**DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT**

**Các ký hiệu**

gcd(a,b) Ước số chung lớn nhất của a và b

H(.) Hàm băm

|| Toán tử nối/trộn 2 xâu

a|b a là ước số của b

M Thông điệp dữ liệu

Khóa bí mật của thực thể cuối

Khóa công khai của thực thể cuối

**Các chữ viết tắt**

RSA Rivest Shamir Adleman

SHA Secure Hash Alorithm

MD5 Message Digest 5



MỤC LỤC

* [LỜI CẢM ƠN 1](#_Toc369123328)
* [Chương 1](#_Toc369123329)
* MỞ ĐẦU

[1.1. Tên đề tài 2](#_Toc369123331)

[1.2. Thông tin đồ án 2](#_Toc369123332)

[1.3. Mục đích nghiên cứu 2](#_Toc369123333)

[1.4. Các lý thuyết cần tìm hiểu khi làm đồ án 2](#_Toc369123334)

[1.5. Tính cấp thiết của đề tài 2](#_Toc369123335)

* [Chương 2](#_Toc369123336)
* [TỔNG QUAN VỀ CHỮ KÝ SỐ](#_Toc369123337)

[2.1. An toàn thông tin và hướng nghiên cứu đề tài 5](#_Toc369123338)

[2.1.1.Các nguy cơ mất an toàn 5](#_Toc369123339)

[2.1.2.Các giải pháp bảo đảm 6](#_Toc369123340)

[2.1.3.Hướng nghiên cứu của đề tài 10](#_Toc369123341)

[2.2. Chữ ký số 11](#_Toc369123342)

[2.2.1.Tổng quan về CKS 11](#_Toc369123343)

[2.2.2.Một số hệ chữ ký số điển hình 14](#_Toc369123344)

* [Chương 3](#_Toc369123345)
* [CƠ SỞ TOÁN HỌC](#_Toc369123346)

[3.1. Đại số trừu tượng 19](#_Toc369123347)

[3.1.1.Nhóm (Đại số) 19](#_Toc369123348)

[3.1.2.Vành 19](#_Toc369123349)

[3.1.3.Trường 21](#_Toc369123350)

[3.2. Lý thuyết số 21](#_Toc369123351)

[3.2.1.Phép modulo 21](#_Toc369123352)

[3.2.2.Số nguyên tố 23](#_Toc369123353)

[3.2.3.Hàm Euler 24](#_Toc369123354)

[3.3. Cơ sở xây dựng các hệ mật khóa công khai và chữ ký số 24](#_Toc369123355)

[3.3.1.Bài toán phân tích số 24](#_Toc369123356)

[3.3.2.Bài toán logarit rời rạc 24](#_Toc369123357)

[3.3.3.Bài toán khai căn 25](#_Toc369123358)

[3.3.4.Hàm băm mật mã 25](#_Toc369123359)

* [Chương 4](#_Toc369123360)
* [LƯỢC ĐỒ CHỮ KÝ SỐ HD – 11](#_Toc369123361)

[4.1. Yêu cầu của một hệ thống chữ ký điện tử 34](#_Toc369123362)

[4.1.1.Mức độ an toàn của hệ mật RSA 34](#_Toc369123363)

[4.1.2.Kiểu tấn công module n dùng chung 35](#_Toc369123364)

[4.2. Lược đồ chung của chữ ký điện tử 36](#_Toc369123365)

[4.2.1.Các bước thực hiện tạo chữ ký điện tử 36](#_Toc369123366)

[4.2.2.Các bước thực hiện kiểm tra tính đúng của chữ ký điện tử 37](#_Toc369123367)

[4.2.3.Các bước thực hiện mã hoá và tạo chữ ký cho thông điệp 38](#_Toc369123368)

[4.2.4.Mô tả các bước kiểm tra chữ ký và giải mã thông điệp 38](#_Toc369123369)

[4.3. Phương pháp hình thành và kiểm tra chữ ký số 38](#_Toc369123370)

[4.3.1.Thuật toán hình thành khóa 38](#_Toc369123371)

[4.3.2.Thuật toán hình thành chữ ký 39](#_Toc369123372)

[4.3.3.Thuật toán kiểm tra chữ ký 39](#_Toc369123373)

[4.4. Tính đúng đắn và mức độ an toàn của lược đồ 40](#_Toc369123374)

[4.4.1.Tính đúng đắn 40](#_Toc369123375)

[4.4.2.Mức độ an toàn 40](#_Toc369123376)

* [Chương 5](#_Toc369123377)
* [CÀI ĐẶT THỬ NGHIỆM HD  - 11](#_Toc369123378)

[5.1. Một số thuật toán cơ sở 42](#_Toc369123379)

[5.1.1.Thuật toán sinh số nguyên tố 42](#_Toc369123380)

[5.1.2.Thuật toán tìm ước số chung lớn nhất 42](#_Toc369123381)

[5.1.3.Thuật toán tính nghịch đảo 42](#_Toc369123382)

[5.1.4.Thuật toán tìm phần tử sinh của nhóm 43](#_Toc369123383)

[5.1.5.Thuật toán "bình phương và nhân" 43](#_Toc369123384)

[5.2. Cài đặt thử nghiệm thuật toán 44](#_Toc369123385)

[5.2.1.Tạo khóa 44](#_Toc369123386)

[5.2.2.Sinh chữ ký 47](#_Toc369123387)

[5.2.3.Việc sinh chữ ký này phải thông qua việc tạo khóa. 47](#_Toc369123388)

[5.2.4.Xác thực chữ ký 47](#_Toc369123389)

* [KẾT LUẬN 49](#_Toc369123390)
* [TÀI LIỆU THAM KHẢO 50](#_Toc369123391)

**Danh mục các hình vẽ**

|  |  |  |
| --- | --- | --- |
| **Hình vẽ** |  | **Trang** |
| Hình 2.1 | Trao đổi khoá trong mật mã đối xứng | 6 |
| Hình 2.2 | Mô hình của mật mã khóa công khai | 8 |
| Hình 2.3 | Quy trình tạo và kiểm tra chữ ký số | 11 |
| Hình 2.4 | Quy trình thẩm định chữ ký số | 11 |
| Hình 2.5 | Quá trình mã hóa C = | 13 |
| Hình 2.6 | Quá trình giải mã P = | 13 |
| Hình 2.6 | Trao đổi khoá trong mật mã đối xứng | 24 |
| Hình 3.1 | Sơ đồ minh họa hàm băm (HASH) | 24 |
| Hình 3.2 | Sơ đồ thực hiện tổng quát của thuật toán MD5 | 27 |
| Hình 3.3 | Sơ đồ xử lý một khối 512bis thứ i | 29 |
| Hình 3.4 | Sơ đồ xử lý một bước trong 64 bước của thuật toán MD5 | 29 |
| Hình 4.1 | Mô hình tổng quát của chữ ký điện tử | 34 |
| Hình 4.2 | Sơ đồ minh họa các bước tạo chữ ký điện tử | 35 |
| Hình 4.3 | Sơ đồ minh họa các bước kiểm tra chữ ký điện tử | 35 |
| Hình 4.4 | Mô hình chữ ký điện tử dùng quá trình mã hóa và giải mã | 36 |
| Hình 5.1 | Sơ đồ chương trình chữ ký số | 45 |

# **LỜI CẢM ƠN**

Trước hết em xin bày tỏ lòng biết ơn sâu sắc nhất tới thầy giáo Ths. Lưu Hồng Dũng đã tận tình hướng dẫn, giúp đỡ và tạo mọi điều kiện thuận lợi để hoàn thành tốt đồ án của mình

Em xin chân thành cảm ơn các thầy cô khoa Công nghệ thông tin trường Học viện kỹ thuật quân sự đã tạo điều kiện cho em thực hiện đề tài.

Cuối cùng, tôi xin cảm ơn các bạn trong lớp đã giúp đỡ và đóng góp ý kiến cho đồ án tốt của tôi!

Sinh viên

Vũ Thị Kim Anh

# Chương 1

# MỞ ĐẦU

* 1. **Tên đề tài**

Nghiên cứu cài đặt thử nghiệm thuật toán chữ ký số HD 11

* 1. **Thông tin đồ án**

Sinh viên thực hiện : Vũ Thị Kim Anh

Lớp : VB2K15

Giáo viên hướng dẫn :Ths.Lưu Hồng Dũng

* 1. **Mục đích nghiên cứu**

Nghiên cứu về lý thuyết mật mã, mật mã hoá khoá công khai RSA, chữ ký số và ứng dụng thuật toán RSA trong mã hoá dữ liệu. Từ đó xây dựng hệ thống cho phép tạo và kiểm tra chữ ký số đối với các tài liệu: công văn, giấy tờ hành chính điện tử để bảo mật nội dung thông tin cũng như xác thực nguồn gốc của thông tin.

* 1. **Các lý thuyết cần tìm hiểu khi làm đồ án**

Đối tượng nghiên cứu bao gồm :

* Cơ sở và nguyên lý xây dựng các hệ mật khóa công khai.
* Mô hình chung về chữ ký số và các lược đồ chữ ký số cụ thể như lược đồ chữ ký RSA.

Phạm vi nghiên cứu bao gồm :

* Hệ mật khóa công khai RSA và các cơ sở toán học liên quan.
* Phương pháp mã hóa và giải mã, phương pháp hình thành và kiểm tra chữ ký số.
  1. **Tính cấp thiết của đề tài**

Hiện nay,với sự bùng nổ của mạng Internet mạng máy tính đang ngày càng đóng vai trò thiết yếu trong mọi lĩnh vực hoạt động của toàn xã hội, Ích lợi của Internet mang lại đối với xã hội là vô cùng, nhưng cũng không thể không kể đến những mặt trái của nó khi con người sử dụng nó với mục đích không tốt. .Vì vậy mà đối với những thông tin quan trọng khi truyền trên mạng như những bản hợp đồng ký kết, các văn kiện mang tính bảo mật, thì vấn đề quan tâm nhất đó là có truyền được an toàn hay không? Do vậy để chống lại sự tấn công hay được nhìn thấy”, hoặc không thể giả mạo văn bản, dù có xâm nhập được vào văn bản. Nhu cầu đó ngày nay đã được đáp ứng khi công nghệ mã hóa và chữ ký số ra đời. Với công nghệ này, thì đã trợ giúp con người giải quyết được bài toán giả mạo, thì nảy sinh yêu cầu là cần phải làm thế nào cho văn bản khi được gửi đi sẽ “không nan giải về bảo mật khi trao đổi thông tin.

Cùng với sự phát triển của mật mã khóa công khai, người ta đã nghiên cứu và đưa ra nhiều phương pháp, nhiều kỹ thuật ký bằng chữ ký số ứng dụng trong các hoạt động kinh tế, xã hội.Chẳng hạn như các ứng dụng trong thương mại điện tử, các giao dịch của các chủ tài khoản trong ngân hàng, các ứng dụng trong chính phủ điện tử đòi hỏi việc xác nhận danh tính phải được đảm bảo.Ngày nay chữ ký số được sử dụng trong nhiều lĩnh vực như trong kinh tế với việc trao đổi các hợp đồng giữa các đối tác kinh doanh, trong xã hội là các cuộc bỏ phiếu kín khi tiến hành bầu cử từ xa, hay trong các cuộc thi phạm vi rộng lớn.

Khi nói đến chữ ký điện tử, chúng ta luôn lấy mục tiêu an toàn lên hàng đầu. Một chữ ký điện tử chỉ thực sự được áp dụng trong thực tế nếu như nó được chứng minh là không thể giả mạo. Xuất phát từ thực tế đó tôi đã chọn đề tài “ Nghiên cứu cài đặt thử nghiệm thuật toán chữ ký số HD 11” với mong muốn nghiên cứu và phát triển về vấn đề an toàn thông tin trong thực tiễn.

* 1. **Phương pháp nghiên cứu**

Nghiên cứu, thu thập các tài liệu đã xuất bản, các bài báo trên các tạp chí khoa học và các tài liệu trên mạng Internet liên quan đến vấn đề đang nghiên cứu của các tác giả trong và ngoài nước. Từ đó chọn lọc và sắp xếp lại theo ý tưởng của mình:

* Tìm hiểu, vận dụng và kế thừa một số các hàm mật mã đã có trên Internet.
* Khai thác hệ thống mã nguồn mở và ngôn ngữ lập trình hướng đối tượng C# để xây dựng một ứng dụng về mã hóa dữ liệu và chữ ký số.
  1. **Nội dung nghiên cứu**

Nội dung nghiên cứu của Đồ án bao gồm :

* Giới thiệu chung về chữ ký số,những công cụ liên quan và cơ sở hình thành nên chữ số.
* Xây dựng một số lược đồ cơ sở dựa trên hệ mật RSA,hệ mật ElGamal.
* Đề xuất và phát triển lược đồ chữ ký số dựa trên các lược đồ cơ sở và thử nghiệm chương trình bằng ngôn ngữ C.

# Chương 2

# TỔNG QUAN VỀ CHỮ KÝ SỐ

* 1. **An toàn thông tin và hướng nghiên cứu đề tài**
     1. ***Các nguy cơ mất an toàn***

Bảo mật và xác thực thông tin luôn là vấn đề cấp thiết được đặt ra cho các cơ quan, xí nghiệp. Đặc biệt là trong nền kinh tế thị trường hiện nay, thông tin luôn đóng vai trò quan trọng trong các chiến lược kinh doanh của đơn vị.

Ngày nay, với sự phát triển mạnh mẽ của Công nghệ thông tin và truyền thông cho phép các cơ quan, xí nghiệp có thể kết nối với Internet thông qua các đường truyền thuê bao tốc độ cao. Tạo ra một môi trường truyền thông thuận lợi cho quá trình trao đổi và chia sẻ thông tin giữa các đơn vị. Tuy nhiên, môi trường này còn tiềm ẩn những vấn đề, gây nên sự mất an toàn cho việc lưu trữ và trao đổi thông tin.

Ngoài những lợi ích to lớn Internet đem lại. Internet còn tạo điều kiện cho các hacker có thể xâm nhập vào hệ thống các máy chủ, để tìm kiếm và đánh cắp thông tin nhằm phục vụ cho lợi ích riêng của mình.

Hơn nữa, khi thực hiện trao đổi qua mạng Internet, người nhận không thể xác thực được nguồn gốc của thông tin, dẫn đến tình trạng thông tin dễ bị giả mạo bởi người khác.

Giả sử sàn giao dịch chứng khoán nhận được email với nội dung “giá dầu thô thế giới có thể sẽ tăng rất mạnh do OPEC cắt giảm khai thác dầu mỏ", điều này sẽ làm thay đổi kế hoạch của người tham gia thị trường chứng khoán, nhưng thật ra thông tin này là giả mạo.

Thanh toán bằng thẻ tín dụng qua dịch vụ web có thể gặp các rủi ro như: Thông tin truyền từ trình duyệt web của khách hang ở dạng thuần văn bản nên có thể bị lọt vào ‘con mắt’ người khác. Trình duyệt web của khách hàng không thể xác định máy chủ mà mình trao đổi thông tin thật hay giả mạo. Không thể đảm bảo được thông tin truyền đi có bị thay đổi hay không.

Vì vậy cần phải có một cơ chế đảm bào an toàn trong quá trình giao dịch điện tử. Một hệ thống thông tin trao đổi dữ liệu an toàn phải đáp ứng các yêu cầu sau:

* Hệ thống phải đảm bảo dữ liệu trong quá trình truyền đi không bị đánh cắp.
* Hệ thống phải có khả năng xác thực, tránh trường hợp giả danh, mạo nhận.
* Hệ thống phải có khả năng kiểm tra tính toàn vẹn dữ liệu.
  + 1. ***Các giải pháp bảo đảm***
       1. Mật mã (Encryption)

*Mật mã (Encryption*) là một kỹ thuật cơ sở quan trọng trong bảo mật thông tin. Nguyên tắc của mật mã là biến đổi thông tin gốc thành dạng thông tin bí mật mà chỉ có những thực thể tham gia xử lý thông tin một cách hợp lệ mới hiểu được. Một thực thể hợp lệ có thể là một người, một máy tính hay một phần mềm nào đó được phép nhận thông tin.Để có thể giải mã được thông tin mật, thực thể đó cần phải biết cách giải mã (tức là biết được thuật tóan giải mã) và các thông tin cộng thêm (khóa bí mật).

Quá trình chuyển thông tin gốc thành thông tin mật theo một thuật toán nào đó được gọi là quá trình mã hoá (encryption). Quá trình biến đổi thông tin mật về dạng thông tin gốc ban đầu gọi là quá trình giải mã (decryption). Đây là hai quá trình không thể tách rời của một kỹ thuật mật mã bởi vì mật mã (giấu thông tin) chỉ có ý nghĩa khi ta có thể giải mã (phục hồi lại) được thông tin đó. Do vậy, khi chỉ dùng thuật ngữ mật mã thì nó có nghĩa bao hàm cả mã hóa và giải mã.

Kỹ thuật mã hoá được chia thành hai loại: mã hoá dùng khoá đối xứng (symmetric key encryption) và mã hoá dùng khoá bất đối xứng (asymmetric key encryption).

* ***Hệ mã bí mật (secret key cryptosystem)***

Hệ mã bí mật (secret key cryptosystem)hay hệ mã đối xứng là hệ mã hóa Kỹ thuật mật mã đối xứng được đặc trưng bởi việc sử dụng một khóa duy nhất cho cả quá trình mã hóa và giải mã thông tin. Bằng một cách an tòan nào đó, khóa chung này phải được trao đổi thống nhất giữa bên gởi và bên nhận (tức bên mã hóa và bên giải mã), đồng thời được giữ bí mật trong suốt thời gian sử dụng.

Kỹ thuật mật mã đối xứng còn được gọi là mật mã quy ước (conventional encryption) hoặc mật mã dùng khóa bí mật (secret key encryption).

Cấu trúc chung của một hệ thống mật mã hóa quy ước như trình bày ở hình 1.1, trong đó, kênh thông tin dùng để trao đổi khóa bí mật phải là một kênh an tòan. Có thể thực hiện việc trao đổi khóa bí mật giữa hai thực thể A và B theo những cách sau đây:

* A chọn ra một khóa bí mật và chuyển trực tiếp cho B (chuyển bằng phương tiện vật lý như ghi lên đĩa, nói trực tiếp, ghi ra giấy, …)
* Một thực thể thứ 3 chọn ra khóa bí mật và thông báo khóa này cho cả A và B (bằng phương tiện vật lý như trên)
* Nếu A và B trước đó đã dùng một khóa nào đó để thông tin với nhau, thì một trong hai thực thể sẽ tiếp tục dùng khóa cũ để gởi thông báo về khóa mới cho thực thể kia.
* Nếu A và B có các kết nối an tòan đến một thựcthể thứ 3 là C, thì C có thể gởi thông báo về khóa cho cả hai thực thể A và B thông qua kết nối an tòan này.

Thuật toán mã hóa

Thuật toán giải mã

Thông tin mật

Thông tin gốc

Thông tin gốc

Khóa bí mật (dùng chung)

**Hình 2.1. Trao đổi khoá trong mật mã đối xứng**

Mã hóa đối xứng dựa chủ yếu trên hai thao tác: thay thế và chuyển vị. Thao tác thay thế sẽ thay từng từ mã bởi một từ mã khác theo một quy ước nào đó, và quy ước này chính là khóa của hệ thống mã. Ví dụ: thay thế từng ký tự trong một thông điệp bằng một ký tự đứng cách nó 3 vị trí trong bảng chữ cái la tinh, thông điệp “HELLO WORLD” sẽ được mã hóa thành “KHOOR ZRUOG”.

Thao tác chuyển vị thực hiện việc thay thế vị trí của các từ mã trong thông tin gốc theo một quy ước nào đó và quy ước này cũng trở thành khóa của hệ thống. Ví dụ: dịch từng ký tự trong một thống điệp qua phải một ví trí có xoay vòng, thông điệp “HELLO WORLD” sẽ được mã hóa thành “DHELLO WORL”.

Sử dụng thuật toán mã hóa đối xứng - giải thuật giải mã ngược với giải thuật tạo bản mã, cả 2 giải thuật dùng chung một khóa (Secret key). Khóa được dùng chung giữa bên gửi và bên nhận nên tồn tại một số điểm yếu :

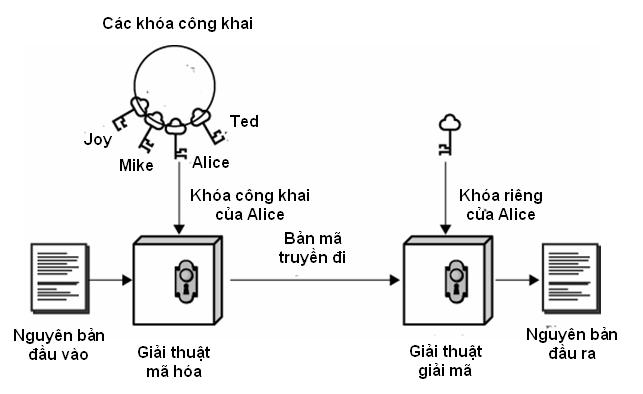
* Vấn đề phân phối khóa khó bảo đảm chia sẻ mà không làm tiết lộ, hoặc trung tâm phân phối khóa có thể bị tấn công.
* Yêu cầu để tạo chữ ký số là phải bí mật chỉ người dùng duy nhất có khóa để tạo chữ ký nên mã hóa đối xứng không được áp dụng cho lĩnh vực chứ ký số.
* ***Hệ mã công khai (public key cryptosystem)***

Hệ mã công khai (public key cryptosystem)hay mã hóa phi đối xứng là hệ mã mà trong đó việc lập mã và giải mã sử dụng 2 chìa khóa riêng biệt, từ chìa khóa này không thể tìm ra chìa khóa kia và ngược lại. Khóa được dùng để mã hóa gọi là khóa công khai, còn khóa giành cho việc giải mã , luôn được giữ bí mật gọi là khóa riêng.

Hệ mật mã khóa công khai được xây dựng dựa trên cơ sở toán học chặt chẽ, được chứng minh về tính đúng đắn của các thuật toán trong sơ đồ của hệ mã. Và đã giải quyết được vấn đề dùng chung khóa trong các hệ mật mã đối xứng.

Giải thuật khóa công khai sử dụng 2 khóa khác nhau :

* Một khóa công khai ai cũng có thể biết dùng để mã hóa thông báo và thẩm tra chữ ký
* Một khóa riêng chỉ nơi giữ được biết dùng để giải mã thông báo và ký chữ ký có tính chất bất đối xứng.
* Bên mã hóa không thể giải mã thông báo (nếu dùng để mã hóa thông báo)
* Bên thẩm tra không thể tạo chữ ký ( nếu dùng để ký )



**Hình 2.2. Mô hình của mật mã khóa công khai**

* + - 1. Chữ ký số

*Chữ ký số(Digital signature)* là một dạng dữ liệu số được sinh ra bởi một lược đồ chữ ký số, có chức năng liên kết một thông điệp dữ liệu (bản tin,thông báo, tài liệu,...) với thực thể (con người, thiết bị kỹ thuật,...) tạo ra nó, nhằm đáp ứng các yêu cầu xác thực về nguồn gốc và tính toàn vẹn của thông điệp dữ liệu. Về nguyên tắc của chữ ký điện tử cũng gần như chữ ký thông thường, ví dụ như nếu người A muốn gửi thông điệp cho người B, thì A sẽ gửi chữ ký cùng với thông điệp của mình cho B. Khi nhận được thông điệp và chữ ký, bằng cách nào đó để B có thể xác định chữ ký kèm theo thông điệp có phải của người A hay không. Điểm khác biệt giữa chữ ký điện tử và chữ ký thông thường là: chữ ký thông thường thì nằm bên trong thông điệp, và chữ ký của một người là luôn giống nhau ở mọi thông điệp, còn chữ ký điện tử thì được gửi kèm với thông điệp nhưng tách biệt, và chữ ký của một người cho các thông điệp khác nhau là hoàn toàn khác nhau.

[So sánh chữ ký số với chữ ký thông thường(chữ ký viết tay) trên văn bản](#_Toc262644819). Chữ ký số và chữ ký thường có nhiều điểm khác nhau :

|  |  |
| --- | --- |
| Chữ ký thông thường | Chữ ký điện tử |
| Vấn đề ký một tài liệu  Chữ ký là một phần vật lý của tài liệu | Vấn đề ký một tài liệu  Chữ ký số không gắn kiểu vật lý vào bức thông điệp nên thuật toán được dùng phải “không nhìn thấy” theo một cách nào đó trên bức thông điệp |
| Vấn đề về kiểm tra  Chữ ký kiểm tra bằng cách so sánh nó với chữ ký xác thực khác. Tuy nhiên, đây không phải là một phương pháp an toàn vì nó dễ bị giả mạo. | Vấn đề về kiểm tra  Chữ ký điện tử có thể kiểm tra nhờ dùng một số thuật toán “kiểm tra công khai”. Như vậy, bất kì ai cũng có thể kiểm tra được chữ ký điện tử. Việc dùng chữ ký điện tử an toàn có thể chặn được giả mạo. |
| Bản copy thông điệp được ký bằng chữ ký thông thường lại có thể khác với bản gốc. | Bản copy thông điệp được ký bằng chữ ký số thì đồng nhất với bản gốc, điều này có nghĩa là cần phải ngăn chặn một bức thông điệp ký số không bị dùng lại. |

* + 1. ***Hướng nghiên cứu của đề tài***

Trong giao dịch điện tử nói chung và thương mại điện tử nói riêng, quá trình trao đổi thông tin tương tác giữa các thành viên, đòi hỏi phải có một cơ chế hay hệ thống xác định nguồn gốc chủ sở hữu của thông tin. Giống như trong lĩnh vực tài liệu thông thường, nếu như chữ ký viết tay là để chứng minh tác giả hay người công nhận nội dung của tài liệu, thì lĩnh vực tài liệu điện tử cũng có một tiêu chuẩn như vậy về “chữ ký”, gọi là chữ ký điện tử (digital signature).

Hiện nay tại Việt Nam, các giao dịch thương mại điện tử đang ngày càng trở nên phổ biến. Để đảm bảo an toàn cho các giao dịch này, giải pháp sử dụng chữ ký số đã được đề xuất sử dụng.

Chữ ký số sẽ giúp cho các giao dịch giữa công dân, doanh nghiệp và nhà nước thuận tiện và đảm bảo hơn. Ví dụ: người dân có thể kê khai, nộp thuế và chuyển tiền trực tiếp qua mạng, doanh nghiệp có thể xây dựng hệ thống mua bán trực tuyến, đảm bảo việc thanh toán qua hệ thống với chứng thư đã được xác nhận, các daonh nghiệp ở các địa phương cũng có thể ký kết hợp đồng qua mạng thay vì phải gặp nhau trực tiếp như hiện nay... Hiện đã có nhiều bộ, ngành triển khai các dịch vụ công trực tuyến có sử dụng chữ ký số: Tổng cục Thuế, Ngân hàng Nhà nước, Bộ Kế hoạch và Đầu tư...

* 1. **Chữ ký số**
     1. ***Tổng quan về CKS***
        1. Khái niệm

Chữ ký điện tử là một đoạn dữ liệu ngắn đính kèm với văn bản gốc để chứng thực tác giả của văn bản và giúp người nhận kiểm tra tính toàn vẹn của nội dung văn bản gốc. Chữ ký số là một dạng đặc biệt của chữ ký điện tử an toàn nhất và cũng được sử dụng rộng rãi nhất. Chữ ký này được tạo nên dựa trên kỹ thuật sử dụng mã khoá công khai (PKI), theo đó mỗi người sử dụng cần có một cặp khóa bao gồm khóa bí mật và công khai. Khóa bí mật dùng để tạo chữ ký số, khóa công khai dùng để thẩm định,xác định chữ ký số.



**Hình 2.3. Quy trình tạo và kiểm tra chữ ký số**



**Hình 2.4. Quy trình thẩm định chữ ký số**

* + - 1. Các thuật ngữ liên quan

Trong [mật mã học](http://vi.wikipedia.org/wiki/M%E1%BA%ADt_m%C3%A3_h%E1%BB%8Dc), một ngành [toán học ứng dụng](http://vi.wikipedia.org/wiki/To%C3%A1n_h%E1%BB%8Dc_%E1%BB%A9ng_d%E1%BB%A5ng) cho [công nghệ thông tin](http://vi.wikipedia.org/wiki/C%C3%B4ng_ngh%E1%BB%87_th%C3%B4ng_tin), mã hóa là phương pháp để biến thông tin ([phim ảnh](http://vi.wikipedia.org/wiki/Phim_%E1%BA%A3nh), [văn bản](http://vi.wikipedia.org/w/index.php?title=V%C4%83n_b%E1%BA%A3n&action=edit&redlink=1), [hình ảnh](http://vi.wikipedia.org/w/index.php?title=H%C3%ACnh_%E1%BA%A3nh&action=edit&redlink=1)...) từ định dạng bình thường sang dạng thông tin không thể hiểu được nếu không có phương tiện giải mã.Văn bản là một thông báo gốc cần chuyển có định dạng là văn bản, âm thanh, hình ảnh, chữ số….

* **Bản rõ** (plaintext or cleartext) : Chứa các xâu ký tự gốc, thông tin trong bản rõ là thông tin cần mã hoá để giữ bí mật.
* **Bản mã** (ciphertext) : Chứa các ký tự sau khi đã được mã hoá, mà nội dung được giữ bí mật.
* **Mật mã học** (Crytography) : Là nghệ thuật và khoa học để giữ thông tin được an toàn.
* **Sự mã hoá** (Encryption) : Quá trình che dấu thông tin bằng phương pháp nào đó để làm ẩn nội dung bên trong gọi là sự mã hoá.
* **Sự giải mã** (Decryption) : Quá trình biến đổi trả lại bản mã bản thành bản rõ gọi là giải mã.
* Khóa là bí quyết lập mã và giải mã. Nếu như việc mã hóa được xem như một hàm y = f(x,k), trong đó x là văn bản đầu vào, còn k là một tham số điều khiển, f là phương pháp mã hóa. Trước đây bí quyết thường là cả f và k. Do yêu cầu hiện nay công nghệ mã hóa đã phải thay đổi quan điểm này. Phương pháp f thường không do một người nắm giữ nên không thể giữ bí mật nên phải coi nó là công khai. Tham số điều khiển k, có tác dụng làm thay đổi kết quả và được coi là chìa khóa mã. Thông thường nó là một xâu bit mà người sử dụng có thể giữ riêng cho mình.

Nguyên tắc chung của mã hóa là việc giải mã phải rất dễ dàng với người trong hệ thống sử dụng, và ngược lại rất khó giải mã (thậm chí không thực hiện được) đối với người ngoài.

Quá trình mã hoá và giải mã được thể hiện trong sơ đồ sau :

**Bản rõ**

**Bản mã**

**Mã hóa**

**Hình 2.5. Quá trình mã hóa C =**

**Bản mã**

**Bản rõ**

**Giải mã**

**Hình 2.6. Quá trình giải mã M =**

* Hệ mật mã : là một hệ bao gồm 5 thành phần (M, C, K, E, D) thoả mãn các tính chất sau :

M (Plaintext) là tập hợp hữu hạn các bản rõ có thể.

C (Ciphertext) là tập hợp hữu hạn các bản mã có thể.

K (Key) là tập hợp các bản khoá có thể.

E (Encrytion) là tập hợp các qui tắc mã hoá có thể.

D (Decrytion) là tập hợp các qui tắc giải mã có thể.

Chúng ta đã biết một thông báo thường được tổ chức dưới dạng bản rõ. Người gửi sẽ làm nhiệm vụ mã hoá bản rõ, kết quả thu được gọi là bản mã. Bản mã này được gửi đi trên một đường truyền tới người nhận sau khi nhận được bản mã người nhận giải mã nó để tìm hiểu nội dung. Dễ dàng thấy được công việc trên khi sử dụng định nghĩa hệ mật mã :

và

* + 1. ***Một số hệ chữ ký số điển hình***
       1. Hệ mã hóa RSA

Hệ mật mã khóa công khai RSA là hệ thống mật mã do các giáo sư Ronald Rivest, Adi Sharmir và Leonard Adleman phát mình năm 1978 tại học viện Công nghệ Massachusetts (MIT).

Thuật toán RSA có hai [khóa](http://vi.wikipedia.org/wiki/Kh%C3%B3a_%28m%E1%BA%ADt_m%C3%A3%29): [khóa công khai](http://vi.wikipedia.org/wiki/Kh%C3%B3a_c%C3%B4ng_khai) (hay khóa công cộng) và [khóa bí mật](http://vi.wikipedia.org/w/index.php?title=Kh%C3%B3a_b%C3%AD_m%E1%BA%ADt&action=edit&redlink=1) (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](http://vi.wikipedia.org/wiki/M%C3%A3_h%C3%B3a). 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. Sơ đồ RSA là sơ đồ mật mã khối, trong đó cả văn bản rõ và văn bản mã là các số nguyên trong dải từ 0 đến n – 1 đối với một giá trị n nào đó.

Hệ mã RSA được xây dựng trên cơ sở mã hóa khối trong đó khóa mã hóa là cặp(e,n) gồm số mũ e và module n. Với n là tích số của 2 số nguyên tố rất lớn nào đó, n = p\*q còn (e, (n)) = 1, với (n) là giá trị hàm Euler của n, trong trường hợp này (n) = (p - 1)\*(q - 1).

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

* ***Thuật toán sinh khóa***

Để sử dụng được hệ mật mã khóa công khai RSA ,trước tiên mỗi người phải tạo riêng cho mình một cặp khóa gồm khóa công khai, và khóa bí mật. Việc tạo ra khóa công khai và khóa bí mật thực hiện theo các bước sau:

1. Sinh ra 2 số nguyên tố lớn p và q ngẫu nhiên (p q).

2. Tính n = p\*q.

3. Tính: giá trị hàm số Ơle(n) = (p - 1)\*(q - 1).

4. Chọn một số tự nhiên e sao cho 1 < e <(n) và là số nguyên tố cùng nhau với (n).

5. Tính d sao cho d\*e ≡ 1 (mod (n)) với 1 < d <(n).

* Khóa công khai (e,n), khóa bí mật (d,n).
* ***Thuật toán mã hóa***

Hệ RSA là một hệ mật mã điển hình về kiểu mã hóa khối.Nghĩa là, thông điệp được chia thành nhiều khối (hoặc chuỗi) có chiều dài cố định, và mỗi khối sẽ được mã hóa riêng. Giả sử để gửi thông điệp bí mật M cho người nhận B trong nhóm gửi thông tin an toàn, người gửi A phải thực hiện các bước như sau:

* Thu nhận khóa công khai (e,n) của người nhận B.
* Thực hiện một thuật toán để biến đổi thông điệp M thành những số nguyên mi tương ứng sao cho mi < n, (i = 1,…, k). A sẽ tính C là bản mã hóa của M theo công thức: C = (mod n)
* ***Thuật toán giải mã***

Để thực hiện quá trình giải mã, khôi phục lại nội dung của thông điệp M từ bản mã C nhận được, người nhận B sẽ thực hiện các bước như sau:

* Tính M =(mod n) với 0 ≤ mi ≤ n.
* Thực hiện phép biến đổi ngược từ các số mi thành chuỗi ký tự tương ứng chứa thông tin M ban đầu.
  + - 1. Hệ El Gamal
* ***Phương pháp hình thành khóa***

Thuật toán hình thành khóa bao gồm các bước sau:

1. Chọn số nguyên tố đủ lớn p sao cho bào toán logarit trong là khó giải.
2. Chọn g là phần tử sinh của nhóm .
3. Chọn khóa mật *x* là số ngẫu nhiên sao cho: 1 <*x <p* .Tính khóa công khai (y) theo công thức:

y = ;

* ***Phương pháp mã hóa và giải mã***
* **Thuật toán mã hóa**

Giả sử đối tượng gửi là A, đối tượng nhận là B. Đối tượng nhận B có khóa bí mật là và khóa công khai là . Khi đó, để gửi thông điệp dữ liệu M cần bảo mật nội dung cho B, với: 0 , đối tượng gửi A sẽ thực hiện các bước như sau:

1. Chọn số ngẫu nhiên *k* thỏa mãn: 1 <*k <p .*
2. Tính giá trị *R* theo công thức:

R = ;

1. Sử dụng khóa công khai của B để tính:

C = M × ;

1. Gửi bản mã gồm (C,R) đến đối tượng nhận B.

* **Thuật toán giải mã**

Để khôi phục thông điệp dữ liệu ban đầu (M) từ bản mã (C,R) nhận được, đối tượng nhận B thực hiện các bước nhưsau:

1. Tính giá trị Z theo công thức:

Z = ;

1. Tính giá trị nghịch đảo của Z:
2. Khôi phục bản tin ban đầu (M):

* ***Mức độ an toàn của hệ mật ElGamal***

Mục đích của việc tấn công vào hệ mật ElGamal là khôi phục thông điệp

dữ liệu M từ bản mã (C,R) và các tham số công khai của hệ thống: p, q, g.

* Hệ mật mã ElGamal sẽ bị phá vỡ nếu khóa mật x hoặc k có thể tính được.
* Để tính được x hoặc k, kẻ tấn công cần phải giải một trong hai bài toán logarit rời rạc: y = hoặc: R = . Tuy nhiên, việc giải bài toán logarit rời rạc này là một việc khó .
* Để khôi phục được nội dung thông điệp dữ liệu M kẻ tấn công cần phải giải đượctrong trường hợp không biết k, hoặctrong trường hợp không biết . Có thể thấy rằng, việc khôi phục M từ việc giải trực tiếp các bài toán này cũng khó như việc giải bài toán logarit rời rạc để tìm x hoặc k đã được chỉ ra ở trường hợp trên .
* Một điểm yếu có thể bị tấn công trong hệ mật mã ElGamal là khi giá trị *k* bị sử dụng lại. Thực vậy, giả sử cùng một giá trị k được sử dụng để mã hóa hai thông điệp dữ liệu M và được các bản mã tương ứng là (C,R) và (). Khi ấy ta có :

Suy ra:

Nghĩa là, chỉ cần biết nội dung của một trong hai thông điệp dữ liệu M hoặc thì kẻ tấn công sẽ dễ dàng biết được nội dung của thông điệp dữ liệu kia.

# Chương 3

# CƠ SỞ TOÁN HỌC

* 1. **Đại số trừu tượng**

Đại số trừu tượng là một ngành [toán học](http://vi.wikipedia.org/wiki/To%C3%A1n_h%E1%BB%8Dc) liên quan đến việc nghiên cứu các [cấu trúc đại số](http://vi.wikipedia.org/w/index.php?title=C%E1%BA%A5u_tr%C3%BAc_%C4%91%E1%BA%A1i_s%E1%BB%91&action=edit&redlink=1) như [nhóm](http://vi.wikipedia.org/wiki/Nh%C3%B3m_(to%C3%A1n_h%E1%BB%8Dc)), [vành (toán học)](http://vi.wikipedia.org/wiki/V%C3%A0nh_(to%C3%A1n_h%E1%BB%8Dc)), [trường](http://vi.wikipedia.org/wiki/Tr%C6%B0%E1%BB%9Dng_(to%C3%A1n_h%E1%BB%8Dc)), hay các cấu trúc tổng quát khác.

* + 1. ***Nhóm (Đại số)***

Trong [đại số trừu tượng](http://vi.wikipedia.org/wiki/%C4%90%E1%BA%A1i_s%E1%BB%91_tr%E1%BB%ABu_t%C6%B0%E1%BB%A3ng), một nhóm (G,\*) là một [tập hợp](http://vi.wikipedia.org/wiki/T%E1%BA%ADp_h%E1%BB%A3p) G, cùng với một [phép toán hai ngôi](http://vi.wikipedia.org/wiki/Ph%C3%A9p_to%C3%A1n_hai_ng%C3%B4i), ký hiệu " \* ", từ G×G vào G thỏa mãn các [tiên đề](http://vi.wikipedia.org/wiki/Ti%C3%AAn_%C4%91%E1%BB%81) sau:

G1. [Tính kết hợp](http://vi.wikipedia.org/wiki/Ph%C3%A9p_to%C3%A1n_hai_ng%C3%B4i): phép toán "\*" có tính kết hợp, nghĩa là(a\*b)\*c = a\*(b\*c) với mọi a, b và c thuộc G.

G2. [Phần tử trung hòa](http://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_t%E1%BB%AD_%C4%91%C6%A1n_v%E1%BB%8B):Trong G tồn tại một phần tử được gọi là phần tử trung hòa θ sao cho với mọi phần tử a thuộc G thì : a\*θ = θ\*a = a.

G3. [Phần tử đối lập](http://vi.wikipedia.org/wiki/Ph%C3%A9p_to%C3%A1n_hai_ng%C3%B4i): với mỗi phần tử a thuộc G tồn tại một phần tử x, gọi là phần tử đối lập của a, sao cho: x\*a = a\*x = θ.

Lý thuyết toán học phát triển cho các nhóm gọi là [lý thuyết nhóm](http://vi.wikipedia.org/wiki/L%C3%BD_thuy%E1%BA%BFt_nh%C3%B3m).Lý thuyết này có nhiều ứng dụng vì nhiều thực thể toán học đã gặp trong khoa học thỏa mãn điều kiện trở thành nhóm.Nhóm đại số cũng giúp nghiên cứu về sự [đối xứng](http://vi.wikipedia.org/wiki/%C4%90%E1%BB%91i_x%E1%BB%A9ng), một tính chất thường gặp trong tự nhiên và [vật lý học](http://vi.wikipedia.org/wiki/V%E1%BA%ADt_l%C3%BD_h%E1%BB%8Dc).

Trong định nghĩa của nhóm phép "\*" không đòi hỏi có tính chất [giao hoán](http://vi.wikipedia.org/wiki/Ph%C3%A9p_to%C3%A1n_hai_ng%C3%B4i) (a\*b=b\*a) nếu G thỏa mãn thêm tính chất này thì G được gọi là nhóm giao hoán, hay [nhóm Abel](http://vi.wikipedia.org/wiki/Nh%C3%B3m_giao_ho%C3%A1n). Nếu G không có tính giao hoán thì G được gọi là phi giao hoán hay không Abel.

Lưu ý: Ký hiệu "\*" là ký hiệu tống quát cho các phép toán, hai ký hiệu thường gặp nhất là "+" (cộng) và "x" hoặc đơn giản là "." (nhân).

* + 1. ***Vành***

Tập hợp R được gọi là vành nếu trên đó có hai [phép toán hai ngôi](http://vi.wikipedia.org/wiki/Ph%C3%A9p_to%C3%A1n_hai_ng%C3%B4i) mà ta ký hiệu là "+" (phép cộng) và "." (phép nhân) thỏa mãn các điều kiện sau:

R là một [nhóm giao hoán](http://vi.wikipedia.org/wiki/Nh%C3%B3m_giao_ho%C3%A1n) đối với phép cộng, nghĩa là:

Phép cộng có tính kết hợp:

:

Phép cộng có phần tử trung hòa, nghĩa là:

;

Mọi phần tử của R có phần tử đối:

Phép cộng có tính giao hoán, nghĩa là :

Phép nhân có tính phân phối với phép cộng, nghĩa là:

Phép nhân có tính kết hợp, nghĩa là:

Phép nhân có phần tử đơn vị, nghĩa là:

Một số loại vành đặc biệt:

* Vành giao hoán là vành R trong đó phép nhân có tính chất giao hoán.
* Vành trong đó phép nhân có phần tử đơn vị được gọi là vành có đơnvị.
* Nếu trong vành R tồn tại hai phần tử sao cho a.b=0 thì các phần tử a, b được gọi là ước của 0. Vành giao hoán, có đơn vị, không có ước của 0 được gọi là vành nguyên hay miền nguyên.
* Miền nguyên X gọi là [vành chính](http://vi.wikipedia.org/w/index.php?title=V%C3%A0nh_ch%C3%ADnh&action=edit&redlink=1) nếu mọi ideal của nó đều là được sinh từ một phần tử.
* Miền nguyên A gọi là [vành Ơclit](http://vi.wikipedia.org/w/index.php?title=V%C3%A0nh_%C6%A0clit&action=edit&redlink=1) nếu có ánh xạ f:Ā→N(vớiĀ là tập các phần tử khác 0 của A) thỏa mãn tính chất sau: nếu blà ước của a và a ≠ 0 thì f(b)≤ f(a). Với a,b là hai phần tử tùy ý của A và b≠0 thì tồn tại duy nhất cặp phần tử q,r của A sao cho a=b×q+r và f(b)≥f(r)nếu r≠0.
* Vành Noether :Vành giao hoán có đơn vị được gọi là vành Noether nếu mọi ideal của nó đều là hữu hạn sinh,tức là tồn tại một tập sinh hữu hạn phần tử.
* Vành Gauss hay vành nhân tử hóa là một miền nguyên A mà mọi phần tử khác không và không khả nghịch đèu được phân tích một cách duy nhất thành tích của hữu hạn phần tử bất khả quy nếu không tính đến thứ tự của các phần tử.
  + 1. ***Trường***

Trường (đại số) là một [tập](http://vi.wikipedia.org/wiki/T%E1%BA%ADp) ***F***  trên đó có hai [phép toán cộng](http://vi.wikipedia.org/w/index.php?title=Ph%C3%A9p_to%C3%A1n_c%E1%BB%99ng&action=edit&redlink=1) và [nhân](http://vi.wikipedia.org/wiki/Nh%C3%A2n) thỏa mãn:

1. ***F*** là [nhóm giao hoán](http://vi.wikipedia.org/wiki/Nh%C3%B3m_giao_ho%C3%A1n) với phép cộng
2. ***F*** là [nhóm giao hoán](http://vi.wikipedia.org/wiki/Nh%C3%B3m_giao_ho%C3%A1n) với phép nhân
3. Trên ***F***, phép nhân [phân phối](http://vi.wikipedia.org/w/index.php?title=Ph%C3%A2n_ph%E1%BB%91i&action=edit&redlink=1) với phép cộng

Chi tiết hơn các điều kiện trên, ta có thể kể ra các [tiên đề](http://vi.wikipedia.org/wiki/Ti%C3%AAn_%C4%91%E1%BB%81) của trường như sau:

Trường là một tập hợp  trên đó xác định hai phép toán cộng và nhân:

Phép cộng (+) :

Phép nhân (\*)  :

thoả mãn các tiên đề sau:

Phép cộng có tính kết hợp:

Phép cộng có tính giao hoán:

Tồn tại phần tử 0:

Tồn tại phần tử đối:

Phép nhân có tính kết hợp:

Phép nhân có tính giao hoán:

Tồn tại phần tử đơn vị:

Tồn tại phần tử nghịch :

Phép nhân phân phối với phép cộng:

* 1. **Lý thuyết số**
     1. ***Phép modulo***

Với n là kết quả của phép nhân hai số nguyên tố lớn được sinh ngẫu nhiên. Cho m và c là những số nguyên nằm trong khoảng (0,n-1) và e là một số nguyên lẻ trong khoảng (3,n-1) và nguyên tố cùng nhau với p-1 và q-1.

Thao tác mã hóa và giải mã trong hệ mã hóa khóa công khai RSA được thực hiện:

Phép tính mũ hóa modul : Cho n,m và e. Việc tính *c = mod n* là dễ dàng.

Giá trị mod n chính thức là kết quả của nâng lũy thừa e của m, chia cho n và lấy phần dư. Điều này có thể là một phép tính toán phức tạp liên quan tới việc nhân (e-1) số m và kết quả trả về là một số nguyên lớn, trước khi việc thực hiện phép chia cho n. Tuy nhiên hai cách tối ưu hóa sau làm cho việc tính toán trở nên dễ dàng :

1. Nhân với một trình tự thích hợp của các giá trị trung gian trước đó, thay vì hơn chỉ bằng m, có thể giảm số lượng các phép nhân để không quá hai lần kích thước của e trong hệ nhị phân.
2. Chia và lấy phần dư sau khi mỗi phép nhân giữ kết quả trung gian có cùng kích thước như n.
3. Phép khai căn module – nghịch đảo của phép lũy thừa module.

Cho n,e,c và những thừa số nguyên tố p, q, việc khôi phục lại giái trị m sao cho c = me mod n là dễ dàng.

Giá trị m có thể khôi phục từ c bởi thao tác mũ hóa modul với một số nguyên lẻ d nằm trong khoảng (3,n-1). Đặc biệt, với số d này, biểu thức sau thể hiện cho tất cả m : m = (me)d mod n.

Số nguyên d này thì dễ dàng tính với e, p, q cho trước.

Phép khai căn modul lại khó ở một hoàn cảnh khác: Cho n,e, và nhưng không biết những thừa số nguyên tố, việc khôi phục lại m là khó khăn.

Phương pháp nhanh nhất thì có sẵn trong việc tính toán khai căn modul dưới điều kiện dựa là n và e là phân tích thừa số n và áp dụng hệ quả 5 để quyết định d. Thực sự, bất kỳ phương thức nào quyết định d đều bị chuyển về một cách khác của việc phân tích thừa số n. Đúng là có thể khi mà tồn tại một phương pháp mà tính toán khai căn modul mà không cần phân tích n hoặc quyết định d. Nhưng cho đến nay chưa phương phàp nào có thể làm như vậy nhanh hơn việc phân tích thừa số n.

Nhận xét : Số học, đặc biệt là số nguyên lớn và các phép tính đồng dư là những công cụ quan trọng trong mật mã học đặc biệt là trong việc tính toán mật mã học khóa công khai, điển hình là RSA. Tuy nhiên chương này cũng chỉ trình bày qua các thuật toán để làm việc với những số nguyên lớn mà hầu hết đều đã được cài đặt thành thư viện nên ở những hệ thống thực tế người ta sẽ sử dụng chúng để tiện cho quá trình cài đặt.

* + 1. ***Số nguyên tố***

***Định nghĩa:*** số nguyên  được gọi là số nguyên tố nếu nó chỉ có các ước số dương là 1 và p. Ngược lại, p được gọi là hợp số.

Ta có một số kết luận về các số nguyên tố như sau:

1. Nếu p là số nguyên tố và p|ab thì hoặc p|a hoặc p|b (hoặc cả hai)
2. Có vô hạn số nguyên tố.

***Định lý:*** Mỗi số nguyên  được phân tích thành tích các luỹ thừa của các số nguyên tố.

 với  là các số nguyên tố khác nhau, còn  là các số nguyên dương. Hơn nữa, phân tích là duy nhất không kể tới thứ tự của các nhân tử.

Số nguyên tố là một số lớn hơn 1, nhưng chỉ chia hết cho 1 và chính nó, ngoài ra không còn số nào nó có thể chia hết nữa. Hệ mật mã thường sử dụng số nguyên tố lớn cỡ 512 bits và thậm chí lớn hơn như vậy.

Hai số gọi là cặp số nguyên tố khi mà chúng không có thừa số chung nào khác 1, hay nói một cách khác, nếu ước số chung lớn nhất của a và n là bằng 1.Chúng ta có thể viết như sau : gcd(a,n)=1

Số 15 và 28 là một cặp số nguyên tố, nhưng 15 và 27 thì không phải cặp số nguyên tố do có ước số chung là 1 và 3, dễ dàng thấy 13 và 500 cũng là một cặp số nguyên tố. Một số nguyên tố là một cặp số nguyên tố với tất cả những số khác loại trừ những số là bội số.Một cách dễ nhất để tính toán ra ước số chung lớn nhất của hai số là nhờ vào thuật toán Euclid. Knuth mô tả thuật toán và một vài mô hình của thuật toán đã được sửa đổi.

* + 1. ***Hàm Euler***

***Định nghĩa :*** Với , giả sử  ký hiệu số các số nguyên trong khoảng  nguyên tố cùng nhau với n. Hàm số  được gọi là hàm số Euler.

Các tính chất của hàm Euler :

1. Nếu p nguyên tố thì
2. Hàm Euler có tính nhân. Tức là, nếu *gcd(m,n)=1* thì
   1. **Cơ sở xây dựng các hệ mật khóa công khai và chữ ký số**
      1. ***Bài toán phân tích số***

* *Bài toán:*

p,q : Số nguyên tố lớn.

Từ *p* và *q* dễ dàng tính được :

Từ *n* rất khó tìm được *p* và *q* !

* *Ứng dụng trong thuật toán hình toán khóa:*

p,q : Số nguyên tố lớn.

Chọn khóa công khai là *e* thỏa mãn:

Tính khóa bí mật *d* theo:

Giữ bí mật: p,q, ; công khai: e,n rất khó tính được *d*.

* + 1. ***Bài toán logarit rời rạc***
* *Bài toán:*

Cho p là số nguyên tố lớn.

Cho: tính

Từ rất khó tính được *a.*

* *Ứng dụng trong thuật toán mã hóa/giải mã:*

Chọn p là nguyên tố lớn và g là phần tử sinh của nhóm

Chọn khóa bí mật

Tính khóa công khai :

Công khai : p,g,y rất khó tính được *x*.

* + 1. ***Bài toán khai căn***
* *Bài toán:*

Cho p và q là hai số nguyên tố lớn và

Cho : tính

Từ *e,n* và *b* rất khó tìm được *a.*

* *Ứng dụng trong thuật toán hình thành chữ ký số:*

p,q : Số nguyên tố lớn.

Chọn e thỏa mãn: và

Cho Tính :

Giữ bí mật: ; công khai: e,n và C => rất khó tính được M.

* *Bài toán:*

Cho p,q : số nguyên tố lớn thỏa mãn:

Cho : tính

Từ *p,y* và *t* rất khó tìm được *x.*

* *Ứng dụng trong thuật toán kiểm tra chữ ký số:*

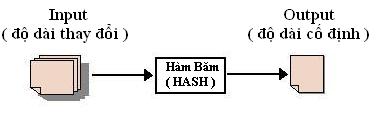
*p,q :* Số nguyên tố lớn:

Chọn *t* thỏa mãn :

Khóa bí mật : *x,* công khai: *y,p* và *t =>* rất khó tính được *x;*

* + 1. ***Hàm băm mật mã***

Hàm băm (hash),là một thuật toán dùng để biến một thông điệp **M** ở đầu vào (input) có chiều dài thay đổi bất kỳ, thành một giá trị **h** ở đầu ra (output) có chiều dài cố định, **h** được gọi là giá trị **hash**. Hàm băm được sử dụng trong rất nhiều lĩnh vực tính toán, mật mã là một trong số đó.



**Hình 3.1: Sơ đồ minh họa hàm băm (HASH)**

Hai ứng dụng phổ biến nhất của hàm hash trong lĩnh vực mật mã là:

* Nén thông điệp thành một khối nhỏ có chiều dài xác định, phục vụ cho các lược đồ chữ ký điện tử, khối dữ liệu nhỏ này gọi là thông điệp thu gọn (Message Digest). Ví dụ như chữ ký điện tử DSA (Digital Signature Algorithm) dùng hàm băm SHA-1 để tạo thông điệp thu gọn dài 160-bits.
* Kiểm tra tính toàn vẹn dữ liệu (Data Integrity), nghĩa là kiểm tra xem dữ liệu có bị thay đổi trên đường truyền hay không, bằng cách tạo mã chứng thực thông điệp MAC (Message Authentication Code).
  + - 1. Yêu cầu của một hàm băm

Hàm hash dùng trong lĩnh vực mật mã phải thỏa các tiêu chuẩn sau:

* Thông điệp (Message) ở đầu vào có chiều dài bất kỳ.
* Thông điệp thu gọn (Message digest) đầu ra có chiều dài cố định (đủ nhỏ).
* Hàm băm H(x) dễ dàng tính toán cho mọi thông điệp x
* Hàm băm H(x) là hàm một chiều (one-way-function): cho trước một giá trị hash **h** thì khó tính toán để tìm ra thông điệp ở đầu vào x sao cho H(x) = h.
* Đụng độ (collision-free): hàm băm H(x) có hai cấp đụng độ là:

1. Đụng độ cấp độ yếu (Weakly collision-free): cho trước thông điệp **x**, không thể tính toán tìm ra một thông điệp **y** khác **x** mà H(x) = H(y).
2. Đụng độ cấp độ mạnh (Strongly collision-free): không thể tính toán để tìm ra hai thông điệp bất kỳ **x** và **y** khác nhau, mà có cùng giá trị **hash**, nghĩa là H(x) = H(y).

Trong phạm vi đề tài này sẽ trình bày chi tiết hai thuật toán băm được ứng dụng phổ biến và có độ an toàn mạnh nhất hiện nay đó là: thuật toán băm MD5 (Message Digest 5), và thuật toán băm SHA (Secure Hash Algorithm).

* + - 1. Hàm băm MD5

Thuật toán băm MD5(Message Digest 5) được thiết kế bởi Ronald Rivest vào năm 1991, thuật toán này là sự nâng cấp mở rộng từ thuật toán băm MD4, nhằm mục đích nâng cao độ an toàn và hiệu quả thực hiện.

* **Mô tả các bước thực hiện thuật toán băm MD5**

Thuật toán băm MD5 có chức năng biến đổi thông điệp ở đầu vào có chiều dài bất kỳ, thành một thông điệp thu gọn (message digest) ở đầu ra có chiều dài 128-bits cố định. Thông điệp được xử lý theo từng khối (block) 512-bits, thực hiện qua bốn vòng (round), và chạy rất tốt trên các bộ xử lý có cấu trúc 32- bits.

**Bước 1:** *(chèn thêm bits)*

Ở bước này, thông điệp có chiều dài k-bits ban đầu sẽ được chèn thêm vào một số bits bằng cách thực hiện như sau: đầu tiên chèn bit có giá trị 1 vào cuối thông điệp, tiếp theo chèn các bits có giá trị 0 mãi cho đến khi chiều dài mới k1 của thông điệp thỏa mãn k1 448 (mod 512). Bước này luôn được thực hiện, ngay cả khi chiều dài k-bits ban đầu của thông điệp đã đồng dư với 448 modulo 512.

**Bước 2:** *(chèn số k để có tổng chiều dài bits là bội của 512)*

Số k là chiều dài thông điệp gốc, được biểu diễn thành hai word 32-bits theo dạng little-endian (word có thứ tự thấp được chèn trước) và chèn lần lược vào kết quả của bước 1. Trường hợp chiều dài k > 264 thì chỉ có 64-bits thấp được chèn vào.

Sau hai bước thực hiện, thông điệp mới có tổng chiều dài: = k1 + 64 là bội số của 512. Chia thông điệp thành nhiều khối có kích cỡ 512-bits, tương đương với 16 words (32-bits), để phục vụ cho bước xử lý tiếp theo.

**Bước 3:** *( khởi động bộ đệm MD)*

Sử dụng bộ đệm MD (buffer) có kích cỡ 128-bits dùng để chứa kết quả tính toán trung gian và cuối cùng của hàm băm. Có thể xem bộ đệm này như là 4 thanh ghi 32-bits (A, B, C, D). Những thanh ghi này được khởi tạo đầu tiên bằng các giá trị hex và theo dạng little-endian (byte có thứ tự thấp của một số thì được lưu trong bộ nhớ tại địa chỉ thấp) như sau:

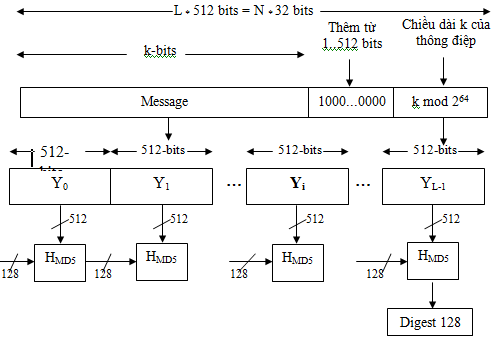
Word A : 01 23 45 67

Word B : 89 AB CD EF

Word C : FE DC BA 98

Word D : 76 54 32 10

* **Các bước thực hiện của thuật toán MD5 có thể được minh họa như sau**:



**Hình 3.2 Sơ đồ thực hiện tổng quát của thuật toán MD5**

**Bước 4:** *( Xử lý từng khối 16 words của thông điệp )*

Các hàm tính toán của các vòng xử lý:

F(X,Y,Z) = (X Y) V (( X) Z)

G(X,Y,Z) = (X Z) V (Y (Z))

H(X,Y,Z) = X Y Z

I(X,Y,Z) = Y (X (Z))

* Các phép toán logic được sử dụng trong các hàm:

X Y bitwise AND

X VY bitwise OR

X Y bitwise XOR

 X bitwise NOT

X + Y phép cộng số nguyên rút gọn modulo 232

X <<< s Quay vòng trái s vị trí của X (0 < s < 31)

* Quá trình xử lý của bốn vòng gần như giống nhau, chỉ khác hai chi tiết sau:
  1. Mỗi vòng dùng một hàm khác nhau (theo thứ tự là : F, G, H, I)
  2. Mỗi vòng sử dụng chỉ số k (của X[k]), t (của T[t]), và s khác nhau.

**Vòng 1 (Round 1)**

Ký hiệu [abcd k s t] là bước thực hiện của phép toán

**a = b + ((a + F(b, c, d) + X[k] + T[t]) <<< s)**

Quá trình thực hiện 16 bước sau:

[ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]

[ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]

[ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]

[ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]

Giải thích**:** ví dụ biểu thức thứ 2 là [DABC 1 12 2], tương đương với:

**D = A + ((D + F(A,B,C) + X[1] + T[2]) <<< 12)**

**Nhận xét:** vòng 1 dùng hàm F, Với giá trị **t** từ 1 🡪 16và k từ 1 🡪 15

**Vòng 2 (Round 2):**

Tương tự, ký hiệu [abcd k s t] là của biểu thức :

**a = b + ((a + G(b, c, d) + X[k] + T[t]) <<< s)**

Quá trình thực hiện 16 bước :

[ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]

[ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]

[ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]

[ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]

**Nhận xét:** vòng 2 dùng hàm G, với t từ 17 🡪 32 và k = 1 + 5k mod 16

**Vòng 3 (Round 3):**

Tương tự, ký hiệu [abcd k s t] là của biểu thức :

**a = b + ((a + H(b, c, d) + X[k] + T[t]) <<< s)**

Quá trình thực hiện 16 bước:

[ABCD 5 4 33] [DABC 8 11 34] [CDAB 1 16 35] [BCDA 14 23 36]

[ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]

[ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]

[ABCD 9 4 45] [DABC 12 11 46] [CDAB 5 16 47] [BCDA 2 23 48]

**Nhận xét**: vòng 3 dùng hàm H, với t từ 33 🡪 48 và k =5 + 3k mod 16

**Vòng (Round) 4:**

Tương tự, ký hiệu [abcd k s t] là của biểu thức :

**a = b + ((a + I(b,c,d) + X[k] + T[t]) <<< s)**

Quá trình thực hiện 16 bước :

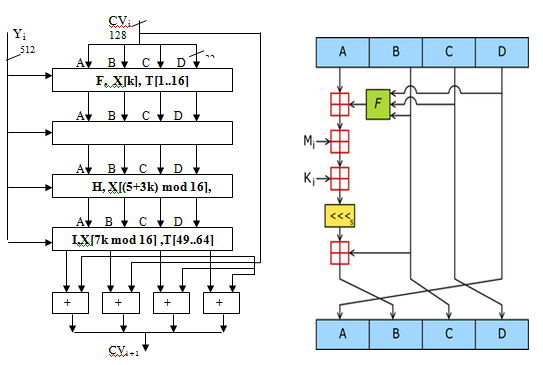
[ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]

[ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]

[ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]

[ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]

Nhận xét: vòng 4 dùng hàm I, với t từ 49 🡪 64 và k =7k mod 16



Hình 3.3. Sơ đồ xử lý một khối Hình 3.4. Sơ đồ xử lý một bước

512bis thứ i trong 64 bước của thuật toán MD5

**Bước 5 :***(tính kết quả message digest )*

Sau khi thực hiện xong bước 4, thông điệp thu gọn nhận được từ 4 thanh ghi A, B, C, D, bắt đầu từ byte thấp của thanh ghi A, và kết thúc với byte cao của thanh ghi D bằng phép nối như sau: **Message Digest = A || B || C || D.** ( **||** phép toán nối).

* **Mô tả các bước thực hiện thuật toán SHA-1:**

Khá giống với thuật toán băm MD5, thuật toán băm SHA-1 nhận thông điệp ở đầu vào có chiều dài k ≤ 264-bits, 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 đối với các bộ vi xử lý có cấu trúc 32-bits.

**Bước 1:***(chèn thêm bits)*

Thực hiện chèn một số bits vào thông điệp, đầu tiên là bit có giá trị 1 vào cuối thông điệp, tiếp theo là các bits có giá trị 0 cho đến khi chiều dài mới của thông điệp đồng dư với 448 modulo 512.

**Bước 2:***(chèn số k để có tổng chiều dài là bội 512-bits)*

Một khối có kích cỡ 64-bits (chia thành hai word 32-bits) biểu diễn giá trị k, là chiều dài của thông điệp ban đầu được lưu trữ theo dạng big-endian (word có thứ tự cao của một số được lưu trong bộ nhớ tại địa chỉ thấp) và chèn vào cuối thông điệp.

Kết quả thu được sau hai bước thực hiện là một khối dữ liệu có chiều dài là bội số của 512. Chia thông điệp này ra nhiều khối nhỏ M(1) M(2)… M(N) (N: số block), mỗi khối có độ dài là 512-bits. Tương đương với 16 word 32-bits.

**Bước 3:** *(khởi tạo các giá trị thanh ghi)*

Một bộ đệm (buffer) MD 168-bits dùng để chứa kết quả tính toán trung gian và cuối cùng của hàm băm. Có thể xem bộ đệm này là 5 thanh ghi 32-bits (H1(0) , H2(0), H3(0), H4(0), H5(0)). Những thanh ghi này được khởi tạo giá trị ban đầu theo dạng big-endian (byte thứ tự cao của một số được lưu trong bộ nhớ tại địa chỉ thấp) như sau:

H0(0) = 67 45 23 01

H1(0) = EF CD AB 89

H2(0) = 98 BA DC FE

H3(0) = 10 32 54 76

H4(0) = C3 D2 E1 F0

**Bước 4:** *( xử lý lần lược từng khối 512-bits)*

Các ký hiệu dùng trong thuật toán:

ROTLn(x) : tương đương với x <<< n (quay vòng trái n bits của x)

+ : phép cộng số nguyên modulo 232.

Các hằng số Kt dùng trong quá trình nén với các t khác nhau:

0 ≤ t ≤ 19 🡪 Kt = 5A 82 79 99

20 ≤ t ≤ 39 🡪 Kt = 6E D6 EB A1

40 ≤ t ≤ 59 🡪 Kt = 8F 1B BC DC

60 ≤ t ≤ 79 🡪 Kt = CA 62 C1 D6

Thuật toán SHA-1 xử lý từng khối 512-bits gồm 16 words (32-bits) giống như MD5. Từ 16 words này ( X[0] 🡪 X[15] ) thuật toán sẽ sinh ra 80 words Wt (với t = 0…79), theo công thức sau:

Wt = X[t] với 0 ≤ t ≤ 15

Wt = ROTL1( Wt-3  Wt-8  Wt-14  Wt-16) với 16 ≤ t ≤ 79

Các hàm SHA-1 dùng trong 80 vòng lặp con:

F1(x, y, z) = (x y)  (x  z) 0 ≤ t ≤ 19

F2 (x, y, z) = (x y z) 20 ≤ t ≤39

F3 (x, y, z) = (x y) v (x y) v (y z) 40 ≤t ≤ 59

F4 (x, y, z) = (x  y z) 60 ≤t ≤79

Trong quá trình biến đổi, tại mỗi lần lặp thứ i xử lý một khối (block) M[i] có chiều dài 512-bits, SHA-1 thực hiện 80 vòng theo **t**, kết quả thu được tại bước i sẽ cộng dồn vào 5 thanh ghi của bộ đệm MD để tiếp tục xử lý khối M[i+1] tiếp theo…

**Bước 5:***(tính kết quả message digest)*

Sau khi N khối (512-bits) của thông điệp được xử lý, ở bước 4 của vòng lặp xử lý khối cuối cùng, giá trị **hash** nhận được tại kết quả của 5 thanh ghi là thông điệp thu gọn cần tìm: **Message digest = H0(N) || H1(N) || H2(N) || H3(N) || H4(N)**

**Chương 4**

**LƯỢC ĐỒ CHỮ KÝ SỐ HD – 11**

* 1. **Yêu cầu của một hệ thống chữ ký điện tử**

Hệ thống chữ ký điện tử cần thỏa mãn các yêu cầu sau :

* Tính an toàn (security): chữ ký không thể làm giả được nếu không biết thông tin bí mật (private key) để tạo ra chữ ký.
* Tính hiệu quả (performance): ký và xác nhận chữ ký nhanh, dễ dàng.
* Tính không thể phủ nhận (non-repudiation): người ký không thể phủ nhận chữ ký của mình khi đã ký vào tài liệu.
  + 1. ***Mức độ an toàn của hệ mật RSA***

Mức độ an toàn của hệ mật RSA được đánh giá qua khả năng :

* Chống tấn công làm lộ khóa mật.
* Chống thám mã
* Chống giả mạo chữ ký
  + - 1. Khả năng chống tấn công làm lộ khóa bí mật

Việc tính khóa mật có thể thực hiện qua 2 cách như sau:

* Phân tích *modulo n* để tìm pvà q, từ đó tính (n) = (p - 1)(q - 1) rồi tính khóa mật theo công thức: *d = mod(n).* Đây là bài toán phân tích một số nguyên tố. Tấn công khóa mật bằng cách này, kẻ tấn công cần phải giải được Bài toán phân tích số.
* Từ thuật toán hình thành chữ k ý: S = , tính khóa bí mật (d) từ việc giải : d= . Đây là bài toán logarit rời rạc trong modulo n với n là hợp số. Để tấn công khóa mật thành công bằng cách này phải giải được bài toán logarit rời rạc trong modulo hợp số.
  + - 1. Khả năng chống thám mã

Do thông điệp dữ liệu M được biểu diễn thành một giá trị tương ứng *m* trong khoảng [ 0, *n-1* ] và mã hóa theo công thức : Nên về mặt lý thuyết, từ bản mã (*C*) và khóa công khai ( e ) có thể tính : M = (mod n), rồi khôi phục lại M.

* + - 1. Khả năng chống giả mạo chữ ký

Bài toán khai căn trong module hợp số cũng được sử dụng trong việc chống giả mạo chữ ký của hệ mật RSA, điều kiện để chữ ký (S) được công nhận là hợp lệ tương ứng với thông điệp dữ liệu (M) nếu cặp (M,S) thỏa mãn điều kiện sau : Do đó , về mặt lý thuyết có thể chọn một giá trị bất kỳ từ việc tính mod*n* làm chữ ký giả mạo. Tuy nhiên,giá trị này dễ dàng bị chủ thể của khóa công khai (*e*) chứng minh là giả mạo, còn việc tính được giá trị chính xác của chữ ký tương ứng với thông điệp dữ liệu M từ việc tính mod*n* thì cũng khó tương tự như việc tìm được thông điệp dữ liệu M từ khóa công khai (*e*) và bản mã ( C). Cũng có thể giả mạo được chữ ký RSA nếu tính được khóa mật (d) từ (M,S) đã biết : mod*n.* Khi đó kẻ tấn công phải giải được bài toán Logarit rời rạc trong modul hợp số.

Hiện tại, các bài toán trên vẫn được coi là các bài toán khó và chưa có một công bố nào cho thấy hệ mật RSA bị phá vỡ bằng việc giải các bài toán này. Tuy nhiên, độ an toàn của hệ mật RSA còn phụ thuộc vào một số yếu tố khác, mà một trong những yếu tố đó là việc dùng chung *modulo n.* Khi nhiều người cùng dùng chung *modulo n*, do biết được chung và khóa công khai của nhau, nên mỗi người trong số họ hoàn toàn có thể tính được khóa bí mật của những người khác. Đây chính là nhược điểm lớn nhất của hệ mật RSA xét theo khía cạnh ứng dụng.

* + 1. ***Kiểu tấn công module n dùng chung***

Simons và Norris cũng chỉ ra rằng hệ thống RSA có thể bị tấn công khi sử dụng module n dùng chung, thực vậy nếu một thông điệp M được mã hoá bằng hai khoá công khai và từ hai thành viên trong hệ thống thì được:

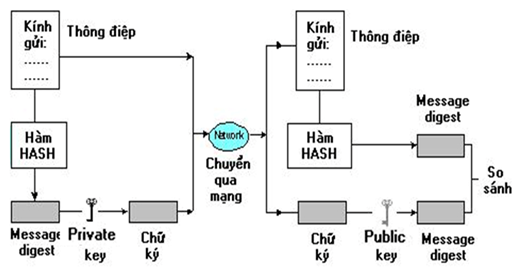
= (mod n)

= (mod n)

Sau đó người tấn công dùng thuật toán Euclide mở rộng: \*a + \*b = 1 sao cho gcd(,) = 1 thì M được khôi phục lại như sau: M =. mod*n*.

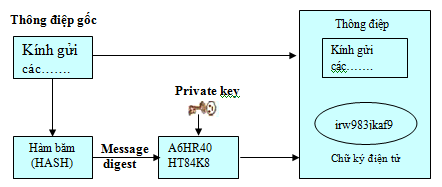
* 1. **Lược đồ chung của chữ ký điện tử**

Một lược đồ chữ ký điện tử bao gồm 2 thành phần: thuật toán ký, và thuật toán xác nhận chữ ký. Nghĩa là, nếu người A muốn gửi cho người B một thông điệp x, thì A dùng một thuật toán và khoá bí mật của mình để tạo chữ ký y = signkdA(x), rồi gửi cả thông điệp x lẫn chữ ký y cho B. Sau khi nhận được thông điệp x và chữ ký y. B sẽ dùng thuật toán cùng với khóa công khai của A để xác nhận chữ ký y có phải là chữ ký của A cho thông điệp x này hay không verifykeA(x, y) = {true, false}.



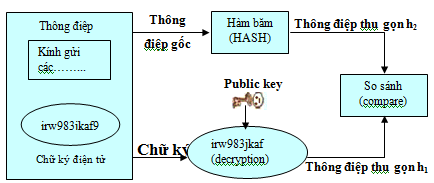
**Hình 4.1. Mô hình tổng quát của chữ ký điện tử**

* + 1. ***Các bước thực hiện tạo chữ ký điện tử***
* Người gửi sử dụng một hàm băm, để biến đổi thông điệp x thành một thông điệp thu gọn (message digest) **h** có chiều dài cố định: **h** = Hash(x).
* Người gửi dùng khoá riêng kd của mình mã hóa chuổi **h**: y = Ekd(h), kết quả y thu được chính là chữ ký điện tử (digital signature) đối với thông điệp x.
* Cuối cùng chữ ký **y** có thể được nối vào cuối thông điệp **x** hoặc lưu vào một file gửi kèm với thông điệp. Sau khi đã ký nhận mọi sự thay đổi của thông điệp sẽ được phát hiện trong quá trình kiểm tra xác nhận chữ ký. Điều này đảm bảo cho người nhận tin rằng thông điệp họ nhận được đích thực là của người gửi và nội dung thông điệp hoàn toàn không bị thay đổi.



**Hình 4.2. Sơ đồ minh họa các bước tạo chữ ký điện tử**

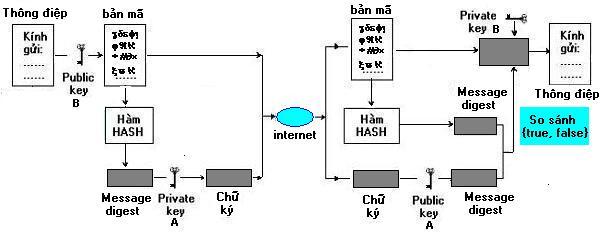
* + 1. ***Các bước thực hiện kiểm tra tính đúng của chữ ký điện tử***
* Người nhận dùng khoá công khai (key public) ke của người gửi để giải mã chữ ký điện tử y vừa nhận, khôi phục lại thông điệp thu gọn: **h1**= Dke(y).
* Người nhận sử dụng hàm băm giống như người gửi để biến đổi thông điệp x nhận được thành thông điệp thu gọn: **h2** = Hash(x).
* So sánh kết quả, nếu h1 = h2 thì chấp nhận chữ ký là của người gửi. Ngược lại, chữ ký trên thông điệp không được chấp nhận.



**Hình 4.3. Sơ đồ minh họa các bước kiểm tra chữ ký điện tử**

**Nhận xét:** Lược đồ chữ ký điện tử theo kiểu này, cho phép xác định được chủ nhân đích thực của thông điệp, đồng thời đảm bảo nội dung của thông điệp không bị sửa đổi hay làm giả mạo bởi người khác trong quá trình truyền đi trên mạng. Nhưng nội dung của thông điệp có thể đọc được, do trong lược đồ này chỉ thực hiện mã hóa một khối dữ liệu nhỏ đặt trưng cho thông điệp mà không mã hoá toàn bộ thông điệp, điều này không phù hợp với nhu cầu trao đổi các thông tin bí mật thông qua internet. Vì vậy để có thể đảm bảo được bí mật của nội dung, người gửi cần thực hiện quá trình mã hóa thông điệp bằng khóa công khai của người nhận, trước khi thực hiện ký xác nhận vào tài liệu, và người nhận phải thực hiện thêm một bước giải mã thông điệp bằng khoá riêng của mình sau khi kiểm tra đúng chữ ký của người gửi.

* + 1. ***Các bước thực hiện mã hoá và tạo chữ ký cho thông điệp***
* Người gửi A mã hóa thông điệp x bằng khóa công khai của người nhận B: C = EkeB(x). (keB là khoá công khai của người nhận B)
* Người gửi A thực hiện bước tạo chữ ký để xác nhận bản mã C với khóa riêng của mình: y = SigkdA(C).
* Gửi chữ ký **y** và bản mã C đến người nhận B.

****

**Hình 4.4. Mô hình chữ ký điện tử dùng quá trình mã hóa và giải mã**

* + 1. ***Mô tả các bước kiểm tra chữ ký và giải mã thông điệp***
* Người nhận kiểm tra chữ ký trên thông điệp bằng khóa công khai của người gửi A: verkeA(C, y) = {true, false}.
* Nếu bước kiểm tra ở trên là đúng (true), thì người nhận tiếp tục thực hiện quá trình giải mã C với khóa riêng của mình: x = DkdB(C), để khôi phục lại thông điệp x. Ngược lại chữ ký của A đối với tài liệu x là không hợp lệ.
  1. **Phương pháp hình thành và kiểm tra chữ ký số**
     1. ***Thuật toán hình thành khóa***
        1. Hình thành các tham số hệ thống

1. Sinh 2 số nguyên tố p và q lớn,mạnh với q|(p-1). Có thể chọn p, q theo chuẩn chữ ký số DSS của Mỹ như sau:  và , với:

L = 1024, N = 160; L = 2048, N = 224; L = 2048, N = 256; L = 3072, N = 256.

2. Chọn giá trị h thỏa mãn : 1< h < p ; g = với g là phần tử sinh của . . Ở đây: h là phần tử có thể chọn ngẫu nhiên trong 

3. Lựa chọn hàm băm

Các giá trị (p, q, g) là các tham số công khai trong quá trình hình thành và kiểm tra chữ ký.

* + - 1. Hình thành khóa

1. Chọn khóa bí mật (x) thỏa mãn : 1< x < n ;

2. Tính khóa công khai (y) theo công thức : y = ;

* + 1. ***Thuật toán hình thành chữ ký***

Dữ liệu đầu vào của thuật toán bao gồm : Thông điệp dữ liệu cần ký (M) và khóa bí mật (x) của đối tượng ký. Kết quả đầu ra của thuật toán là chữ ký số (R,S) tương ứng với M. Thuật toán bao gồm các bước :

Hình thành phần thứ nhất của chữ ký (R) theo công thức :

;

ở đây, H() là hàm băm được chọn đủ an toàn, chẳng hạn: SHA-1, MD5 và toán tử || là phép nối 2 xâu.

Hình thành phần thứ hai của chữ ký (S) theo công thức :

;

Cặp giá trị (R,S) là chữ ký vào văn bản M.

* + 1. ***Thuật toán kiểm tra chữ ký***

Dữ liệu đầu vào của thuật toán bao gồm : Thông điệp dữ liệu cần thẩm tra (M) và chữ ký tương ứng (R,S),khóa công khai (y) của đối tượng ký. Đầu ra của thuật toán là sự khẳng định về tính hợp lệ của chữ ký (R,S) hay sự công nhận về nguồn gốc và tính toàn vẹn của thông điệp dữ liệu được thẩm tra (M). Thuật toán bao gồm các bước sau :

Tính R’ theo công thức :

;

Kiểm tra nếu R’=R thì chữ ký (R,S) hợp lệ,do đó nguồn gốc và tính toàn vẹn của thông điệp dữ liệu cần thẩm tra (M) được công nhận.

* 1. **Tính đúng đắn và mức độ an toàn của lược đồ**
     1. ***Tính đúng đắn***

Tính đúng đắn của lược đồ mới xây dựng thể hiện sự phù hợp giữa thủ tục hình thành và thủ tục kiểm tra chữ ký.

Lược đồ HD-11 tồn tại đẳng thức: .

Chứng minh:

Từ tính hợp lệ của chữ ký (e,s) ta có:



Từ tính toàn vẹn của văn bản M suy ra:



* + 1. ***Mức độ an toàn***

Mức độ an toàn của lược đồ cơ sở HD 11 được đánh giá qua các khả năng sau

* + - 1. Khả năng chống tấn công làm lộ khóa bí mật

Khóa công khai (y) được hình thành từ khóa bí mật (x) theo công thức y = . Vì vậy ,để tính khóa bí mật x thừ khóa công khai (y) cần phải giải được bài toán khai căn trong modulo hợp số tương tự như với hệ mật Elgamal. Như vậy khả năng chống tấn công làm lộ khóa mật của HD 11 phụ thuộc vào mức độ khó giải của bài toán khai căn trong hợp số.

Điều đặc biệt đối với lược đồ này là ở chỗ, chỉ một khóa duy nhất được sử dụng để ký cho các ký văn bản khác nhau, nhưng tính an toàn của lược đồ vẫn được bảo đảm.

Thật vậy, do thành phần *r* được tính theo công thức:

,

nên để tính  từ *r*, rồi từ đó tính khóa *x* theo công thức:



kẻ tấn công buộc phải giải bài toán logarit rời rạc, mà việc giải bài toán logarit rời rạc trên Z\*p hiện nay xem như là không thực hiện được.

Mặt khác, với công thức tính thành phần thứ hai *s* của chữ ký:



kẻ tấn công cũng không thể tính được khóa mật x. Giả sử có k văn bản: M1, M2, ..., Mk , khi đó thành phần thứ 2 của chữ ký được tính như sau:





...........................................



Ở đây, (*e*1,*s*1), (*e*2,*s*2),..., (*ek*,*sk*) là chữ ký tương ứng với các văn bản *M*1, *M*2,..., *Mk*.

Như vậy, để ký vào các văn bản khác nhau người ký chỉ cần chọn một khóa x duy nhất. Để phá khóa, kẻ tấn công phải giải hệ *k* phương trình với *km*+1 ẩn, nhưng đây là công việc không khả thi. Vậy tính an toàn của lược đồ vẫn được bảo đảm.

* + - 1. Khả năng chống giả mạo chữ ký

Giả sử (,) là chữ ký giả mạo tương ứng với thông điệp dữ liệu M. Chữ ký giả mạo này sẽ được coi là hợp lệ,do đó nguồn gốc và tính toàn vẹn của M sẽ được công nhận nếu (M,,) thỏa mãn các điều kiện sau :

# Chương 5

# CÀI ĐẶT THỬ NGHIỆM HD  - 11

* 1. **Một số thuật toán cơ sở** 
     1. ***Thuật toán sinh số nguyên tố***

Input : n – nguyên dương.

Output : p – nguyên tố lẻ.

Thuật toán:

1. p=n
2. while (i<>n/2)

if (p *mod* 2 = 0) then p=p+1

for i=2 to n/2 do

r= p *mod* i

if r = 0 then

p=p+1

break

1. return p
   * 1. ***Thuật toán tìm ước số chung lớn nhất***

Input: a,b – nguyên, dương, a>b.

Output: gcd(a,b)- ước số chung lớn nhất của a và b.

Thuật toán:

1. While (b<>0) do

r =a mod b

a=b

b=r

1. Return a
   * 1. ***Thuật toán tính nghịch đảo***

* Input: a,n – nguyên, dương, gcd(a,n)=1.
* Output:
* Thuật toán:

1. k=0
2. t=1
3. while (t<>0) do

k=k+1

1. Return b
   * 1. ***Thuật toán tìm phần tử sinh của nhóm***

* Input: p – số nguyên tố; – số nguyên tố;
* Output: a – phần tử sinh của
* Thuật toán:

1. Select a: 1<a<p
2. For i=1 to k do

If (b<>) then return a

1. go to
   * 1. ***Thuật toán "bình phương và nhân"***

Thuật toán“bình phương và nhân” để tính: . Trong thuật toán này, ta coi rằng số mũ b được biểu thị ở dạng nhị phân như sau:

* Input: a,k,n – nguyên dương.
* Output:
* Thuật toán:

1. …….
2. b=1
3. c=a
4. for i=0 to t do

if

if

1. return b

**Ví dụ** : tính 97263533 mod 11413

3533=211+210+28+27+26+23+22+1=110111001101

d:=1

|  |  |  |
| --- | --- | --- |
| I | bi | d mod 11413 |
| 11 | 1 | 12 \* 9726 = 9726 |
| 10 | 1 | 97262 \* 9726 = 2659 |
| 9 | 0 | 26592 = 5634 |
| 8 | 1 | 56342 \* 9726 = 9167 |
| 7 | 1 | 91672 \* 9726 = 4958 |
| 6 | 1 | 49582 \* 9726 = 7783 |
| 5 | 0 | 77832 = 6298 |
| 4 | 0 | 62982 = 4629 |
| 3 | 1 | 46292 \* 9726 = 10185 |
| 2 | 1 | 101852 \* 9726 = 105 |
| 1 | 0 | 1052 = 11025 |
| 0 | 1 | 110252 \* 9726 = 5761 |

97263533 mod 11413 = 5761

* 1. **Cài đặt thử nghiệm thuật toán**

Tạo khóa

Tạo ra cặp khóa bí mật - công khai cho người dùng.

***- Sinh số nguyên tố***

long sinh\_so\_nt(long pp)

{

long i,r,pn;

pn=pp;

if ((pp%2)==0)pp++;

i=0;

while(i != pn/2)

{

if ((pp%2)==0)pp++;

for(i=2;i<pn/2;i++)

{

r=pp%i;

if (r==0)

{

pp++;

break;

}

***- Tính khóa công khai***

printf("\n\n TINH KHOA CONG KHAI:");

printf("\n 1. Khoa bi mat: x = ");scanf("%d",&X);

l = toBinary(X,m);

Z=luy\_thua(g,p,m,l);

Y=nghich\_dao(Z,p);

printf("\n 2. Khoa cong khai: y = %d",Y);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int toBinary(long n,int \*pm)

{

int i,j,t,t1,l1;

i=0;

t=n;

while(t !=0)

{

pm[i]=t%2;

i++;

t=t/2;

}

l1=i;

return l1;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

long nghich\_dao(long x,long n)

{

int k;

long t;

k=1;

while((k\*n+1)%x != 0)

{

k++;

}

t=(k\*n+1)/x;

return t;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

long luy\_thua(long xx, long nn,int \*pm , int ll)

{

int i;

long bb,cc;

cc=1;

bb=xx;

for(i=0; i<ll;i++)

{

if (pm[i]==1)

{

cc=(cc\*bb)%nn;

}

bb=(bb\*bb)%nn;

}

return cc;

Sinh chữ ký

Việc sinh chữ ký này phải thông qua việc tạo khóa.

printf("\n\n SINH CHU KY:");

printf("\n 1. Gia tri dai dien cua Ban tin can ky: E = H(M) = ");scanf("%d",&E);

printf("\n 2. Chon: k = ");scanf("%d",&K);

l = toBinary(K,m);

Z=luy\_thua(g,p,m,l);

R = Z % q;

printf("\n 3. R = %d",R);

Z = nghich\_dao(R,q);

Z = (Z\*K)%q;

S = (X\*E)%q;

S = (S + Z)%q;

printf("\n 4. Chu ky (R,S) = (%d,%d)",R,S);

Xác thực chữ ký

Kiểm tra tính đúng đắn của chữ ký số.

Tạo khóa

Tạo chữ ký số

Xác thực chữ ký số

**Hình 5.1. Sơ đồ chương trình chữ ký số**

printf("\n\n KIEM TRA CHU KY:");

e = (R\*S)%q;

l = toBinary(e,m);

Z=luy\_thua(g,p,m,l);

e = (R\*E)%q;

l = toBinary(e,m);

Z2=luy\_thua(Y,p,m,l);

R\_ = (Z\*Z2)%p;

R\_ = R\_ %q;

printf("\n 1. R\* = {g^(R.S)\* y^(R.E) mod p} mod q = %d",R\_);

if (R\_ == R)

{

printf("\n 2. R\* = R: Chu ky hop le!");

}

else

{

printf("\n 2. R\* # R: Chu ky khong hop le!");

}

* [Kết luận](#_Toc262644864)
* Mục đích của chương này là nghiên cứu cài đặt thử nghiệm việc tạo và kiểm tra chữ ký chính xác.

# KẾT LUẬN

Vấn đề chữ ký số điện tử là một trong những vấn đề khó trong lĩnh vực mật mã học. Nó là một vấn đề không mới, đang được phát triển ở nước ta hiện nay và có nhiều công việc cần giải quyết nếu muốn xây dựng một hệ thống ký số điện tử đạt tiêu chuẩn quốc gia. Hướng tiếp cận theo mật mã học khóa công khai là hướng tiếp cận dựa vào yêu cầu thực tế công nghệ là công khai và khóa mới là cái bí mật, độ an toàn của hệ thống không dựa vào độ an toàn của công nghệ mà chính là khóa. Qua bốn chương đồ án đã trình bày các công cụ để xây dựng chữ ký số, lược đồ chữ ký số RSA, ElGamal và nghiên cứu thử nghiệm lược đồ chữ ký số HD 11 dựa trên lược đồ cơ sở chữ ký số ElGamal. Đồ án mới ở mức nghiên cứu đưa ra định hướng xây dựng thuật toán chữ ký số HD 11 khắc phục nhược điểm dùng chung module n và chưa tạo được đa chữ ký của thuật toán RSA.

**Kết quả đạt được**

- Trình bày tổng quan về chữ ký số điện tử, các phân loại, mô hình cũng như vai trò của chữ ký số

- Trình bày về mật mã học, hàm băm mật mã học – những công cụ chính để tạo ra chữ ký số RSA

- Đưa ra đề xuất,lược đồ chữ ký số HD 11 dựa trên cơ sở lược đồ chữ ký số ElGamal.

**Hướng phát triển**

* Nghiên cứu xây dựng thành chương trình hoàn thiện trên cơ sở ngôn ngữ lập trình C#. Xây dựng chương trình sử dụng hàm băm SHA-1 với độ dài của giá trị băm là 160 bit – an toàn gần như tuyệt đối với độ tính toán hiện nay, giá trị của khóa là 1024 bit, đầu vào của chương trình tạo chữ ký số là các file có độ lớn bất kỳ - tốc độ băm file tỉ lệ với độ lớn của file.

- Thử việc gán nhãn thời gian cho dữ liệu để đảm bảo độ an toàn cũng như dễ dàng giải quyết trong các trường hợp ví dụ như người dùng bị mất khóa bí mật…

- Xây dựng việc cung cấp xác thực khóa công khai cho cả tổ chức, cá nhân, mở rộng mô hình tùy vào yêu cầu sử dụng…

TÀI LIỆU THAM KHẢO

[1] Phan Đình Diệu ,Giáo trình an toàn thông tin và mật mã,Đại học quốc gia Hà Nội 2000

[2] Nguyễn Hiểu Minh,Slide bài giảng An ninh mạng,Học viện kỹ thuật quân sự 2012

[3] Lê Phúc , Bảo mật hệ thống thông tin , Học viện bưu chính viễn thông TP.HCM 2007

[4] Trịnh Nhật Tiến ,Giáo trình an toàn thông tin 2008