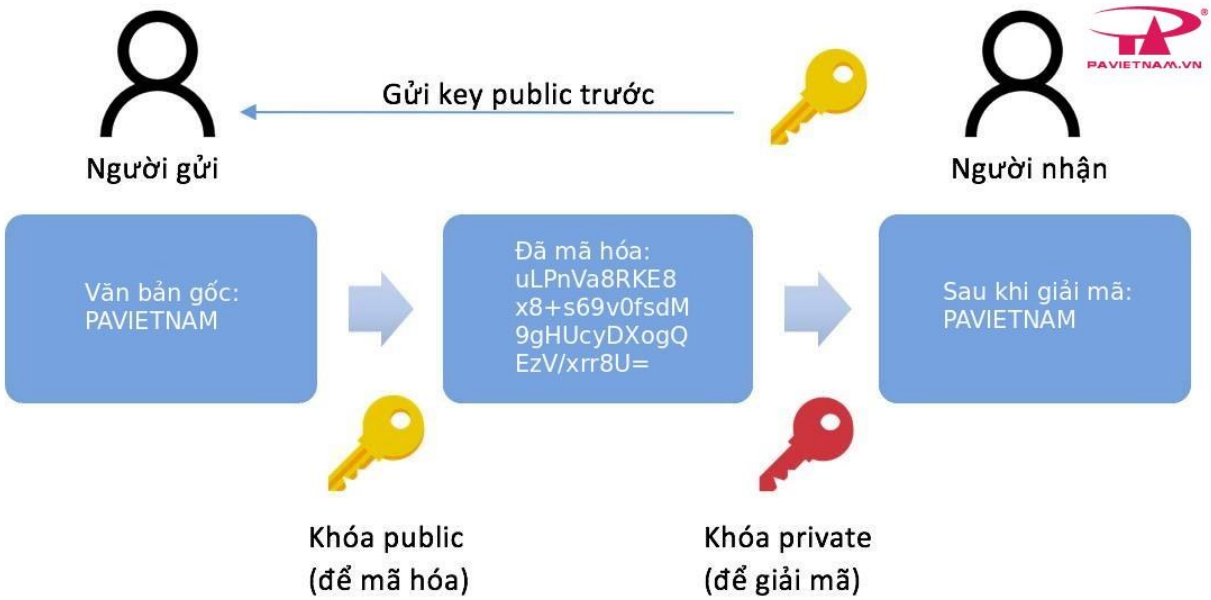
- **Mã hóa**: giữ bí mật thông tin và chỉ có người có khóa bí mật mới giải mã được.
- **Tạo chữ ký số**: cho phép kiểm tra một văn bản có phải đã được tạo với một khóa bí mật nào đó hay không.
- **Thỏa thuận khóa**: cho phép thiết lập khóa dùng để trao đổi thông tin mật giữa 2 bên.

Thông thường, các kỹ thuật mật mã hóa khóa công khai đòi hỏi khối lượng tính toán nhiều hơn các kỹ thuật mã hóa khóa đối xứng nhưng những lợi điểm mà chúng mang lại khiến cho chúng được áp dụng trong nhiều ứng dụng.

Trong hầu hết lịch sử mật mã học, khóa dùng trong các quá trình mã hóa và giải mã phải được giữ bí mật và cần được trao đổi bằng một phương pháp an toàn khác (không dùng mật mã) như gặp nhau trực tiếp hay thông qua một người đưa thư tin cậy. Vì vậy quá trình phân phối khóa trong thực tế gặp rất nhiều khó khăn, đặc biệt là khi số lượng người sử dụng rất lớn. Mật mã hóa khóa công khai đã giải quyết được vấn đề này vì nó cho phép người dùng gửi thông tin mật trên đường truyền không an toàn mà không cần thỏa thuận khóa từ trước.

Thuật toán đầu tiên cũng được Rivest, Shamir và Adleman tìm ra vào năm 1977 tại MIT. Công trình này được công bố vào năm 1978 và thuật toán được đặt tên là RSA. RSA sử dụng phép toán tính hàm mũ môđun (môđun được

tính bằng tích số của 2 số nguyên tố lớn) để mã hóa và giải mã cũng như tạo chữ ký số. An toàn của thuật toán được đảm bảo với điều kiện là không tồn tại kỹ thuật hiệu quả để phân tích một số rất lớn thành thừa số nguyên tố.



Hình 1.3 Sơ đồ mã hoá công khai

1.2 Một số hệ mật mã đối xứng và khoá công khai (bất đối xứng)

*Hệ mật mã đối xứng

Trong hệ thống mã hóa quy ước, quá trình mã hóa và giải mã một thông điệp sử dụng cùng một mã khóa gọi là khóa bí mật (secret key) hay khóa đối xứng (symmetric key). Do đó, vấn đề bảo mật thông tin đã mã hóa hoàn toàn phụ thuộc vào việc giữ bí mật nội dung của mã khóa đã được sử dụng.

Với tốc độ và khả năng xử lý ngày càng được nâng cao của các bộ vi xử lý hiện nay, phương pháp mã hóa chuẩn (Data Encryption Standard – DES) đã trở nên không an toàn trong bảo mật thông tin. Do đó, Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (National Institute of Standards and Technology – NIST) đã quyết định chọn một chuẩn

mã hóa mới với độ an toàn cao nhằm phục vụ nhu cầu bảo mật thông tin liên lạc của chính phủ Hoa Kỳ cũng như trong các ứng dụng dân sự. Thuật toán Rijndael do Vincent Rijmen và Joan Daeman đã được chính thức chọn trở thành chuẩn mã hóa nâng cao (Advanced Encryption Standard –AES) từ 02 tháng 10 năm 2000.

Ví dụ thông điệp nguồn được mã hóa với mã khóa k được thống nhất trước giữa người gửi A và người nhận B. Người A sẽ sử dụng mã khóa k để mã hóa thông điệp x thành thông điệp y và gửi y cho người B người B sẽ sử dụng mã khóa k để giải mã thông điệp y này. Vấn đề an toàn bảo mật thông tin được mã hóa phụ thuộc vào việc giữ bí mật nội dung mã khóa k.Nếu người C biết được mã khóa k thì C có thể "mở khóa" thông điệp đã được mã hóa mà người A gửi cho người B.

1.2.1 Phương pháp mã hoá dịch chuyển

Phương pháp mã hóa dịch chuyển là một trong những phương pháp lâu đời nhất được sử dụng để mã hóa. Thông điệp được mã hóa bằng cách dịch chuyển xoay vòng từng ký tự đi k vị trí trong bảng chữ cái.

Trong trường hợp đặc biệt k=3, phương pháp mã hóa dịch chuyển được gọi là phương pháp mã hóa Caesar.

Ví dụ: Dùng khóa k=9 để mã hóa dòng thư:
"hentoithubay"
Dòng thư đó tương ứng với dòng số

h	e	n	t	o	i	t	h	u	b	a	y
7	4	13	19	14	8	19	7	20	1	0	24

Qua phép mã hóa e_9 sẽ được:

16	13	22	2	23	17	2	16	3	10	9	7
q	n	w	c	x	r	c	q	d	k	j	h

Như vậy bản mã sẽ là: "**qnwcxrcqdkjh**"
Dùng d_9 giải mã ta sẽ được bản rõ ban đầu

Cách đây *2000* năm mã dịch chuyển đã được Julius Ceasar sử dụng, với khóa k=3 mã dịch chuyển được gọi là mã Ceasar.

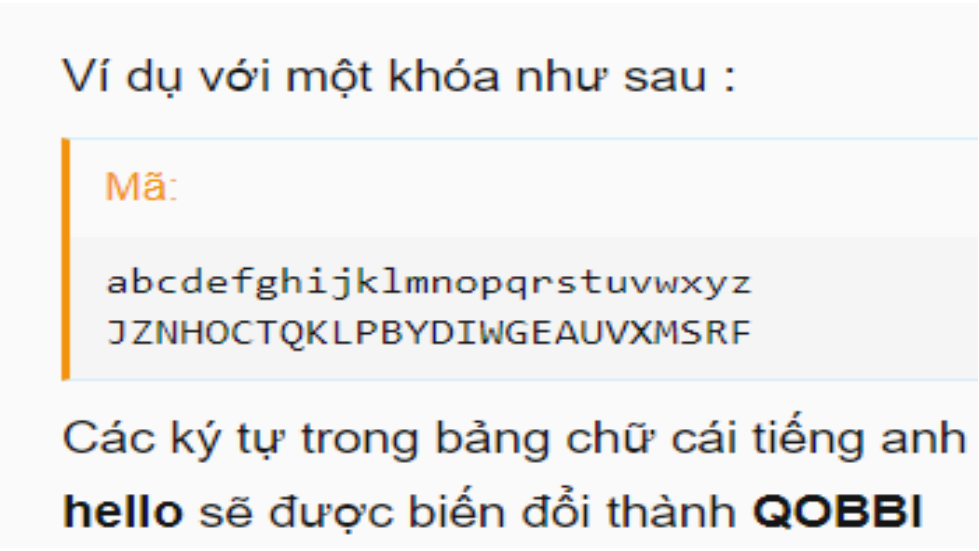
Hình 1.4 Ví dụ về phương pháp mã hoá dịch chuyển.

Mã hóa dịch chuyển là một phương pháp mã hóa đơn giản, thao tác xử lý mã hóa và giải mã được thực hiện nhanh chóng. Tuy nhiên, trên thực tế, phương pháp này có thể dễ dàng bị phá vỡ bằng cách thử mọi khả năng khóa $k \in K$. Điều này hoàn toàn có thể thực hiện được do không gian khóa K chỉ có n phần tử để chọn lựa.

Ví dụ: Để mã hóa một thông điệp được biểu diễn bằng các chữ cái từ A đến Z (26 chữ cái), ta sử dụng P = C = K = Z26. Khi đó, thông điệp được mã hóa sẽ không an toàn và có thể dễ dàng bị giải mã bằng cách thử lần lượt 26 giá trị khóa k € K. Tính trung bình, thông điệp đã được mã hóa có thể bị giải mã sau khoảng *n/2* lần thử khóa $k \in K$.

1.2.2 Phương pháp mã hoá thay thế:

Phương pháp mã hóa thay thế (Substitution Cipher) là một trong những phương pháp mã hóa nổi tiếng và đã được sử dụng từ hàng trăm năm nay. Phương pháp này thực hiện việc mã hóa thông điệp bằng cách hoán vị các phần tử trong bảng chữ cái hay tổng quát hơn là hoán vị các phần tử trong tập nguồn P.



Hình 1.5 Ví dụ về phương pháp mã hoá thay thế

Đây là một phương pháp đơn giản, thao tác mã hóa và giải mã được thực hiện nhanh chóng. Phương pháp này khắc phục điểm hạn chế của phương pháp mã hóa bằng dịch chuyển là có không gian khóa K nhỏ nên dễ dàng bị giải mã bằng cách thử nghiệm lần lượt n giá trị khóa $k \in K$. Trong phương pháp mã hóa thay thế có không gian khóa K rất lớn với $n!$ phần tử nên không thể bị giải mã bằng cách "vét cạn" mọi trường hợp khóa k. Tuy nhiên, trên thực tế thông điệp được mã hóa bằng phương pháp này vẫn có thể bị giải mã nếu như có thể thiết lập được bảng tần số xuất hiện của các ký tự trong thông điệp hay nắm được một số từ, ngữ trong thông điệp nguồn ban đầu.

1.2.3 Phương pháp Affine

Nếu như phương pháp mã hóa bằng dịch chuyển là một trường hợp đặc biệt của phương pháp mã hóa bằng thay thế, trong đó chỉ sử dụng n giá trị khóa k trong số $n!$ phần tử, thì phương pháp Affine lại là một trường hợp đặc biệt khác của mã hóa bằng thay thế.

Mã hóa Affine là một bộ năm (P, C, K, E, D) thỏa mãn:

- $P = C = Z_n$
- $K = \{(a, b) \in Z_n \times Z_n\} : gcd(a,n) = 1\}$
- $E = \{e_k, k \in K\}$ trong đó: $e_k(x) = (ax + b) \bmod n$ với $x \in Z_n$
- $D = \{d_k, k \in K\}$ trong đó: $d_k(y) = (a^{-1} (y - b)) \bmod n$ với $y \in Z_n$

Ví dụ:

- p = "abcde"
- k = (a, b) = (5, 3); n =26
- Tính c = ?

Ta có c :

P	a	b	c	d	e
X	0	1	2	3	4
(5x + 3)mod 26	3	8	13	18	23
c	d	i	n	s	x

⇨ c = “dinsx”

Tương tự với bài toán ngược ta có:

- "dinsx"
- k = (a, b) = (5, 3); n= 26; tính được a mũ -1 = 21 từ công thức :
- $1 = aa^{-1} \bmod m$
- tính p = ?

Ta có p:

c	d	i	n	s	x
Y	3	8	13	18	23
(21(y-3))mod 26	0	1	2	3	4
p	a	b	c	d	e

⇨ p = “abcde”

1.2.4 Phương pháp mã hoá Vigenere

Trong phương pháp mã hóa bằng thay thế cũng như các trường hợp đặc biệt của phương pháp này (mã hóa bằng dịch chuyển, mã hóa Affine,…), ứng với một khóa *k* được chọn, mỗi phần tử $x \in P$ được ánh xạ vào duy nhất một phần tử $y \in C$. Nói cách khác, ứng với mỗi khóa $k \in K$, một song ánh được thiết lập từ P vào C.

Khác với hướng tiếp cận này, phương pháp Vigenere sử dụng một từ khóa có độ dài *m*. Có thể xem như phương pháp mã hóa Vigenere Cipher bao gồm *m* phép mã hóa bằng dịch chuyển được áp dụng luân phiên nhau theo chu kỳ.

Không gian khóa *K* của phương pháp Vigenere Cipher có số phần tử là n^m, lớn hơn hẳn phương pháp số lượng phần tử của không gian khóa *K* trong phương pháp mã hóa bằng dịch chuyển. Do đó, việc tìm ra mã khóa *k* để giải mã thông điệp đã được mã hóa sẽ khó khăn hơn đối với phương pháp mã hóa bằng dịch chuyển.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Hình 1.6 Bảng mật mã của Vigenere

Ví dụ ta có một chuỗi cần mã hóa như sau: “ATTACKATDAWN”

Và key dùng để mã hóa là “LEMON”. Trước hết ta sẽ nhân chuỗi LEMON này lên để nó có cùng độ dài với chuỗi cần mã hóa: LEMONLEMONLE

Khi này ta sẽ sử dụng bảng mã hóa như sau: bắt đầu từ trái qua phải, lấy ký tự của key làm *dòng*, ký tự của chuỗi cần mã hóa là *cột* và dóng vào trong bảng mã ta được một ký tự, ký tự đó chính là ký tự đã được mã hóa.

Dòng L, cột A = L; Dòng E, cột T = X; …

Áp dụng với key LEMONLEMONLE và chuỗi ATTACKATDAWN bên trên ta được chuỗi: LXFOPVEFRNHR

1.2.5 Phương pháp mã hoá Hill

Sơ đồ mật mã này được đề xuất bởi Lester S. Hill năm 1929. Cũng giống như sơ đồ mã Vigenère, các hệ mã này được thực hiện trên từng bộ *m* ký tự liên tiếp, điều khác là mỗi ký tự của bản mã được xác định bởi một tổ hợp tuyến tính (trên vành $\boldsymbol{Z_{26}}$**)** của *m* ký tự trong bản rõ. Như vậy, khoá sẽ được cho bởi một ma trận cấp *m*, tức là một phần tử của $K \in Z^{m\,\mathrm{xm}}$. Để phép biến đổi tuyến tính xác định bởi ma trận *K* có phép nghịch đảo, bản thân ma trận *K* cũng phải có ma trận nghịch đảo K^{-1} theo mod26; mà điều kiện cần và đủ để *K* có nghịch đảo là định thức của nó, ký hiệu detK, nguyên tố với 26. Vậy, sơ đồ mật mã Hill được định nghĩa là sơ đồ

$$S = (P, C, K, E, D),$$

trong đó: $P = C = Z_{26}^{m}$, $K = \left\{K \in Z_{26}^{m\times m} : \gcd(\det K, 26) = 1\right\}$,

các ánh xạ *E* và *D* được cho bởi:

$$E_k(x_1,..., xm) = (x_1,..., xm).K \mod 26,$$

$$D_k(y_1,..., ym) = (y_1,..., ym).K^{-1} \mod 26$$

với mọi $x = (x_1,..., xm) \in P$, $y = (y_1,..., ym) \in C$, $k \in K$

1.2.6 Phương pháp mã hoá hoán vị

Những phương pháp mã hóa nêu trên đều dựa trên ý tưởng chung: thay thế mỗi ký tự trong thông điệp nguồn bằng một ký tự khác để tạo thành thông điệp đã được mã hóa. Ý tưởng chính của phương pháp mã hóa hoán vị là vẫn giữ nguyên các ký tự trong thông điệp nguồn mà chỉ thay đổi vị trí các ký tự; nói cách khác thông điệp nguồn được mã hóa bằng cách sắp xếp lại các ký tự trong đó.

Phương pháp mã hóa bằng hoán vị chính là một trường hợp đặc biệt của phương pháp Hill. Với mỗi hoán vị π của tập hợp {1, 2, ..., m} , ta xác định ma trận $k_\pi=(k_{i,j})$ theo công thức sau:

- $k_{i,j} = 1$ *nếu* $i = \pi(j)$
- $k_{i,j} = 1$ *nếu 0 trong trường hợp ngược lại*

Ma trận k_π là ma trận mà mỗi dòng và mỗi cột có đúng một phần tử mang giá trị 1, các phần tử còn lại trong ma trận đều bằng 0. Ma trận này có thể thu được bằng cách hoán vị các hàng hay các cột của ma trận đơn vị I_m nên k_π là ma trận khả nghịch. Rõ ràng, mã hóa bằng phương pháp Hill với ma trận k_π hoàn toàn tương đương với mã hóa bằng phương pháp hoán vị với hoán vị π.

Ví dụ:

Ghi các ký tự trong bản rõ theo từng hàng, sau đó kết xuất bản mã dựa trên cột. Sau đó đọc bảng mã theo từng hàng.

Cho bản rõ “meet me after the toga party” với hành rào sắt độ sâu là 2 (Tách bản rõ thành 2 hàng)

bản rõ: “meet me at the toga party”

được viết thành:

m e m a t r h t g p r y

e t e f e t e o a a t

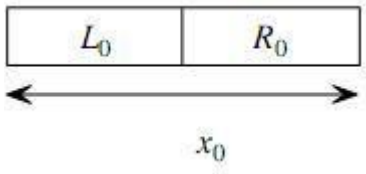
Cho bản mã MEMATRHTGPRYETEFETEOAAT.

1.2.7 Phương pháp DES (Data Encyption Standard)

Thuật toán mã khối DES (Data Encryption Standard) là một thuật toán mã khối với kích thước khối 64 bít và kích thước khóa 56 bít, được công bố chính thức bởi Tổ chức Tiêu chuẩn xử lý thông tin liên bang

Hoa Kỳ (FIPS) vào tháng 11/1976 và được xuất bản trong tài liệu FIPS PUB 46 (01/1977). Thuật toán DES đã trải quả nhiều lần cập nhật: năm 1988 (FIPS-46-1), 1993 (FIPS-46-2), 1998 (FIPS-46-3). Tiền thân của thuật toán DES là thuật toán Lucifer, một thuật toán do IBM phát triển. Cuối năm 1976, DES được chọn làm chuẩn mã hóa dữ liệu của Hoa Kỳ, sau đó được sử dụng rộng rãi trên toàn thế giới trong lĩnh vực an toàn, bảo mật thông tin trên môi trường số.

1. Tạo dãy 64 bit x_0 bằng cách hoán vị x theo hoán vị IP (Initial Permutation). Biểu diễn $x_0 = IP(x) = L_0 R_0$, L_0 gồm 32 bit bên trái của x_0, R_0 gồm 32 bit bên phải của x_0.



Hình 1.7 : Biểu diễn 64 bit x thành 2 phần L và R

2. Thực hiện 16 vòng lặp từ 64 bit thu được và 56 bit của khoá k (chỉ sử dụng 48 bit của khoá k trong mỗi vòng lặp). 64 bit kết quả thu được qua mỗi vòng lặp sẽ là đầu vào cho vòng lặp sau. Các cặp từ 32 bit L_i, R_i (với $1 \leq i \leq 16$) được xác định theo quy tắc sau:

$L_i = R_{i-1}$

$R_i = L_{i-1}$ *XOR* $f(R_{i-1}, K_i)$

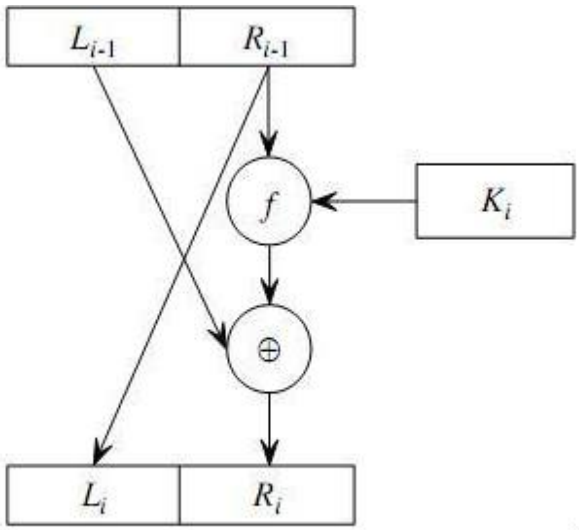
XOR trên hai dãy bit, $K_1, K_2, ..., K_{16}$ là các dãy 48 bit phát sinh từ khóa K cho trước (Trên thực tế, mỗi khóa K_i được phát sinh bằng cách hoán vị các bit trong khóa K cho trước).

3. Áp dụng hoán vị ngược IP^{-1} đối với dãy bit $R_{16}L_{16}$, thu được y gồm 64 bit. Như vậy, $y = IP^{-1}(R_{16} L_{16})$.

Hàm f được sử dụng ở bước 2 là hàm có gồm hai tham số: Tham số thứ nhất A là một dãy 32 bit, tham số thứ hai J là một dãy 48 bit. Kết

quả của hàm f là một dãy 32 bit. Các bước xử lý của hàm $f(A, J)$ như sau:

Tham số thứ nhất A (32 bit) được mở rộng thành dãy 48 bit bằng hàm mở rộng E. Kết quả của hàm $E(A)$ là một dãy 48 bit được phát sinh từ A bằng cách hoán vị theo một thứ tự nhất định 32 bit của A, trong đó có 16 bit của A được lặp lại hai lần trong $E(A)$.



Hình 1.8 : Quy trình phát sinh dãy L_i, R_i từ dãy L_{i-1}, R_{i-1} và khóa K_i

Thực hiện phép toán XOR cho hai dãy 48 bit $E(A)$ và J, ta thu được một dãy 48 bit B. Biểu diễn B thành từng nhóm 6 bit như sau: $B = B_1B_2B_3B_4B_5B_6B_7B_8$.

Sử dụng tám ma trận $S_1, S_2,..., S_8$, mỗi ma trận S_i có kích thước 4x16 và mỗi dòng của ma trận nhận đủ 16 giá trị từ 0 đến 15. Xét dãy gồm 6 bit $B_i = b_1b_2b_3b_4b_5b_6$, $S_j(B_j)$ được xác định bằng giá trị của phần tử tại dòng r cột c của S_j, trong đó, chỉ số dòng r có biểu diễn nhị phân là b_1b_6, chỉ số cột c có biểu diễn nhị phân là $b_1b_2b_4b_5$. Bằng cách này, ta xác định được dãy 4 bit $C_j = S_j(B_j)$, $1 \leq j \leq 8$.

Tập hợp các dãy 4 bit C_j lại, ta có được dãy 32 bit $C=C_1C_2C_3C_4C_5C_6C_7C_8$. Dãy 32 bit thu được bằng cách hoán vị C theo

một quy luật *P* nhất định chính là kết quả của hàm *F(A,J)*.

Quá trình giải mã chính là thực hiện theo thứ tự đảo ngược các thao tác của quá trình mã hóa.

***Hệ mật mã khoá công khai (bất đối xứng).**

Mã hóa bất đối xứng là một dạng mật mã hóa cho phép người sử dụng trao đổi các thông tin mật mà không cần phải trao đổi các khóa chung bí mật trước đó. Điều này được thực hiện bằng cách sử dụng một cặp khóa có quan hệ toán học với nhau là khóa công khai và khóa cá nhân (hay khóa bí mật). Trong mã bất đối xứng, khóa bí mật phải được giữ kín trong khi khóa công khai được phổ biến công khai. Trong 2 khóa, một dùng để mã hóa và khóa còn lại dùng để giải mã. Điều quan trọng đối với hệ thống là khó có thể tìm ra khóa bí mật nếu chỉ biết khóa công khai. Hệ thống mã bất đối xứng có thể sử dụng với các mục đích như:

- Mã hóa: giữ bí mật thông tin và chỉ có người có khóa bí mật mới giải mã được.
- Tạo chữ ký số: Việc kiểm tra một chữ ký nào đó dễ dàng được thực hiện nhờ khóa công khai cho trước.
- Thỏa thuận khóa: cho phép thiết lập khóa dùng để trao đổi thông tin mật giữa 2 bên.
- Phân phối khóa: Phân phối khóa được định nghĩa là một cơ chế theo đó một bên chọn khóa bí mật và sau đó truyền nó tới một hoặc nhiều bên khác nhau.
- Thỏa thuận khóa để chỉ một giao thức theo đó hai (hoặc nhiều hơn) bên cùng thiết lập khóa bí mật bằng cách liên lạc trên kênh công cộng

Các kỹ thuật mã bất đối xứng đòi hỏi khối lượng tính toán nhiều hơn các kỹ thuật mã hóa khóa đối xứng nhưng những lợi điểm mà chúng mang lại khiến cho chúng được áp dụng trong nhiều ứng dụng.

1.2.8 Hệ mật mã RSA

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.

Ta có thể mô phỏng trực quan một hệ mật mã khoá công khai như sau: Bình muốn gửi cho An một thông tin mật mà Bình muốn duy nhất An có thể đọc được. Để làm được điều này, An gửi cho Bình một chiếc hộp có khóa đã mở sẵn và giữ lại chìa khóa. Bình 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 Bình cũng không thể mở lại được-không đọc lại hay sửa thông tin trong thư được nữa). Sau đó Bình gửi chiếc hộp lại cho An. An 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.

Tạo khóa

Giả sử An và Bình cần trao đổi thông tin bí mật thông qua một kênh không an toàn (ví dụ như Internet). Với thuật toán RSA, An đầu tiên cần tạo ra cho mình cặp khóa gồm khóa công khai và khóa bí mật theo 6 bước sau:

1. Chọn 2 số nguyên tố lớn khác nhau p và q với $p \neq q$.

2. Tính tích của nó $n = p*q$

3. Tính giá trị hàm Phi Euler của n: $\varphi(n) = (p-1)*(q-1)$

4. Chọn một số tự nhiên e sao cho: $(1 < e < \varphi(n)$ và $UCLN(\varphi(n),e) = 1$

5. Tính d: $d \equiv e^{-1} \pmod{(\varphi(n))}$.

6. Khoá công khai bao gồm: n và e. Khoá bí mật: d và n.

Quá trình mã hoá:

Giả sử Bình muốn gửi đoạn thông tin m<n cho An, thì Bình tính bản mã như sau:

Công thức mã hoá với khoá công khai là cặp số (e, n):

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

Cuối cùng Bình gửi c cho An.

Quá trình giải mã:

An nhận c từ Bình và khoá bí mật d. An có thể tìm được m từ c theo công thức sau:

Công thức giải mã với khoá bí mật d:

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

Ví dụ:

1. Chọn các số nguyên tố: p = 7 và q = 11.

2. Tính n = p * q = 7 * 11 = 77

3. Tính m = (p - 1)(q - 1) = 6 * 10 = 60

4. Chọn e: UCLN(e, 60) = 1; lấy e = 7

5. Xác định d: de ≡ 1 mod 60 và d < 77

Giá trị cần tìm là d = 43, vì 43 * 7 = 301 = 5 * 60 + 1

=> Khoá công khai KU = {n, e} = {77, 7}

=> Khoá riêng bí mật KR = {43}

1.2.9 Hệ mật mã El-gamal

Giả sử Alice và Bob muốn trao đổi thông tin bằng mật mã El-gamal thì Alice thực hiện quá trình hình thành khoá như sau:

Chọn số nguyên tố đủ lớn p và 2 số nguyên tố nhỏ hơn p là alpha và a (khoá bí mật của người nhận) sau đó tính khoá công khai:

$beta = alpha^a \bmod p$

Để mã hoá thông điệp M (một số nguyên tố trên Z_p) thành bản mã C người gửi chọn một số ngẫu nhiên k nhỏ hơn p và tính cặp bản mã:

$C_1 = alpha^k \bmod p$

$C_2 = (M*beta^k) \bmod p$

Bản mã $E(C_1,C_2)$ được gửi đi với:

$C_1 = alpha^k \bmod p$

$C_2 = (M*beta^k) \bmod p$

Sau đó k sẽ bị huỷ đi.

Giải mã:

Ta dùng khoá bí mật a và tính theo công thức:

$M = (C_2*(C_1^a)^{-1}) \bmod p$ với $((C_1^a)^{-1}) \bmod p = (C_1^{(p-1-a)}) \bmod p$

Kết luận: K = (p, alpha, a, beta) với:

Thành phần khoá công khai:

Ku = (alpha, beta, p)

Thành phần khoá bí mật:

Kr = (a,p)

Ví dụ:

Cho: p = 2579; alpha = 2; a = 765; chọn k ngẫu nghiên là 853.

Bản rõ: M = 1299

Trước hết tính: $beta = alpha^a \bmod p = 2^{765} \bmod 2579 = 949$

Để mã hoá thông điệp M = 1299 ta tính theo k = 853

$C_1 = alpha^k \bmod p = 2^{853} \bmod 2579 = 435$

$C_2 = (M*beta) \bmod p = (1299*949^{853}) \bmod 2579 = 2396$

Vậy bản mã được gửi là: C = (435, 2396)

Giải mã: với khoá bí mật a = 765:

$((C1^a)^{-1}) \bmod p = (C1^{(p-1-a)}) \bmod p = (435^{(2579-1-765)}) \bmod 2579$

$= (435^{1813}) \bmod 2579 = 1980$

$M = (C2*(C1^a)^{-1}) \bmod p = (2396*1980) \bmod 2579 = 1299$

Xây dựng được hệ mã Elgamal bộ khoá:

K = (p, alpha, a, beta) = (2579, 2, 765, 949) với:

Thành phần khoá công khai:

Ku = (alpha, beta, p) = (2, 949, 2579)

Thành phần khoá bí mật:

Kr = (a,p) = (756, 2579)

Mã hoá M = 1299 với E(C1,C2) = (435,2396)

1.2.10 Phương pháp trao đổi khoá Diffie-Hellman

Trao đổi khoá Diffie-Hellman là thiết lập một khoá chia sẻ bí mật được sử dụng cho thông tin liên lac bí mật bằng cách trao đổi dữ liệu thông qua mạng công cộng.

Trao đổi khoá Diffie-Hellman là phương pháp được sử dụng rộng rãi đầu tiên để phát triển và trao đổi khoá một cách an toàn trên một kênh không an toàn về độ bảo mật.

Thuật toán mã hoá:

Điểm chủ chốt của ý tưởng này là Alice và Bob trao đổi màu sơn bí mật thông qua hỗn hợp sơn.

- Đầu tiên Alice và Bob trộn màu đã biết chung (màu vàng) với màu bí mật riêng của mỗi người.
- Sau đó, mỗi người chuyển hỗn hợp của mình tới người kia thông qua một kênh vận chuyển công cộng.
- Khi nhận được hỗn hợp của người kia, mỗi người sẽ trộn thêm với màu bí mật của riêng mình và nhận được hỗn hợp cuối cùng.

Hỗn hợp sơn cuối cùng là hoàn toàn giống nhau cho cả hai người và chỉ có riêng hai người biết. Mấu chốt ở đây là đối với một người ngoài sẽ rất khó (về mặt tính toán) cho họ để tìm ra được bí mật chung của hai người (nghĩa là hỗn hợp cuối cùng). Alice và Bob sẽ sử dụng bí mật chung này để mã hóa và giải mã dữ liệu truyền trên kênh công cộng. Lưu ý, màu sơn đầu tiên (màu vàng) có thể tùy ý lựa chọn, nhưng được thỏa thuận trước giữa Alice và Bob. Màu sơn này cũng có thể được giả sử là không bí mật đối với người thứ ba mà không làm lộ bí mật chung cuối cùng của Alice và Bob.

Giao thức được diễn giải dưới dạng toán học như sau:

Giao thức sử dụng nhóm nhân số nguyên modulo p, trong đó p số nguyên tố, và g là căn nguyên thủy mod p. Giải thuật được thực hiện như sau:

Bước 1: Alice và Bob thoả thuận sử dụng chung một số nguyên tố p và căn nguyên thuỷ g.

Bước 2: Alice chọn một số nguyên tố bí mật a, và gửi cho Bob giá trị $A = g^a \bmod p$.

Bước 3: Tương tự Bob chọn một số nguyên tố bí mật b và gửi cho Alice giá trị $B = g^b \bmod p$.

Bước 4: Alice tính được khoá $s = B^a \bmod .p$

Bước 5: Tương tự Bob tính $s = A^b \bmod p$.

Bước 6: Như vậy Alice và Bob cùng khoá chung là s.

Tạo Khoá:

Bước 1: Alice và Bob thoả thuận sử dụng chung một nhóm cyclic hữu hạn G và một phần tử sinh g của G. Phần tử sinh g công khai với tất cả mọi người, kể cả kẻ tấn công. Dưới đây chúng ta giả sử nhóm G là nhóm nhân.

Bước 2: Alice chọn một số tự nhiên ngẫu nhiên a và gửi g^a mod p cho Bob.

Bước 3: Bob chọn một số tự nhiên ngẫu nhiên b và gửi g^b mod p cho Alice.

Bước 4: Alice tính $(g^b)^a$ mod p.

Bước 5: Bob tính $(g^a)^b$ mod p.

Vì giá trị $(g^b)^a$ và $(g^a)^b$ là bằng nhau (do nhóm G có tính kết hợp), cả Alice và Bob đều tính được giá trị g^{ab} và có thể sử dụng nó cho khoá bí mật chung.

Mã hoá:

Thông điệp m trước khi gửi đi bởi Alice hoặc Bob sẽ được mã hoá thành mg^{ab}

Giải mã:

Để giải mã thông điệp m, gửi dưới dạng mg^{ab}, Bob (hoặc Alice) phải tính được giá trị $(g^{ab})^{-1}$. Giá trị $(g^{ab})^{-1}$ được tính như sau: vì Bob biết |G|, b và g^a, mặt khác theo định lý Lagrange trong lý thuyết nhóm ta có $x^{|G|} = 1$ với mọi x thuộc G, nên Bob tính được $(g^a)^{|G|-b} = g^{a(|G|-b)} = g^{a|G|-ab} = g^{a|G|}g^{-ab} = (g^{|G|})^a g^{-ab} = 1^a g^{-ab} = g^{-ab} = (g^{ab})^{-1}$

Việc giải mã bây giờ trở nên dễ dàng: Bob sử dụng $(g^{ab})^{-1}$ đã tính và phục hồi thông điệp nguyên thuỷ bằng cách tính: $mg^{ab}(g^{ab})^{-1} – m(1) – m$.

1.3 Một số tính chất của mật mã khóa công khai.

Từ một số mã khoá công khai kể trên ta có thể thấy được một vài tính chất của nó như sau:

1. Khoá mã hoá và khoá giải mã là khác nhau.

2. Mỗi bên có khoá bí mật của riêng mình và khoá công khai tương ứng (Ks, Kp).
3. Từ khoá công khai không thể tìm ra khoá bí mật.
4. Dữ liệu được mã hoá bằng khoá công khai, giải mã bằng khoá bí mật.
5. Mọi người đều có thể mã hoá nhưng chỉ một người có thể giải mã, chính người mã hoá cũng không thể giải mã.
6. Thường tính toán trên số lớn nên cho tốc độ thực thi thấp.

Kết luận chương:

Chương này đã trình bày tổng quan về mật mã và mật mã khoá công khai cũng như liệt kê ra được một số hệ mật mã hoá đối xứng và không đối xứng. Từ đó, chỉ ra được những tính chất của hệ mật mã khoá công khai.

CHƯƠNG 2: CHỮ KÝ SỐ

2.1 Định nghĩa chữ ký số và ví dụ.

***Khái niệm chữ ký số*:**

Theo quy định của pháp luật và căn cứ vào **Khoản 6 Điều 3 Nghị định 130/2018 Nghị định Chính Phủ** nêu rõ:

"Chữ ký số là một dạng chữ ký điện tử được tạo ra bằng sự biến đổi một thông điệp dữ liệu sử dụng hệ thống mật mã không đối xứng, theo đó, người có được thông điệp dữ liệu ban đầu và khóa công khai của người ký có thể xác định được chính xác:

Việc biến đổi nêu trên được tạo ra bằng đúng khóa bí mật tương ứng với khóa công khai trong cùng một cặp khóa;

Sự toàn vẹn nội dung của thông điệp dữ liệu kể từ khi thực hiện việc biến đổi nêu trên."

Bên cạnh đó, nếu hiểu theo tính ứng dụng thì chữ ký số được hiểu là một loại chữ ký điện tử. Chữ ký này sẽ thay thế hoàn toàn chữ ký thường bằng tay và sử dụng trên các thiết bị điện tử.

Vai trò của chữ ký số được hiểu như là một chữ ký tay của cá nhân hay một con dấu của cơ quan, doanh nghiệp. Sử dụng chữ ký số, các cá nhân, cơ quan, doanh nghiệp sẽ được pháp luật thừa nhận về mặt pháp lý khi thực hiện giao dịch trên môi trường điện tử. Tiêu biểu một số giao dịch như ký kê khai thuế, hợp đồng điện tử, giao dịch tài chính,...

Hình 2.1 Hình ảnh chữ ký số

Ví dụ:

Chữ ký số dùng cho cá nhân	Chữ ký số trong doanh nghiệp/tổ chức	Chữ ký số cho cá nhân thuộc tổ chức
– Mã hóa dữ liệu, bảo mật thông tin – Kê khai, quyết toán thuế thu nhập cá nhân – Giao dịch ngân hàng, tín dụng – Chứng khoán điện tử – Mua bán trực tuyến – Mua bán, thanh toán qua mạng – Ký hợp đồng lao động, hợp đồng kinh tế – Ký email, ký kết văn bản điện tử…	– Kê khai thuế điện tử – Hóa đơn điện tử – Khai hồ sơ BHXH điện tử – Khai báo Thống kê điện tử – Nộp thuế điện tử – Dịch vụ công Kho bạc Nhà nước – Hải quan điện tử – Giao dịch ngân hàng điện tử – Đăng ký doanh nghiệp – Mua bán, thanh toán qua mạng, thương mại điện tử B2B – Ký kết hợp đồng lao động, hợp đồng kinh tế, văn bản điện tử	Giao dịch nghiệp vụ trong nội bộ tổ chức hoặc đại diện tổ chức thực hiện giao dịch với bên ngoài khi được ủy quyền: – Nghiệp vụ nội bộ: Ký xác nhận văn bản điện tử, email, login hệ thống bảo mật công ty; Ký chứng từ trong giao dịch nội bộ như: thanh toán tạm ứng, phiếu thu, phiếu chi… – Giao dịch được tổ chức ủy quyền: Giao dịch/ thanh toán thương mại điện tử, ký

Chữ ký số dùng cho cá nhân	Chữ ký số trong doanh nghiệp/tổ chức	Chữ ký số cho cá nhân thuộc tổ chức
	– Chứng từ trong giao dịch nội bộ như: Phiếu tạm ứng, Phiếu thu, Phiếu chi, báo cáo quản trị…	kết văn bản điện tử, ngân hàng điện tử…

2.2 So sánh chữ ký số với chữ ký viết tay

Chữ ký số và chữ ký thông thường hiện nay đang được sử dụng phổ biến song song với nhau trong đời sống cá nhân và cả trong hoạt động của doanh nghiệp. Chữ ký số được sinh ra là để thay thế chữ ký tay thông thường.

Khái niệm chữ ký viết tay

Chữ ký thường hay còn gọi là chữ ký viết tay được sử dụng cho những văn bản giấy tờ thực. Nó được ký kết trên giấy dạng chữ viết do mỗi người tự tạo nên. Khi nhìn vào chữ viết đó có thể xác định được người ký kết các văn bản giấy tờ.

So sánh những điểm khác biệt giữa chữ ký số và chữ ký thông thường

Chữ ký số

- Tính chất: Chữ ký số có thể được hình dung như một dấu vân tay điện tử, được mã hoá và xác định người thực sự ký nó.
- Tiêu chuẩn: sử dụng các phương thức mã hoá mật mã.
- Tính năng: bảo mật tài liệu.
- Xác nhận: được thực hiện bởi các cơ quan chứng nhận tin cậy nhà cung cấp dịch vụ uy tín.
- Bảo mật: độ an toàn cao.

Chữ ký viết tay

- Tính chất: in ấn văn bản, ký tay,…
- Tiêu chuẩn: ký hàng loạt văn bản, hợp đồng, hoá đơn, thời gian chờ đợi chuyển tiếp.
- Tính năng: xác minh tài liệu.
- Xác nhận: không có quá trình xác nhận cụ thể.
- Bảo mật: dễ bị giả mạo, bắt chước và không có tính xác thực cao.

Từ những thông tin được trình bày ở trên, bạn sẽ thấy rằng: Chữ ký số là phương thức xác thực được phát triển để thay thế chữ ký tay trong hoạt động của doanh nghiệp. Chữ ký số được sử dụng rộng rãi và an toàn hơn chữ ký tay rất nhiều. Vì vậy các doanh nghiệp ngày nay hầu hết đã chuyển sang Chữ ký số để đảm bảo tính pháp lý và an toàn khi sử dụng.

Có thể hiểu rằng, Chữ ký số hiện nay chủ yếu dùng cho các hoạt động của doanh nghiệp. Trong khi đó, chữ ký tay thường được cá nhân sử dụng nhiều hơn mặc dù độ an toàn và bảo mật cũng như tính xác thực của chữ ký tay không cao như Chữ ký số.

Lý do sử dụng chữ ký số RSA:

Chữ ký số RSA là dạng chữ ký số sử dụng hệ mã hóa RSA để tăng độ an toàn và mức độ bảo mật cao hơn khi truyền đạt dữ liệu dạng số hóa. Hiện nay, RSA là chữ ký số được ứng dụng phổ biến trong các giao dịch trực tuyến và thương mại điện tử bởi khả năng đảm bảo an toàn khi điều kiện độ dài khóa đủ lớn.

1. Xác định nguồn gốc: Tuy vẫn có những thách thức về an ninh nhất định nhưng hệ mã hóa RSA vẫn khá an toàn. Hệ mã hóa RSA cho phép tạo chữ ký với khóa bí mật mà chỉ người chủ mới biết.

2. Dữ liệu được bảo toàn một cách toàn vẹn: Tin nhắn gửi từ khoá bí mật rất khó bị giả mạo bởi giá trị Hash cũng sẽ bị thay đổi theo nếu có thay đổi tin nhắn. Do đó, những kẻ nghe lén không thể thay đổi tin nhắn mặc dù có thể tìm cách đọc tin nhắn gốc và cả Hash của nó. Lý do là vì họ không có khoá bí mật để sửa đổi chữ ký số cho phù hợp.

2.3 Hàm băm và các tính chất của hàm băm.

Hàm băm:

là hàm thực hiện quá trình biến một dữ liệu đầu vào có độ dài bất kỳ thành một chuỗi đầu ra đặc trưng có độ dài cố định. Các giá trị được trả về bởi hàm băm được gọi là giá trị băm, mã băm, thông điệp băm, hoặc đơn giản là "hash". Điều này trở nên quan trọng khi bạn xử lý một lượng lớn dữ liệu và giao dịch. Khi đó, thay vì bạn phải xử lý toàn bộ lượng dữ liệu đầu vào (có thể có kích thước rất lớn), bạn chỉ cần xử lý và theo dõi một lượng dữ liệu rất nhỏ là các giá trị băm.

Giả dụ:

Bạn tải một video trên Youtube về, sau đó cho nó chạy qua hàm băm có tên MD5 sẽ trả về một chuỗi dài 32 ký tự, hoặc bạn tải một bức ảnh trên mạng về, cho chạy qua hàm MD5, thứ bạn nhận được vẫn là một chuỗi dài 32 ký tự. Thậm chí, nếu bạn cho chạy từ "apple" qua hàm hash MD5 kia, kết quả sẽ là "1f3870be274f6c49b3e31a0c6728957f", lại là một chuỗi có 32 ký tự. Những thuật toán băm khác cũng hoạt động tương tự như vậy, bạn cho bất kỳ thứ gì vào hàm, đầu ra sẽ luôn là một chuỗi có độ dài nhất định.

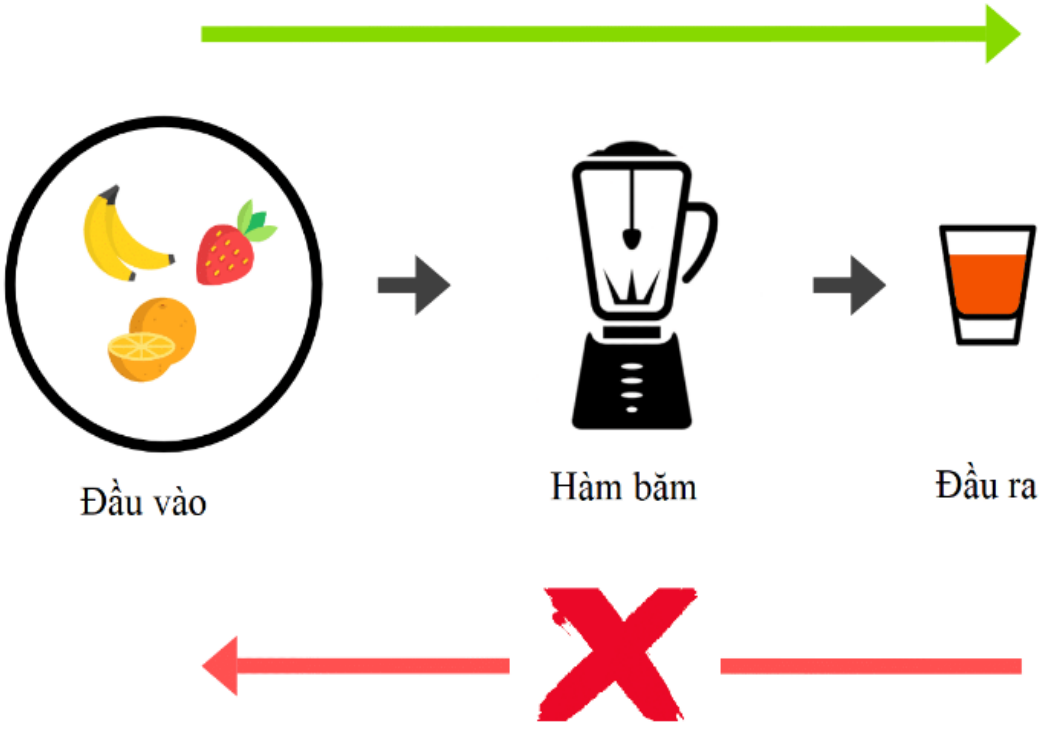
Hàm băm SHA-1, viết tắt của "Secure Hash Algorithm" (thuật toán băm an toàn) và như cái tên nói lên mục đích – mã SHA được tối ưu cho khả năng bảo mật, được phát triển như một phần của dự án Capstone của Chính phủ Hoa

Kỳ. Nó đã bị NSA rút lại ngay sau khi xuất bản và được thay thế bởi phiên bản sửa đổi, được xuất bản năm 1995 trong FIPS PUB 180-1 và thường được đặt tên là SHA-1. SHA-1 tạo ra bản tóm tắt có kích thước 160 bit (20 byte).

Hàm băm SHA được coi là hàm băm an toàn nhất bởi:

- Mã hóa hàm băm SHA tạo ra những kết quả băm không thể đảo ngược và là duy nhất. Không thể đảo ngược nghĩa là dù cho có được kết quả băm cũng không thể tìm ra dữ liệu ban đầu được băm, do đó đảm bảo tính bảo mật tuyệt đối của dữ liệu.
- Hai đoạn dữ liệu có cùng kết quả băm tạo ra bởi một trong những giải thuật SHA là không thể xảy ra. Chỉ cần một sự thay đổi nào trên đoạn dữ liệu gốc, dù nhỏ nhất, cũng sẽ tạo nên một giá trị băm hoàn toàn khác với hiệu ứng lở tuyết.

Chính vì thế, em đã sử dụng hàm băm SHA trong chương trình thử nghiệm của mình.



Hình 2.2: Ví dụ về minh hoạ về hàm băm

Các tính chất của hàm băm:

- Tính chất cơ bản của hàm băm mật mã là tính một chiều. Nghĩa là, một hàm mà trên thực tế không thể có ngược. Nếu bạn có một giá trị băm đầu ra, bạn sẽ không thể suy ngược lại được giá trị đầu vào là gì để có thể băm ra một thông điệp băm như vậy, hoặc ít nhất là rất khó suy luận được ra, trừ khi bạn vét cạn hết toàn bộ các khả năng có thể của thông điệp đầu vào. Đây là tính chất vô cùng quan trọng của hàm băm mật mã biến nó thành một công cụ cơ bản của mật mã hiện đại.

- Hàm băm có tính chất bảo mật, một chuỗi đầu vào cụ thể luôn tạo ra một giá trị đầu ra cụ thể và không thể tạo ra hai giá trị đầu ra khác nhau cho cùng một chuỗi đầu vào.

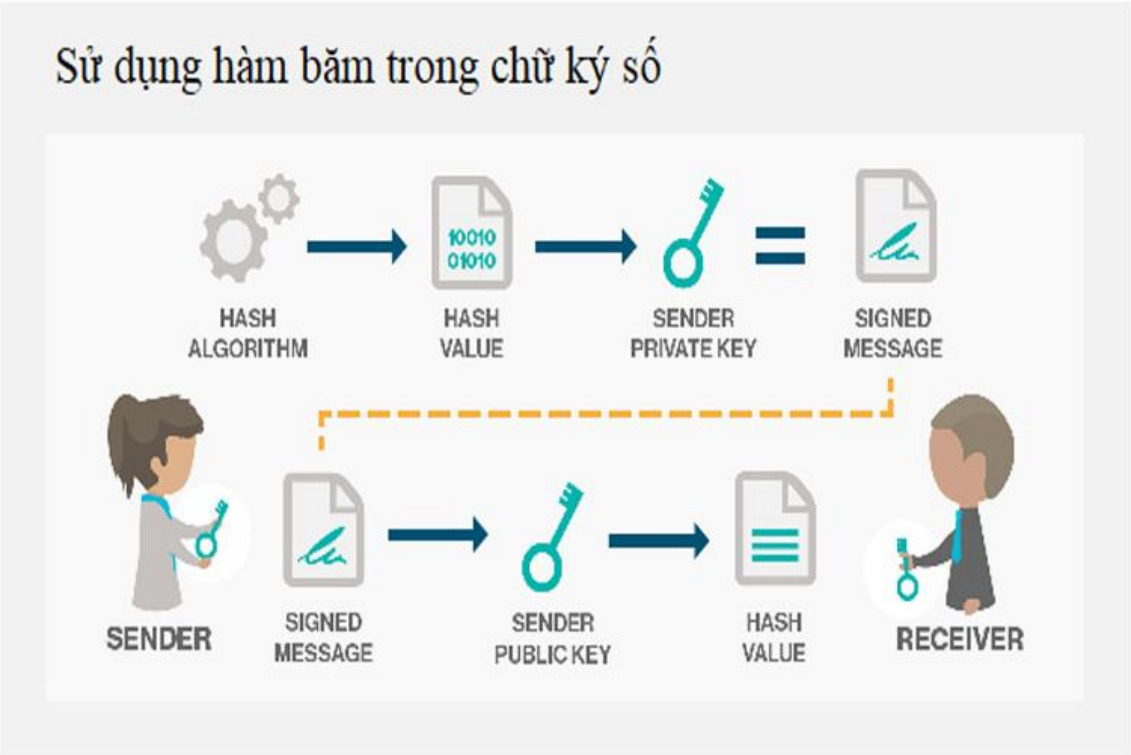
- Hàm băm có tính chất định hướng, hai chuỗi đầu vào khác nhau sẽ tạo ra hai giá trị đầu ra khác nhau.

2.4 Vai trò của hàm băm với chữ ký số.

Các hàm băm mật mã có nhiều ứng dụng trong an toàn thông tin. Nó được sử dụng nhiều trong chữ ký số, mã xác thực thông điệp và các hình thức xác thực khác.

Hàm băm trợ giúp cho các chữ ký số nhằm giảm dung lượng của dữ liệu cần thiết.

Không những thế nó thường kết hợp với chữ ký số để tạo ra một loại chữ ký điện tử vừa an toàn hơn (không thể cắt/dán) vừa có thể dùng để kiểm tra tính toàn vẹn của thông điệp.



Hình 2.3 Vai trò của hàm băm với chữ ký số.

Hàm băm giúp nâng cao hiệu quả chữ ký số.

Hầu như tất cả các lược đồ chữ ký số đều yêu cầu tính toán bản tóm lược của thông điệp bằng các hàm băm mật mã. Điều này cho phép việc tính toán và tạo chữ ký được thực hiện trên một khối dữ liệu có kích thước tương đối nhỏ và cố định thay vì trên toàn bộ văn bản dài. Tính chất toàn vẹn thông điệp của hàm băm mật mã được sử dụng để tạo các chữ ký số an toàn và hiệu quả.

Kết luận chương:

Chương 2 đã nêu được tổng quan về chữ ký số là gì, sự khác nhau giữa chữ ký số với chữ ký viết tay như về tính chất, tiêu chuẩn, tính năng, bảo mật, xác nhận. Đồng thời chương này cũng tìm hiểu về hàm băm, các tính chất cơ bản của hàm băm như tính một chiều, tính bảo mật, tính định hướng cũng như trình bày được vai trò của hàm băm với chữ ký số.

CHƯƠNG 3: Hệ mật RSA

3.1 Khái niệm và tính chất của mật mã RSA

**Khái niệm:*

RSA là một thuật toán mật mã khóa công khai 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. Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa. Nó đánh dấu một sự tiến bộ vượt bậc của lĩnh vực mật mã học trong việc sử dụng khóa công cộng. 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.

**Tính chất của mật mã RSA:*

1. Trong các hệ mật mã RSA, một bản tin có thể được mã hoá trong thời gian tuyến tính.

Đối với các bản tin dài, độ dài của các số được dùng cho các khoá có thể được coi như là hằng. Tương tự như vậy, nâng một số lên luỹ thừa được thực hiện trong thời gian hằng, các số không được phép dài hơn một độ dài hằng. Thực ra tham số này che giấu nhiều chi tiết cài đặt có liên quan đến việc tính toán với các số dài, chi phí của các phép toán thực sự là một yếu tố ngăn cản phổ biến ứng dụng của phương pháp này. Phần quan trọng nhất của việc tính toán có liên quan đến việc mã hoá bản tin. Nhưng chắc chắn là sẽ không có hệ mã hoá nào hết nếu không tính ra được các khoá của chúng là các số lớn.

2. Các khoá cho hệ mã hoá RSA có thể được tạo ra mà không phải tính toán quá nhiều.

Một lần nữa, ta lại nói đến các phương pháp kiểm tra số nguyên tố. Mỗi số nguyên tố lớn có thể được phát sinh bằng cách đầu tiên tạo ra một số ngẫu nhiên

lớn, sau đó kiểm tra các số kế tiếp cho tới khi tìm được một số nguyên tố. Một phương pháp đơn giản thực hiện một phép tính trên một con số ngẫu nhiên, với xác suất 1/2 sẽ chứng minh rằng số được kiểm tra không phải nguyên tố. Bước cuối cùng là tính p dựa vào thuật toán Euclid.

Như phần trên đã trình bày trong hệ mã hoá công khai thì khoá giải mã (private key) Kb và các thừa số p,q là được giữ bí mật và sự thành công của phương pháp là tuỳ thuộc vào kẻ địch có khả năng tìm ra được giá trị của Kb hay không nếu cho trước N và Kb. Rất khó có thể tìm ra được Kb từ Kb cần thiết về p và q, như vậy cần phân tích N ra thành thừa số để tính p và q. Nhưng việc phân tích ra thừa số là một việc làm tốn rất nhiều thời gian, với kỹ thuật hiện đại ngày này thì cần tới hàng triệu năm để phân tích một số có 200 chữ số ra thừa số.

Độ an toàn của thuật toán RSA dựa trên cơ sở những khó khăn của việc xác định các thừa số nguyên tố của một số lớn. Bảng dưới đây cho biết các thời gian dự đoán, giả sử rằng mỗi phép toán thực hiện trong một micro giây.

Số các chữ số trong số được phân tích

Bảng sau cho biết một số thao tác cần thiết để phân tích n thành thừa số và thời gian cần thiết (giả sử một thao tác cần 1 micro giây).

Chữ số	Số lượng thao tác	Thời gian
50	1.4*10^10	3.9 giờ
75	9.0*10^12	104 ngày
100	2.3*10^15	74 năm
200	1.2*10^23	3.8*10^9 năm
300	1.5*10^29	4.8*10^15 năm
500	1.3*10^39	4.2*10^25 năm

Theo khuyến nghị chiều dài n nên vào khoảng 200 chữ số, dài hơn hay ngắn hơn phụ thuộc vào tốc độ mã hoá và tính bảo mật trong từng ứng dụng.

3.2 Một số lỗ hổng của RSA ta cần lưu ý.

Một lỗ hổng RSA nghiêm trọng mới được các công ty Microsoft, Google, Lenovo, HP và Fujitsu cảnh báo, ảnh hưởng tới hàng tỷ thiết bị. Lỗ hổng CVE-2017-15361 là lỗ hổng liên quan tới mã hóa nằm trong phương thức tạo cặp khóa RSA bởi Trusted Platform Module (TPM) thuộc công ty Infineon. TPM Infineon được sử dụng rỗng rãi, thiết kế vi xử lý chuyên biệt nhằm bảo vệ phần cứng bằng cách tích hợp khóa mã hóa vào thiết bị và được sử dụng cho quá trình mã hóa an toàn.

Lỗ hổng RSA được phát hiện bởi các nhà nghiên cứu tại đại học Masaryk, cộng hòa Séc. Các nhà nghiên cứu cũng đã phát hành bài đăng chi tiết về lỗ hổng cũng như một công cụ trực tuyến nhằm kiểm tra khóa RSA có bị ảnh hưởng bởi lỗ hổng hay không.

ROCA: Tấn công khôi phục khóa riêng tư RSA

Tấn công khôi phục khóa được các nhà nghiên cứu đặt tên ROCA (Return of Coppersmith’s Attack), cho phép tin tặc tính toán ngược khóa mã hóa riêng tư thông qua khóa công khai. Tin tặc từ đó có thể giả mạo, giải mã dữ liệu của nạn nhân, đưa mã độc vào chứng chỉ kí số và vượt các cơ chế bảo vệ.

Tấn công ROCA ảnh hưởng hàng tỷ thiết bị

Tấn công ROCA ảnh hưởng trên các chip xử lý sản xuất bởi Infineon kể từ năm 2012, được sử dụng rộng rãi trong thẻ căn cước, bo mạnh chủ để lưu trữ mật khẩu, mã ủy quyền, khi duyệt web an toàn, khi các phần mềm và ứng dụng kí số và cơ chế bảo vệ tin nhắn như PGP. Lỗ hổng làm suy yến an ninh chính phủ và các tổ chức sử dụng thư viện và chip mã hóa của Infineon. Hầu hết các phiên bản Windows, thiết bị Google Chromebook phát triển bởi HP, Lenovo và

Fujitsu có nguy cơ bị tấn công. Số lượng khóa mã hóa được xác định bị ảnh hưởng khoảng 760,000 nhưng có khả năng cao hơn nhiều lần.

Số n nhỏ (Theo Hồ Văn Canh (2017): 20 năm tấn công mật mã RSA)

Nếu số n nhỏ (chiều dài n < 256 bit), số n có thể bị tách ra thành các thừa số nguyên tố dễ dàng bởi các công cụ có sẵn như factordb.

Chiều dài của số n được khuyến cáo là 1024 bit.

Nhưng đã có những trường hợp số n lớn, nhưng phân tách của n thành thừa số nguyên tố đã có sẵn trong cơ sở dữ liệu của các trang như factordb hoặc alpertron.

Đây là 1 cách tìm p và q rất dễ nên thường được thử đầu tiên.

Số e nhỏ, số m nhỏ(Theo Hồ Văn Canh (2017): 20 năm tấn công mật mã RSA)

1 nhược điểm lớn của mã hoá RSA đó là tốc độ mã hoá chậm hơn nhiều so với mã hoá DES, do vậy trong một vài trường hợp để tăng tốc độ, người mã hoá sẽ mã hoá tài liệu bằng 1 mã hoá khác nhưng khoá sẽ được mã hoá bằng RSA.

Đồng thời để tối ưu hoá thời gian mã hoá, số e cũng được chọn theo dạng e = 2n+1, khi đó e nhỏ nhất là e = 3.

Nếu ta chọn số e nhỏ và tin nhắn M (m nhỏ) -> ciphertext: c = m3 (mod n). Vì m nhỏ nên m3 < n khi đó phép toàn module không có tác dụng, Vì vậy để tìm người lại m từ c ta có m = c1/3

Tấn công lặp liên tục (Theo Hồ Văn Canh (2017): 20 năm tấn công mật mã RSA)

Ban đầu ta có 2 số p, q lần lượt là 3, 5 => n = p*q = 15 => chọn m = 7.

Tính φ(n) = (p-1)*(q-1) = 8 => chọn e = 3.

Tính c = me mod n = 13

c1 = ce mod n = 7

c2 = c1e mod n = 13

Do c2 = c => m = c1 = 7

Hiệu p – q nhỏ - Fermat Attack (Theo Hồ Văn Canh (2017): 20 năm tấn công mật mã RSA)

p, q được chọn có cùng độ dài bit để tạo được 1 mã RSA mạnh, nhưng điều này có thể khiến q,p quá gần nhau khiến cho kẻ tấn công dễ dàng phân tách n thành thừa số nguyên tố. Điều kiện (p – q) < n¼

Số e trùng nhau, số e nhỏ - Hastad Broadcast Attack (Theo Hồ Văn Canh (2017): 20 năm tấn công mật mã RSA)

Trong mạng LAN, đôi khi số e được đặt giống nhau đối với các máy tình cùng mức độ. Nghĩa là e1 = e2 = ... = e = 3

Kịch bản tấn công xảy ra nếu máy chủ gửi cùng 1 tin nhắn broadcast m (đã được mã hóa thành c1, c2, ... cho nhiều máy tính trong mạng, và ta bắt được ít nhất e ciphertext c1, c2, ..., ce. Lúc này, ta sẽ có thể khôi phục lại plaintext m không mấy khó khăn.

Giả sử e = 3, đặt M = m3. Nhiệm vụ của ta là giải hệ phương trình đồng dư:

Sau khi tính được M, ta sẽ tìm được m (bằng căn bậc 3 của M)

Số n trung nhau - Common modulus (Theo Hồ Văn Canh (2017): 20 năm tấn công mật mã RSA)

Giống với ví dụ ở phần trước nhưng thay vì e trùng nhau, lần này n trùng nhau, nghĩa là n1 = n2 = ... = n và số e được chọn ngẫu nhiên. Như vậy mỗi thành viên trong mạng lưới sẽ được cấp một bộ tham số (n,ei,di) riêng.

Vì ed ≡ 1 (mod φ(n)) nên tồn tại số k sao cho ed - kφ(n) = 1. Do đó: k = (ed-1)/φ(n) > (ed-1)/n

Vậy ta sẽ brute-force số k từ (ed-1)/n trở lên, tính ngược lại φ(n) = (ed-1)/k, cho đến khi thu được kết quả φ(n) là số nguyên. Có φ(n)ta dễ dàng tính được Private Key của victim: dvictim = evictim-1 mod φ(n)

Phân phối khoá (Theo Hồ Văn Canh (2017): 20 năm tấn công mật mã RSA)

Giả sử C có thể gửi cho A một khóa bất kỳ và có thể khiến cho A tin đó là khóa công khai của B. Đồng thời C có thể đọc thông tin trao đổi giữa A và B. Khi đó, C sẽ gửi cho A khóa công khai của chính mình (mà A nghĩ rằng đó là khóa của B). Sau đó C sẽ đọc tất cả văn bản mã hóa do A gửi, giải mã với khóa bí mật của mình, giữ 1 bản copy, đồng thời mã hóa bằng khóa công khai của B và gửi cho B

3.3 Thuật toán ký và xác thực.

Việc ký tên và xác thực chữ ký số sử dụng hệ mã hóa RSA tương tự như quá trình mã hóa và giải mã ở trên. Tuy nhiên vai trò của public key và private thì có thay đổi đôi chút.

Để tạo chữ ký, người gửi sẽ dùng private key và người nhận sẽ dùng public key để xác thực chữ ký đó.

Tuy nhiên, vì bản tin rất dài nên việc mã hóa toàn bộ bản tin sẽ rất mất thời gian. Vì vậy, trong thực hành, chữ ký số thường sử dụng phương pháp mã hóa giá trị hash của bản tin. Việc này mang lại rất nhiều lợi ích như:

- 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.
- Giá trị hash còn có thể dùng để kiểm tra lại bản tin nhận được có nguyên vẹn hay không.

Chữ ký số đem lại nhiều giá trị hơn chữ ký tay rất nhiều. Có lẽ cũng vì vậy, việc xử lý chữ ký số phức tạp hơn hẳn chữ ký tay truyền thống.

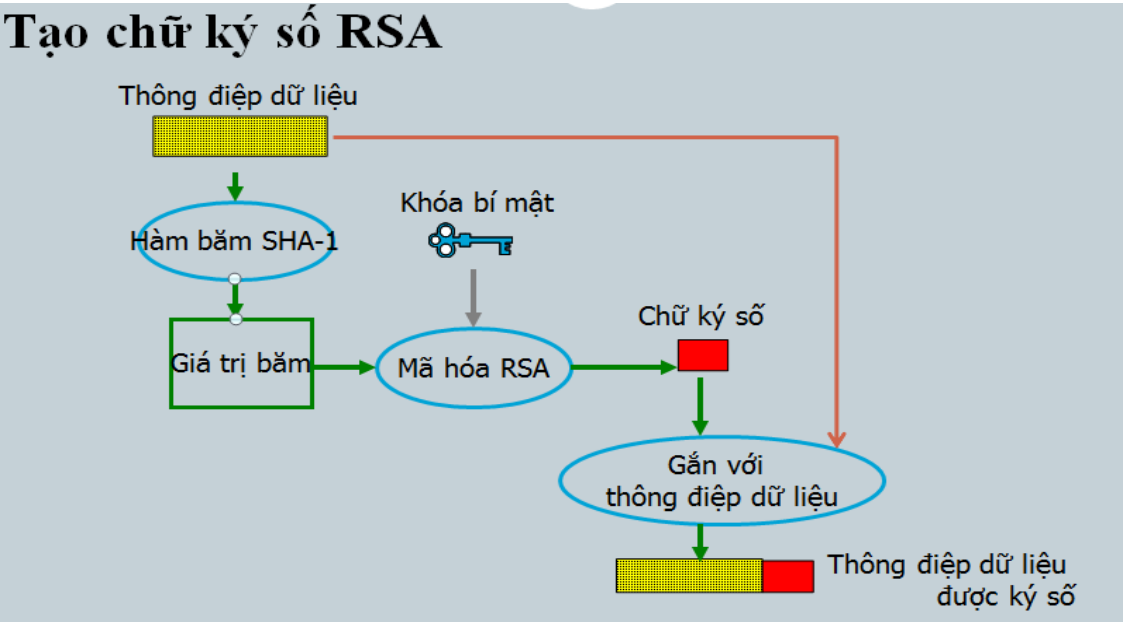
Quá trình ký (bên gửi)

Tính toán chuỗi đại diện (message digest/ hash value) của thông điệp sử dụng một giải thuật băm (Hashing algorithm) SHA-1

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

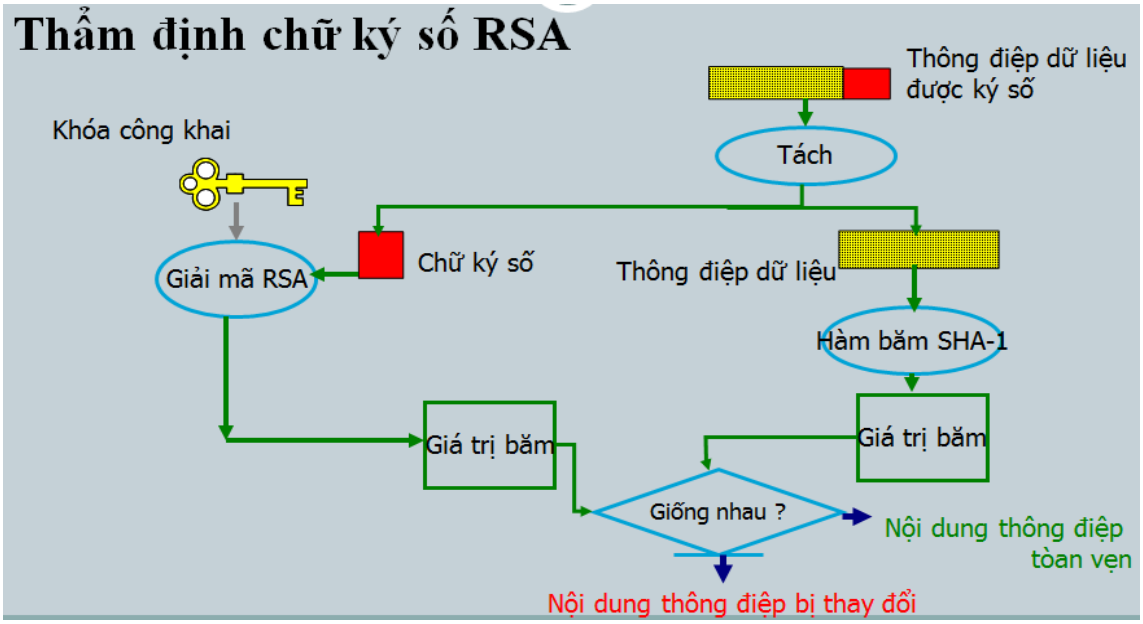
Thông điệp đã được ký (Signed message) được gửi cho người nhận.



Hình 3.1 Sơ đồ thuật toán tạo chữ ký số.

Xác thực nguồn gốc

Hệ mã hóa bất đối xứng cho phép tạo chữ ký với private key mà chỉ người chủ mới biết. Khi nhận gói tin, người nhận xác thực chữ ký bằng cách dùng public key giải mã, sau đó tính giá trị hash của bản tin gốc và so sánh với hash trong gói tin nhận được. Hai chuỗi này phải trùng khớp với nhau.



Hình 3.2 Sơ đồ thuật toán xác thực chữ ký.

Quá trình kiểm tra chữ ký (bên nhận)

Tách chữ ký số RSA 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 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).

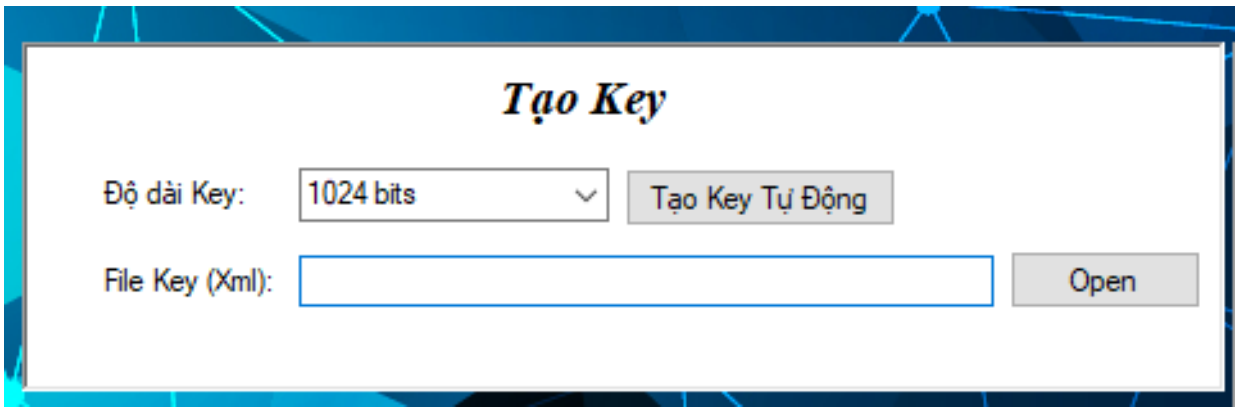
Nếu MD1 <>MD2 -> chữ ký không hợp lệ. Thông điệp có thể đã bị sửa đổi hoặc không thực sự xuất phát từ người gửi.

3.4 Cài đặt chương trình thử nghiệm

Chương được chia làm 4 khối:

- Khối tạo key.
- Khối thông tin key.
- Khối tạo chữ ký số.
- Khối thẩm định chữ ký.

*Đầu tiên là khối tạo key



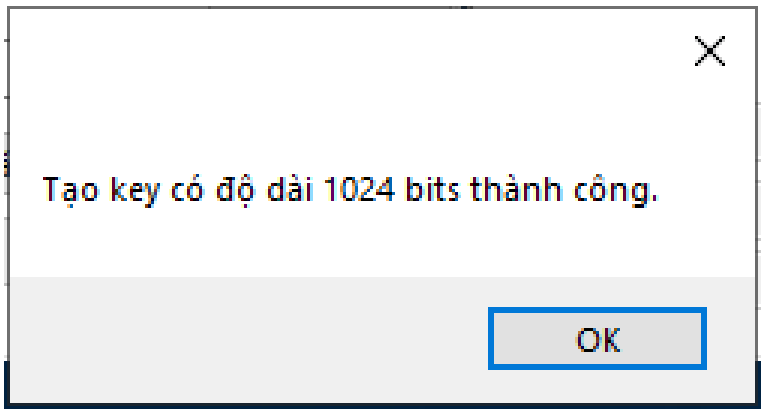
Hình 3.3 Khối tạo key.

Có thể mở tập tin key đã có hoặc có thể tạo key.

Độ dài key cung cấp 4 độ dài là: 512 bits, 1024 bits, 2048 bits, 4096 bits.

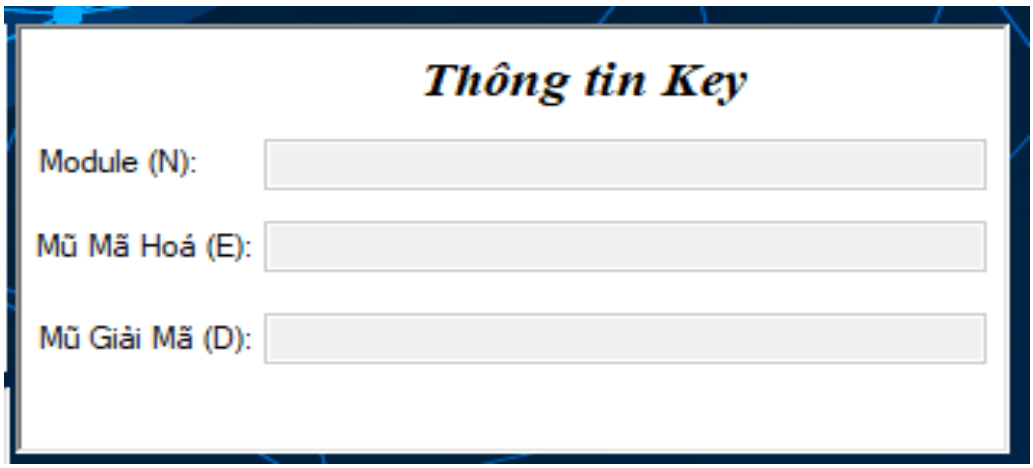
Tập tin key là tập tin xml.

Khi tạo key sẽ hiện bảng thông báo.



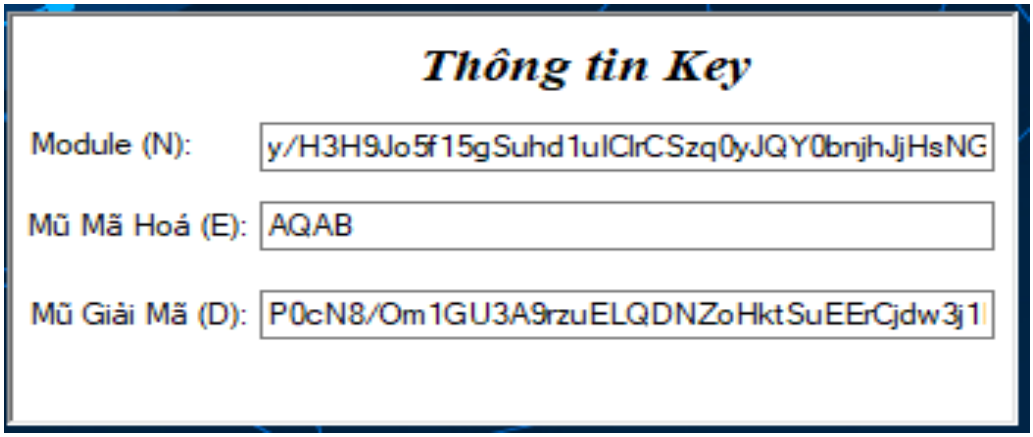
Hình 3.4 Bảng thông báo.

*Khối thứ 2 là thông tin key.



Hình 3.5 Khối thông tin key

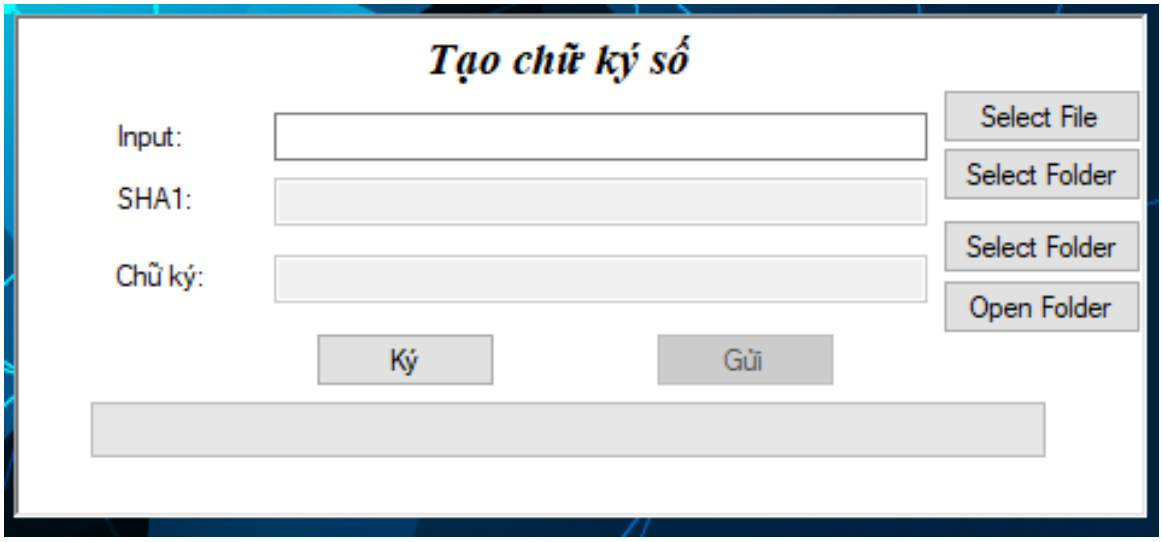
Khối này sẽ có Module (N), mũ mã hoá (E), mũ giải mã (D)



Hình 3.5 Khối thông tin key

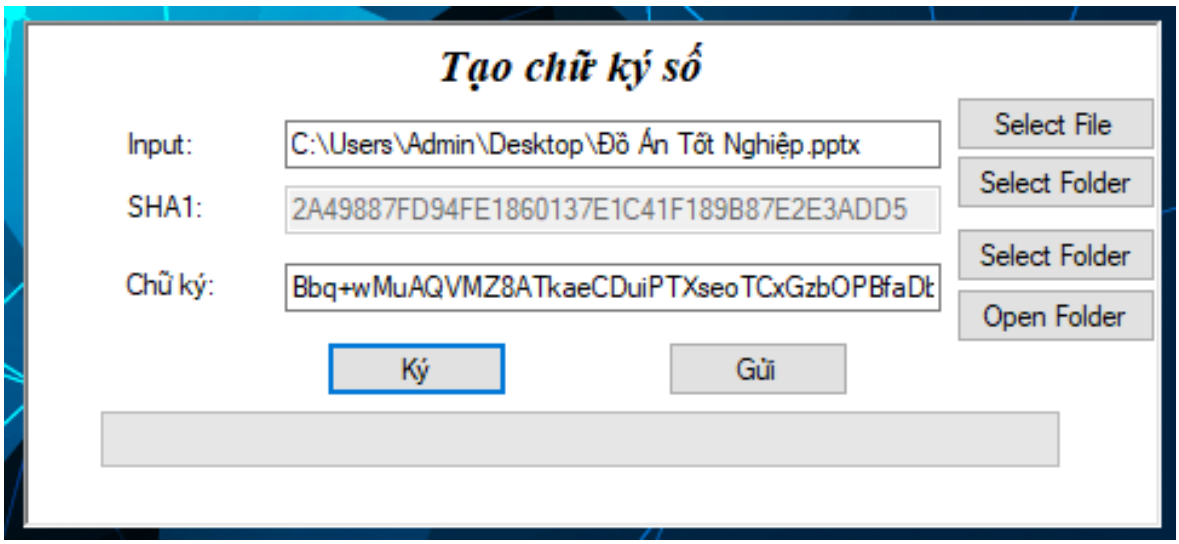
Khi tạo key thành công sẽ hiển thị ra thông tin key.

*Khối thứ 3: Tạo chữ ký số



Hình 3.6 Khối tạo chữ ký số.

Trong khối này có thể ký và gửi được một file tuỳ chọn.

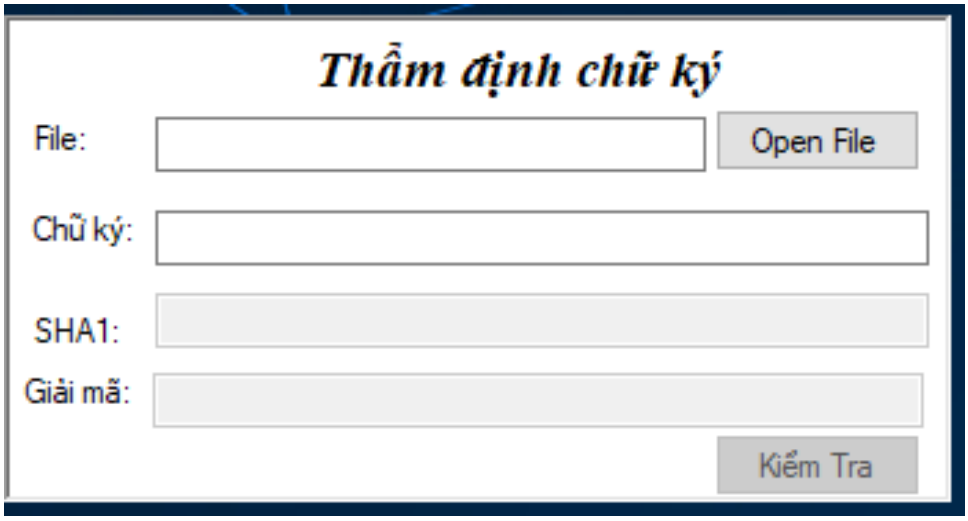


Hình 3.6 Khối tạo chữ ký số.

Chọn đường dẫn file để ký, thanh tiến trình sẽ chạy và bảng thông báo thời gian thực thi sẽ hiện lên.

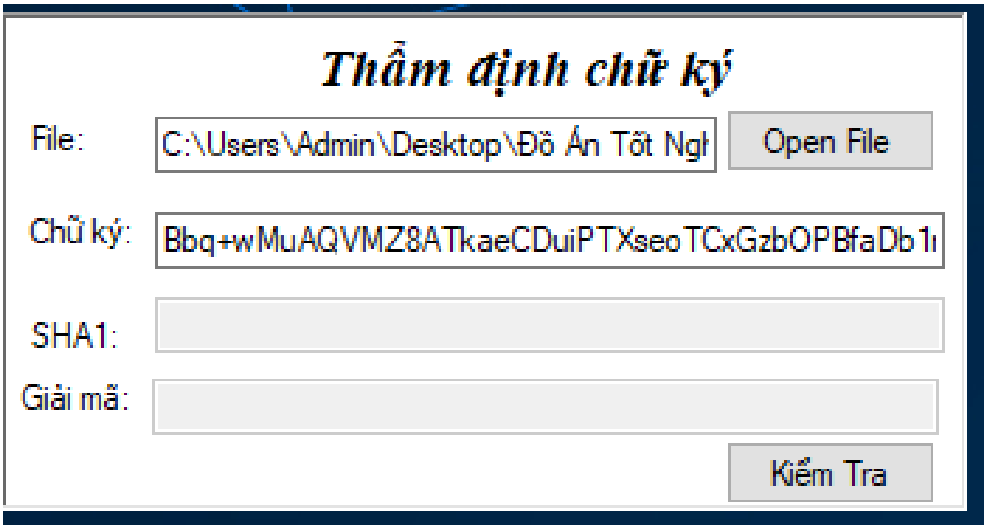
Khi ký thành công ta sẽ gửi file vừa được ký sang khối thẩm định chữ ký.

*Khối cuối cùng: Thẩm định chữ ký



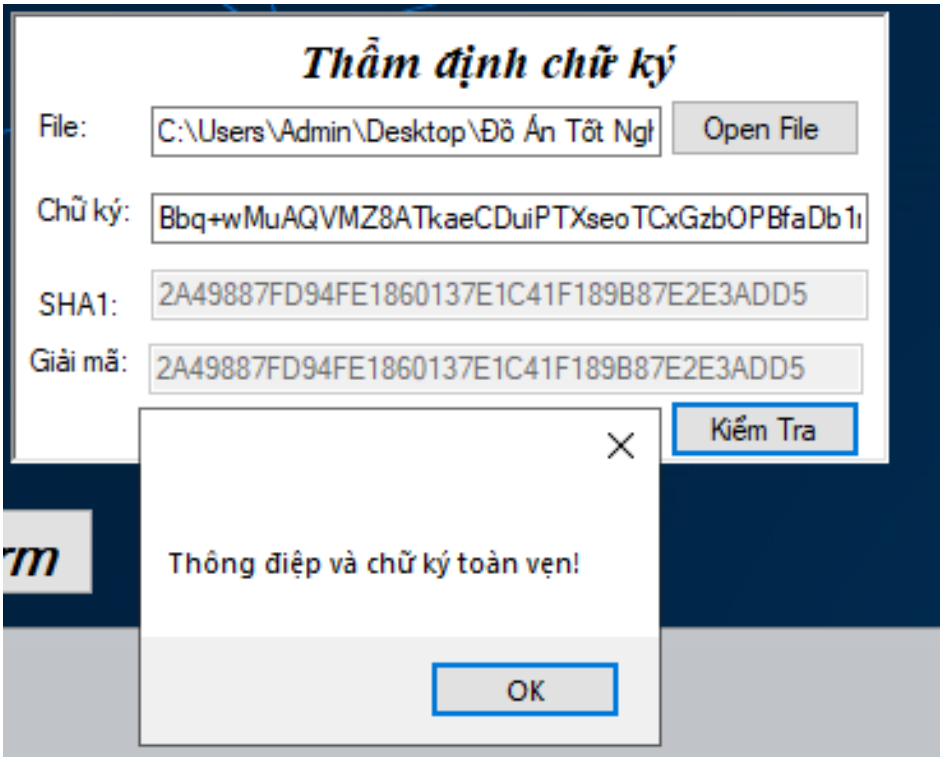
Hình 3.7 Khối thẩm định chữ ký.

Ta có thể chọn file vừa được ký để kiểm tra.



Hình 3.7 Khối thẩm định chữ ký.

Sau khi ấn "Kiểm tra" file xong chương trình sẽ chạy và hiện thị thông báo.



Mã SHA1và giải mã RSA trùng khớp với file cũ nên thông điệp và chữ ký của file được bảo toàn.

Kết luận chương:

Chương 3 đã trình bày về RSA đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa và một vài tính chất của hệ mật mã RSA, cũng như đã nêu ra được một số lỗ hổng quan trọng của RSA ta cần lưu ý như lỗ hổng liên quan tới mã hóa nằm trong phương thức tạo cặp khóa, tấn công khôi phục khóa riêng tư RSA,… Phần cuối chương là thuật toán ký và xác thực chữ ký RSA giúp ta hiểu được cách vận hành của thuật toán và cách xác thực chữ ký.

Kết luận

Chữ ký số là một công nghệ cho phép xác định chủ nhân của dữ liệu nào đó và kiểm tra dữ liệu có nguyên vẹn hay không. Còn hệ mã hóa RSA cho phép tạo chữ ký với chìa khoá bí mật mà chỉ người chủ mới biết.Tuy nhiên hệ mã hóa RSA vẫn có những thách thức về an ninh nhất định nhưng dù sao thì nó vẫn khá an toàn.

Trên đây là bài nghiên cứu ứng dụng hệ mật mã RSA trong chữ ký điện tử của em. Trong quá trình làm đồ án, em đã đạt được một số kết quả như sau:

- Trình bày về mật mã và mật mã khoá công khai cùng các tính chất của nó.
- Tìm hiểu về chữ ký số, hàm băm cùng các tính chất của hàm băm và vai trò của hàm băm với chữ ký số.
- Trình bày về mã khoá RSA và các tính chất của nó, một số lỗ hổng của RSA cũng như tìm hiểu về ký và xác thực chữ ký RSA.
- Xây dựng được chương trình chạy thử nghiệm.

Bên cạnh những phần đã đạt được, đồ án vẫn tồn tại một số hạn chế như sau:

- Những nội dung đã trình bày và nghiên cứu chỉ dừng lại ở mức độ tìm hiểu, chưa nghiên cứu sâu.
- Mã hoá RSA về thuật toán ký và xác thực còn nhiều thiếu sót.
- Chương trình ứng dụng còn đơn giản.
- Giao diện chương trình chưa thân thiện.

Mặc dù đã cố gắng trong việc nghiên cứu và thực hiện đồ án, nhưng do sự hiểu biết của em còn hạn chế, kinh nghiệm cũng như thời gian có hạn của một sinh viên, đồ án tốt nghiệp này không thể tránh được những thiếu sót. Em rất mong nhận được sự góp ý, chỉ bảo của các thầy cô để em có thể khắc phục khuyết điểm làm cho chương trình được hoàn thiện hơn. Em xin chân thành cảm ơn các Thầy Cô !

Tài liệu tham khảo

[1] Hồ Văn Canh (2017): 20 năm tấn công mật mã RSA (bản dịch Việt ngữ).

[2] Neal Koblitz (2000): A Course in Number Theory and Cryptography. NXB: Springer Verlag, NewYork, Berlin Heidelberg, London, Pái, Tokyo, 2000.

[3] Phan Đình Diệu (2002): Mật mã và an toàn thông tin. NXB: ĐHQG Hà Nội, năm 2002.

[4] Trịnh Nhật Tiến (2003): Mật mã và an toàn CSDL. NXB ĐHQG Hà Nội, năm 2003.