**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM**

**TP. HCM**

KHOA CÔNG NGHỆ THÔNG TIN

Ngành An Toàn Thông Tin

---------------o0o---------------



**ĐỒ ÁN MÔN HỌC**

**ĐỀ TÀI :** **TÌM HIỂU VỀ CHỮ KÝ ĐIỆN TỬ VÀ XÂY DỰNG ỨNG DỤNG TẠO VÀ XÁC THỰC CHỮ KÝ ĐIỆN TỬ**

Sinh viên thực hiện:

Nguyễn Văn Quang 2033181060

Đào Chiến Thắng 2033181066

Giảng viên hướng dẫn: Mạnh Thiên Lý

TP. Hồ Chí Minh, Tháng 6 năm 2021

# LỜI CẢM ƠN

Chúng em xin chân thành gửi lời cảm ơn sâu sắc tới cô Mạnh Thiên Lý - người luôn chỉ bảo, nhắc nhở, hướng dẫn, cung cấp những tài liệu quý giá, kiến thức bổ ích và dẫn dắt tụi em hoàn thành đồ án này.

Chúng em cũng xin gửi lời cảm ơn các thầy cô giáo trong khoa Công nghệ thông tin - trường Đại học Công nghiệp Thực phẩm TP. HCM và gia đình đã tạo điều kiện giúp đỡ về vật chất và tinh thần để chúng em có thể tập trung và có động lực để hoàn thành tốt đồ án này.

Sinh viên thực hiện

Nguyễn Văn Quang Đào Chiến Thắng

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Nhóm sinh viên gồm : 1. Nguyễn Văn Quang MSSV: 2033181060

2. Đào Chiến Thắng MSSV: 2033181066

Nhận xét :  ……………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

Điểm đánh giá:

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

………………………………………………………………………………………………….

Ngày . ……….tháng ………….năm 2021

( Ký tên, ghi rõ họ và tên)

# MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc79240283)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 3](#_Toc79240284)

[MỤC LỤC 4](#_Toc79240285)

[DANH MỤC HÌNH ẢNH 7](#_Toc79240286)

[THUẬT NGỮ VIẾT TẮT 8](#_Toc79240287)

[LỜI NÓI ĐẦU 9](#_Toc79240288)

[CHƯƠNG 1. TỔNG QUAN VỀ CÁC LOẠI MÃ HOÁ 11](#_Toc79240289)

[1.1 Giới thiệu về mật mã học 11](#_Toc79240290)

[1.1.1 Mật mã học là gì? 11](#_Toc79240291)

[1.1.2 Các thành phần mật mã học 11](#_Toc79240292)

[1.1.3 Quy trình cơ bản của mật mã học 12](#_Toc79240293)

[1.2 Phân loại hệ mã hoá 12](#_Toc79240294)

[1.2.1 Hệ mã dòng - Stream Cipher 12](#_Toc79240295)

[1.2.2 Hệ mã khối - Block Cipher 13](#_Toc79240296)

[1.2.3 Hệ mã hoá đối xứng 14](#_Toc79240297)

[1.2.4 Hệ mã hoá bất đối xứng 20](#_Toc79240298)

[1.2.5 Hệ mã hoá RSA 21](#_Toc79240299)

[CHƯƠNG 2. HÀM BĂM (HASH) 26](#_Toc79240300)

[2.1 Sơ lược về hàm băm 26](#_Toc79240301)

[2.2 Khái niệm hàm băm 26](#_Toc79240302)

[2.3 Tính chất của hàm băm 26](#_Toc79240303)

[2.4 Thuộc tính cần thiết 26](#_Toc79240304)

[2.5 Cách hoạt động của hàm băm 27](#_Toc79240305)

[2.6 Các loại hàm băm phổ biến 28](#_Toc79240306)

[2.6.1 MD5 (Message - Digest Algorithm 5) 28](#_Toc79240307)

[2.6.2 SHA (Secure Hashing Algorithm) 29](#_Toc79240308)

[2.7 Ứng dụng của hàm băm 30](#_Toc79240309)

[CHƯƠNG 3. LÝ THUYẾT VỀ CHỮ KÝ SỐ 32](#_Toc79240310)

[3.1 Giới thiệu về chữ ký số 32](#_Toc79240311)

[3.2 Khái niệm chữ ký số 32](#_Toc79240312)

[3.3 Các ưu điểm của chữ ký số 32](#_Toc79240313)

[3.4 Quy trình tạo và xác thực chữ ký số 34](#_Toc79240314)

[3.4.1 Thuật toán tạo khoá 34](#_Toc79240315)

[3.4.2 Tạo chữ ký và ký vào thông điệp 34](#_Toc79240316)

[3.4.3 Kiểm tra và xác thực chữ ký số 34](#_Toc79240317)

[CHƯƠNG 4. MỘT SỐ QUY ĐỊNH, THÔNG TƯ VỀ CHỮ KÝ SỐ 37](#_Toc79240318)

[4.1 Quy định về ký số, kiểm tra chữ ký số trên văn bản điện tử trong cơ quan nhà nước việt nam 37](#_Toc79240319)

[4.1.1 Quy định về giá trị pháp lý của chữ ký số 37](#_Toc79240320)

[4.1.2 Quy định về điều kiện tạo chữ ký số đảm bảo an toàn đối với các đơn vị cung cấp 37](#_Toc79240321)

[4.2 Chứng thư số, chữ ký số nước ngoài tại việt nam 38](#_Toc79240322)

[4.2.1 Đối tượng sử dụng chứng thư số nước ngoài 38](#_Toc79240323)

[4.2.2 Đối tượng sử dụng chứng thư số nước ngoài 38](#_Toc79240324)

[4.2.3 Phạm vi hoạt động và thời hạn giấy phép sử dụng chứng thư số nước ngoài tại Việt Nam 38](#_Toc79240325)

[4.2.4 Đối với thuê bao sử dụng chứng thư số nước ngoài tại Việt Nam: 38](#_Toc79240326)

[4.2.5 Đối với tổ chức cung cấp dịch vụ chứng thực chữ ký số nước ngoài có chứng thư số được công nhận tại Việt Nam 39](#_Toc79240327)

[CHƯƠNG 5. XÂY DỰNG ỨNG DỤNG TẠO VÀ XÁC THỰC CHỮ KÝ SỐ 40](#_Toc79240328)

[5.1 Giới thiệu về phần mềm 40](#_Toc79240329)

[5.2 Thực hiện tạo khóa bí mật và khóa công khai 41](#_Toc79240330)

[5.3 Thực hiện tạo chữ ký số bằng giải thuật RSA 43](#_Toc79240331)

[5.4 Thực hiện giải mã để xác thực cho chữ ký được tạo bằng giải thuật RSA 45](#_Toc79240332)

[KẾT LUẬN 48](#_Toc79240333)

[PHỤ LỤC 49](#_Toc79240334)

[TÀI LIỆU THAM KHẢO 51](#_Toc79240335)

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Mô hình mã hoá dữ liệu cơ bản 13](#_Toc79245240)

[Hình 1.2 Mô hình về hệ mã hoá dòng Stream Cipher 14](#_Toc79245241)

[Hình 1.3 Mô hình về hệ mã hoá khối Block Cipher 15](#_Toc79245242)

[Hình 1.4 Bảng ví dụ sắp xếp thuật toán mã hoá thay thế 16](#_Toc79245243)

[Hình 1.5 Bảng mã dịch chuyển 18](#_Toc79245244)

[Hình 1.6 Cách hoạt động của playfair 19](#_Toc79245245)

[Hình 1.7 Cách xếp ma trận khóa của playfair 20](#_Toc79245246)

[Hình 1.8 Kết quả trả về khi dùng playfair 20](#_Toc79245247)

[Hình 1.9 Quy trình mã hoá RSA 24](#_Toc79245248)

[Hình 2.1 Sơ đồ hoạt động hàm băm 28](#_Toc79245249)

[Hình 2.2 Ví dụ về hàm băm 29](#_Toc79245250)

[Hình 2.3 Ví dụ về hàm băm MD5 bằng công cụ HashCalc 30](#_Toc79245251)

[Hình 3.1 Quy trình tạo và xác thực chữ ký số 36](#_Toc79245252)

[Hình 5.1 Giao diện started của phần mềm. 41](#_Toc79245253)

[Hình 5.2 Các độ dài khóa mà chương trình hỗ trợ. 42](#_Toc79245254)

[Hình 5.3 Code truyền giá trị độ dài khoá vào đối tượng RSA 42](#_Toc79245255)

[Hình 5.4 Code tạo khoá công khai và bí mật 43](#_Toc79245256)

[Hình 5.5 Khóa bí mật và khoá công khai 44](#_Toc79245257)

[Hình 5.6 Giao diện tạo chữ ký số. 44](#_Toc79245258)

[Hình 5.7 Tuỳ chọn nén file sau khi tạo thành công chữ ký 46](#_Toc79245259)

[Hình 5.8 Giao diện xác thực chữ ký số. 47](#_Toc79245260)

[Hình 5.9 Chữ ký giống với văn bản. 47](#_Toc79245261)

[Hình 5.10 Chữ ký không giống với văn bản. 48](#_Toc79245262)

# THUẬT NGỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Tên viết tắt** | Tên tiếng anh | Tên tiếng việt |
| **DES** | Data Encryption Standard | Chuẩn mã hoá dữ liệu |
| **RSA** | Rivest - Shamir - Adleman | Hệ mã hoá khoá công khai RSA |
| **MAC** | Message Authentication Code | Mã xác thực thông báo |
| **ƯCLN** |  | Ước chung lớn nhất |
| **OTP** | One Time Pad | Một trong những hệ mã dòng |
| **MD** | Message - Digest Algorithm | Thuật toán tóm tắt hóa thông điệp |
| **SHA** | Secure Hash Algorithm | Thuật toán băm an toàn |

# LỜI NÓI ĐẦU

Từ thuở xa xưa, trong các hoạt động giao dịch giữa các bên đều cần các biện pháp xác thực để đảm bảo tính an toàn của giao dịch, và để có thể chứng thực hay xác minh ta có rất nhiều cách, trong số đó chữ ký viết tay là hình thức đơn giản và an toàn nhất. Nhưng trong thời đại hiện nay chữ ký viết tay ngày càng kém hiệu quả, vì nó hoàn toàn có thể bị làm giả và sự bất tiện của việc phải đồng thời có sự hiện diện của hai bên để có thể ký kết hợp đồng. Nhất là thời gian gần đây khi dịch bệnh ngày càng hoành hành và con người phải hạn chế tiếp xúc với nhau khiến điều đó càng trở nên khó khăn.

Ở thời điểm công nghệ phát triển như bây giờ, việc giao dịch trên mạng trở nên khá quen thuộc với mỗi người, và trong các cuộc giao dịch yêu cầu cần có các hợp đồng thì lại đặt ra các vấn đề lớn về tính xác thực và an toàn của hợp đồng vì nó có thể bị lợi dụng bởi “bên thứ ba” hoặc cả hai bên thực hiện giao dịch cho các mục đích xấu.

Từ những vấn đề về trên, chữ ký điện tử đã được cho ra đời để giải quyết chúng. Bản thân chữ ký điện tử mang một số nét tương đồng với chữ ký viết tay nhưng bảo mật an toàn hơn vì có sử dụng các công nghệ mã hóa, chứng thực và hàm băm. Đó cũng là lý do tại sao mà chúng em chọn đề tài đồ án môn học: “TÌM HIỂU VỀ CHỮ KÝ ĐIỆN TỬ VÀ XÂY DỰNG ỨNG DỤNG TẠO VÀ XÁC THỰC CHỮ KÝ ĐIỆN TỬ” cùng với mục tiêu nắm vững được kiến thức về chữ ký số và tạo ra một ứng dụng trên nền tảng Window có thể tạo và xác thực chữ ký số, góp phần cống hiến cho đời sống thành quả từ những kinh nghiệm và kiến thức thu được sau khi thực hiện đồ án này.

Ngoài phần mở đầu và kết luận, đồ án này bao gồm 5 chương

Chương 1: Tổng quan về các loại mã hoá

Trong chương này chúng ta sẽ được biết thêm về các khái niệm mã hoá, một số các phương pháp mã hoá, các dẫn chứng, ví dụ cụ thể và giới thiệu chi tiết về thuật toán khoá công khai RSA.

Chương 2: Hàm băm

Trong chương này chúng ta sẽ tìm hiểu thêm về các loại hàm băm MD5, SHA, các phương pháp băm, các thuật toán và ứng dụng của chúng trong cuộc sống, lựa chọn các hàm băm để có thể áp dụng trong quy trình tạo chữ ký số.

Chương 3: Lý thuyết về chữ ký số

Trong chương này chúng ta sẽ tìm hiểu khái niệm về chữ ký số và các quy trình để tạo và xác thực chữ ký số áp dụng từ kiến thức ở chương 1 và 2.

Chương 4: Một số luật và quy định về dịch vụ chữ ký số

Trong chương này chúng ta tìm hiểu và nắm rõ được một số thông tư, nghị định, luật được Nhà nước ban hành về chữ ký số, ứng dụng thực tế trong giao dịch giữa các quốc gia.

Chương 5: Xây dựng ứng dụng tạo và xác thực chữ ký số

Trong chương này chúng ta sẽ tiến hành chắt lọc những tinh hoa về chữ ký số và tạo ra một chương trình có khả năng xác tạo ra chữ ký số cũng như có thể xác thực chữ ký số đã tạo bằng các thuật toán mã hoá RSA và hàm băm.

# TỔNG QUAN VỀ CÁC LOẠI MÃ HOÁ

## Giới thiệu về mật mã học

### Mật mã học là gì?

Mật mã học là ngành đã có lịch sử đã hàng nghìn năm, bắt đầu từ thời đại Hy Lạp cổ đại với mật mã học cổ điển sơ khai chỉ bằng bút và giấy, nó phát triển thành mật mã học hiện đại thời nay với điện cơ, điện tử, máy tính. Claude Shannon - cha đẻ của mật mã toán học từ những năm 1949 đã mở ra thời đại mới của mật mã học hiện đại khi ông công bố các tài liệu lý thuyết về tin học và truyền thông trong các hệ thống bảo mật.

Mật mã học đi liền với quá trình mã hoá *(Cryptography)* - cách thức chuyển đổi nội dung của một văn bản, một tập tin sang một dạng dữ liệu khác *(bản mờ)*, khiến chúng khó có thể đọc được bởi những truy cập bất hợp pháp đến dữ liệu được truyền đi - và quá trình giải mã (*Cryptanalysis*) - quá trình chuyển đổi bản mờ đã qua mã hoá thành dạng văn bản gốc có thể đọc được *(bản gốc).* Hai quá trình này đảm bảo tính bí mật cho các thông tin, dữ liệu quan trọng chẳng hạn như trong quân sự, tình báo, ngoại giao, kinh tế, thương mại, …

Vào các năm gần đây, phạm vi ứng dụng của mật mã hóa được mở rộng vô cùng rộng rãi. Mật mã hóa hiện đại cung cấp cơ chế cho nhiều hoạt động trong đời sống sinh hoạt hơn là chỉ duy nhất có công dụng giữ bí mật và có một loạt các ứng dụng như: Các giao dịch tài chính, chuyển khoản, mua sắm hàng hoá, thư từ, tài liệu, chứng thực khóa công khai, bầu cử điện tử hay tiền điện tử, … Mọi thứ được thực hiện nhiều qua môi trường mạng đòi hỏi dữ liệu phải được bảo mật tốt dẫn đến việc tất cả quá trình trao đổi đó đều phải được mã hoá. Thậm chí những người không có nhu cầu về tính bí mật cũng sử dụng các công nghệ mật mã hóa, thường được các kỹ sư thiết kế và thiết lập sẵn trong các cơ sở hạ tầng của công nghệ tính toán và liên lạc viễn thông.

### Các thành phần mật mã học

*Một hệ mã hoá bao gồm các yếu tố quan trọng sau:*

* Thông báo, văn bản gốc M: Là một chuỗi hữu hạn các ký hiệu lấy từ một bảng chữ cái Z nào đó và được ký hiệu là M.
* Mã hoá: Là quy trình biến đổi văn bản gốc khiến nó không thể đọc được đối với bất kỳ người khác ngoài người nhận được mong muốn.
* Phép mã hoá thường được ký hiệu là e (M), với M là thông báo cần mã hoá.
* Khoá K: Là một thông số đầu vào của phép mã hoá hoặc giải mã. Khoá dùng để mã hoá ký hiệu là Ke, khoá dùng để giải mã ký hiệu là Kd.
* Chuỗi mật mã, bản mờ C: Là dữ liệu bản gốc đã được mã hoá, ký hiệu là:

c = e (m, ke).

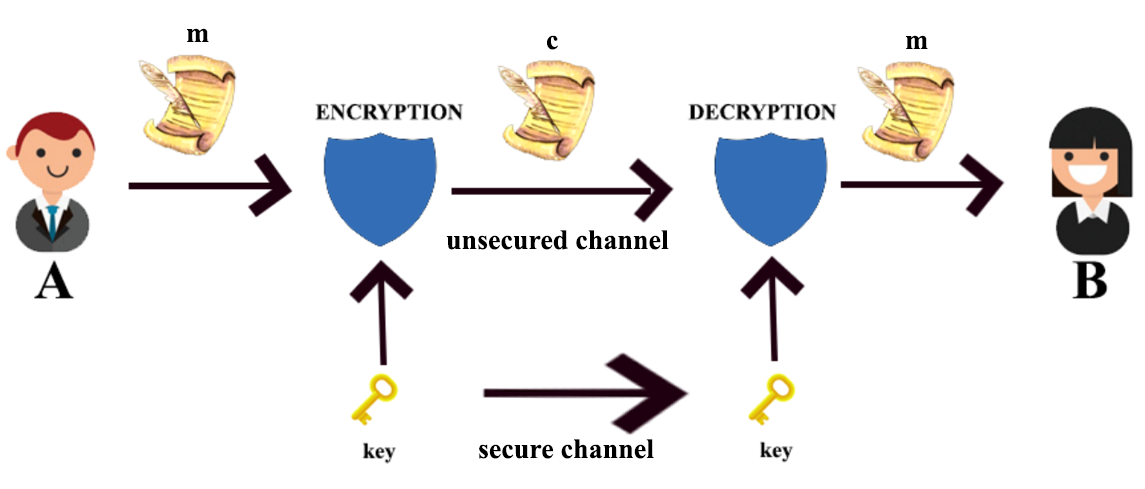
* Giải mã: Là quá trình ngược lại với mã hoá, dùng khoá K để chuyển dữ liệu bản mờ C về lại dữ liệu gốc M, ký hiệu là:

d (c, kd) = m.

### Quy trình cơ bản của mật mã học

*Ví dụ 1.1 về quy trình cơ bản của mật mã học*

A (người gửi) muốn gửi cho B (người nhận) một thông báo m (bản rõ), A dùng thuật toán phép mã hoá e kết hợp khoá K (key) để biến thông báo m thành chuỗi mật mã c (bản mờ) và gửi cho B. Bản c được truyền đi qua bằng các kênh truyền bình thường, giả sử có bị một người xấu hay một hacker bắt được gói tin c thì cũng không thể đọc được vì thông báo đã bị mã hoá, còn khoá K, A gửi cho B bằng một thuật toán truyền khoá bí mật. Đến nơi, B thu thập khoá K và chuỗi mật mã c lại, sử dụng thuật toán phép giải mã d kết hợp khoá K để chuyển chuỗi mật mã c trở về thông báo gốc m. Vậy là thông báo đã được truyền đi một cách an toàn từ A đến tay B.



Hình 1.1 Mô hình mã hoá dữ liệu cơ bản

## Phân loại hệ mã hoá

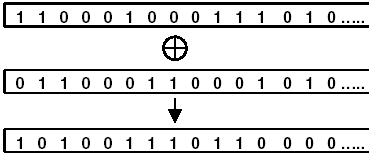
Mật mã học được chia ra thành hai phương pháp mã hoá dữ liệu, đó là *hệ mã hoá khoá đối xứng* và hệ *mã hoá khoá công khai*. Trong đó, *mã hoá khoá công khai* đóng góp một phần lớn không thể thiếu trong việc tạo lập và xác thực chữ ký số. Trước khi chúng ta đi vào phần tìm hiểu hai hệ mã hoá trên thì chúng ta sẽ nghiên cứu qua lý thuyết về hệ mã dòng và hệ mã khối.

### Hệ mã dòng - Stream Cipher

Với các hệ mã dòng *(stream cipher)*, chúng ta sẽ thực hiện xử lý trên từng bit của bản rõ. Điển hình nhất cho một hệ mã dòng đã rất nổi tiếng đó là One Time Pad (OTP), chú ý OTP này khác với One Time Password. Ta có m và khóa k có cùng độ dài (bit), One - Time - Pad được xác định như sau:

E (m, k) = m XOR k = c

D (c, k) = c XOR k = (m XOR k) XOR k = m



Hình 1.2 Mô hình về hệ mã hoá dòng Stream Cipher

Với OTP, khóa k phải đáp ứng đủ 3 điều kiện sau đây:

* Độ dài của khóa phải bằng kích thước bản rõ.
* Khóa phải được chọn hoàn toàn ngẫu nhiên (truly random)
* Và khóa chỉ được sử dụng một lần

Nếu thỏa mãn 3 điều kiện trên, hệ mã OTP sẽ được xem là an toàn tuyệt đối (perfect security) theo định lý của Claude Shannon, tức là kẻ tấn công sẽ không thể biết được thông tin gì của bản rõ m chỉ từ bản mã c.

Bởi vì các hàm mã hóa / giải mã chỉ đơn giản là thực hiện phép toán XOR trên các bit của dữ liệu đầu vào, do đó OTP cho ta tốc độ tính toán rất nhanh.

Vì độ dài khoá và bản rõ bằng nhau, nên chúng ta thường truyền 2 thứ đó chung với nhau một cách bí mật. Đây cũng nhược điểm của OTP. Trong thực tế, người ta thường tạo ngẫu nhiên khoá K có kích thước ngắn hơn độ dài bản rõ, sau đó dùng một hàm tạo số ngẫu nhiên (Pseudo Random Generator - PRG) để tăng độ dài của khóa k đó.

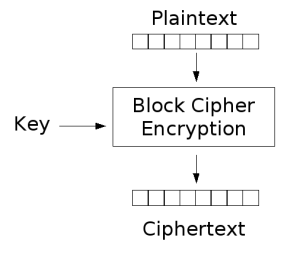
Vì vậy, thực tiễn của OTP được sử dụng như sau:

E'(m, k) = E(m, PRG(k)) = m XOR PRG(k) = c

D'(c, k) = D(c, PRG(k)) = (m XOR PRG(k)) XOR PRG(k) = m, trong đó, k có kích thước nhỏ hơn rất nhiều so với m.

### Hệ mã khối - Block Cipher

Với các hệ mã khối, giống như tên gọi của chúng, là các hệ mã hoá chia dữ liệu đầu vào thành các khối bit có kích thước cố định và thực hiện tính toán và mã hoá trên từng khối đó, kích thước thông thường là 64 hoặc 128 bit. Vì lý do đó, khi đầu vào bản rõ có độ dài không phải là bội số của khối, ta cần phải thực hiện thao tác đệm (padding) sao cho số bit của đầu vào sau khi đệm phải là bội số của khối.



Hình 1.3 Mô hình về hệ mã hoá khối Block Cipher

Các hệ mã khối nổi tiếng đó là DES có kích thước khối là 64 bit (tức là mỗi lần mã hóa một khối bản rõ 64 bit và cho ra khối bản mã 64 bit) và kích thước khóa là 56 bit; AES có kích thước khối là 128 bit và kích thước khóa là 128, 192 hoặc 256 bit.

Thông thường các hệ mã hoá khối chậm hơn so với các hệ mã hoá dòng, nhưng làm việc tốt với những khối dữ liệu đã biết trước kích thước, ví dụ mã hóa file, mã hóa tin nhắn trên giao thức như là HTTP.

### Hệ mã hoá đối xứng

Trong mật mã học, các thuật toán khóa đối xứng *(symmetric - key algorithms)* là một lớp các thuật toán mật mã hóa trong đó các khóa dùng cho việc mật mã hóa và giải mã là một. Khoá của hệ mã hoá này được dùng chung cho cả hai người và được thống nhất với nhau trước mỗi giao dịch.

Hệ mã hoá đối xứng phát triển thành hai giai đoạn, giai đoạn sử dụng các giải thuật mã hoá cổ điển và giai đoạn sử dụng các giải thuật mã hoá hiện đại.

#### Hệ mã hoá cổ điển

**Các tính chất cơ bản của các giải thuật mật mã hoá là**

* Phải có tính bảo mật cao
* Thuật toán phải dễ dàng tiếp cận nhưng vẫn bảo mật, không phụ thuộc vào giải thuật mà chú trọng về khoá
* Có thể áp dụng vào các ứng dụng trên các thiết bị điện tử.

**Một số giải thuật mã hoá cổ điển**

#### Mã thay thế đơn giản (Substitution Cipher)

Là phương pháp mà từng ký tự (hay từng nhóm ký tự) trong thông tin gốc được thay thế bằng một ký tự (hay một nhóm ký tự) khác tạo ra bản mã hoá. Bên nhận khi nhận được bản mờ chỉ cần đảo ngược quá trình thay thế ở trên để có được thông tin gốc ban đầu.

Khoá thường được biểu diễn bằng một chuỗi 26 ký tự. Có 26! (≈ 4.1026) hoán vị (khoá)

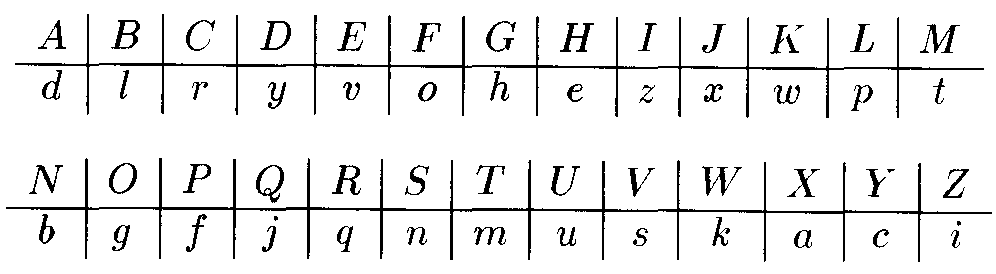
*• Ví dụ 1.2:*

Khoá là chuỗi **dlryvohezxwptbgfjqnmuskaci**.

Dựa vào khoá này chúng ta thay thế ký hiệu A trong thông báo bằng d, ký hiệu B sẽ được thay bằng l, … cứ thế đến khi hết khoá.

Cho một đoạn thông tin gốc là MATMAHOC

Chúng ta tìm ra bản mờ bằng cách thay thế các ký tự tương ứng trong bảng dưới đây



Hình 1.4 Bảng ví dụ sắp xếp thuật toán mã hoá thay thế

Từ đó chúng ta thu được kết quả: **tdmtdegr**

#### Mã thay thế n - gram

Mã thay thế n – gram là dạng mã hoá cao cấp hơn mã thay thế, thay vì thay thế ký tự, người ta thay thế bản gốc cho từng cụm 2 ký tự (diagram) hoặc cụm 3 ký tự (trigram) hoặc thay thế tổng quát cho từng cụm n ký tự (n - gram).

Với bảng chữ cái gồm 26 ký tự tiếng Anh thì phép thay thế n - gram sẽ có khoá là một hoán vị của 26 n - gram khác nhau.

Trong trường hợp diagram thì chúng ta có thể biểu diễn bằng một dãy 2 chiều 26 x 26 trong đó các hàng ngang biểu diễn ký hiệu đầu tiên, các cột dọc biểu diễn ký hiệu thứ hai, nội dung của các ô biểu diễn chuỗi thay thế ứng với các cột và hàng ngang của bảng.

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | … |
| A | EG | RS |  |
| B | BO | SC |  |
| ... |  |  |  |

#### Mã hoán vị bậc d (Permutation Cypher)

Đối với một số nguyên dương d bất kỳ, mã hoán vị bậc d sẽ thực hiện chia thông báo m thành từng khối có chiều dài d. Rồi lấy một hoán vị h của 1, 2,3, …, d và áp dụng h vào mỗi khối.

*• Ví dụ 1.3:* Nếu d = 5 và h = (4 1 3 2 5), hoán vị (1 2 3 4 5) sẽ được thay thế bằng hoán vị mới (4 1 3 2 5).

*• Ví dụ 1.4:* Ta có thông báo

m = JOHN IS A GOOD ACTOR

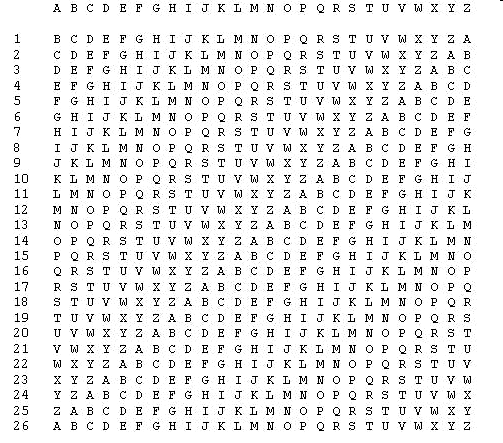
Qua phép mã hoá này m sẽ trở thành chuỗi mật mã c sau:

c = NJHO AI S DGOO OATCR

#### Mã dịch chuyển (Shift Cypher)

• Trong phương pháp Vigenère, khoá bao gồm một chuỗi có d ký tự. Chúng được viết lặp lại bên dưới thông báo và được cộng modulo 26. Các ký tự trắng được giữ nguyên không cộng.

• Nếu d = 1 thì khoá chỉ là một ký tự đơn và được gọi là phương pháp Caesar (được đưa ra sử dụng đầu tiên bởi Julius Caesar).



Hình 1.5 Bảng mã dịch chuyển

*Ví dụ 1.5:*

Từ khoá: CHIFFRE

Mã hoá: VIGENERE

Kết quả thu được dựa trên bảng mã: XPOJSVVG

#### Mã tuyến tính (Affine Cipher)

Phương pháp mã tuyến tính là một dạng mã hoá có dạng như sau:

e(x) = ax + b (mod 26), với a, b

Nếu a = 1 ta có mã dịch chuyển.

Giải mã: Tìm x?

y = ax + b (mod 26)

ax = y - b (mod 26)

x = (y - b) (mod 26).

#### Mã Playfair

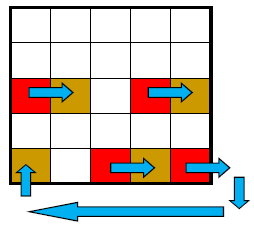
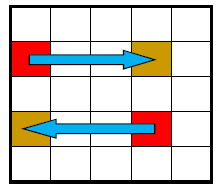
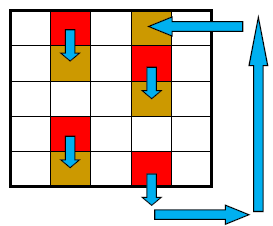
Mã Playfair là một hệ mã hoá đa ký tự (mỗi lần mã hoá 2 ký tự liên tiếp nhau), sử dụng giải thuật dựa trên một ma trận các chữ cái cố định dạng n × n (đối với bảng chữ cái tiếng anh là 26 ký tự thì n = 5 hoặc bằng chữ cái tiếng anh kèm theo 10 ký tự số thì n = 6) được xây dựng từ một khóa (chuỗi các ký tự).

***Xây dựng ma trận khóa:***

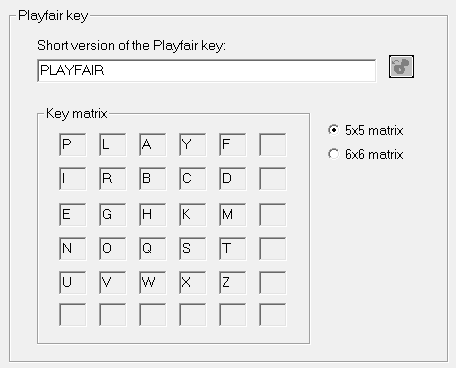
* Chúng ta thực hiện thêm lần lượt các ký tự của khoá vào ma trận theo chiều từ trên xuống, trái sang phải.
* Nếu khoá đã hết mà ma trận chưa đủ các ô, thực hiện thêm các ký tự còn lại trong bảng chữ cái vào ma trận theo thứ tự A – Z (bỏ qua những ký tự đã có)
* Ký tự I và J xem như 1 ký tự và xếp vào cùng 1 ô của ma trận (đối với ma trận n = 5)
* Các ký tự trong ma trận khoá không được trùng nhau.

***Giải thuật mã hóa:***

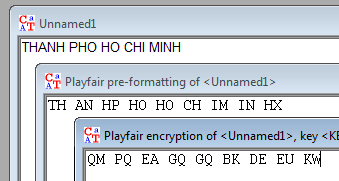
* + - * Mã hóa từng cặp 2 ký tự liên tiếp nhau.
      * Nếu dư 1 ký tự, thêm ký tự “X” vào cuối.
      * Nếu 2 ký tự nằm trên một dòng, thay thế bằng 2 ký tự bên phải của 2 ký tự tương ứng. Nếu đến cột cuối cùng thì ký tự được thay bằng ký tự của cột đầu tiên.
      * Nếu 2 ký tự nằm trên cùng một cột, thay thế bằng 2 ký tự bên dưới tương ứng. Nếu đến hàng cuối cùng thì ký tự được thay bằng ký tự của hàng đầu tiên.
      * Trường hợp còn lại (2 ký tự sẽ tạo thành 2 góc của hình chữ nhật) sẽ được thay thế bằng 2 ký tự tương ứng trên cùng dòng ở hai góc còn lại.



Hình 1.6 Cách hoạt động của playfair



Hình 1.7 Cách xếp ma trận khóa của playfair



Hình 1.8 Kết quả trả về khi dùng playfair

#### Mã Hill

***Giải thuật mã hóa của mã hill:***

• Sử dụng m ký tự liên tiếp của plaintext và thay thế bằng m ký tự trong ciphertext với một phương trình tuyến tính trên các ký tự được gán giá trị lần lượt là A = 01, B = 02, …

Z = 26.

• Chọn ma trận vuông Hill (ma trận H) làm khoá.

• Mã hoá từng chuỗi n ký tự trên plaintext (vector P) với n là kích thước ma trận vuông Hill.

• C = HP mod 26

• P = C mod 26

Ưu điểm của tất cả các phương pháp mã hoá trên là tốc độ mã hoá và giải mã rất nhanh, bù lại nhược điểm là khoá phải được truyền trên kênh truyền đảm bảo an toàn nên chi phí tốn kém, không kịp thời và độ bảo mật kém.

#### Hệ mã hóa hiện đại

Thời đại của hệ mã hoá hiện đại bắt đầu khi tiêu chuẩn mật mã hóa dữ liệu (Data Encryption Standard) - một phương thức mã hoá - được công bố tại Mỹ vào ngày 17/03/1975. Với chiều dài khoá chỉ là 56 bit, DES đã bị chứng minh là không đủ sức chống lại những tấn công kiểu vét cạn (brute force attack - tấn công dùng bạo lực). Có khá là nhiều thuật toán mã hoá khác được đề xuất để thay thế cho DES. Cứ thế, vào năm 2001, hệ mã hoá DES đã chính thức được thay thế bởi hệ mã hoá AES (Advanced Encryption Standard - Tiêu chuẩn mã hóa tiên tiến).

Trước đây, đa số các thuật toán mật mã hóa hiện đại đều chỉ là những thuật toán khóa đối xứng *(symmetric key algorithms),* bắt buộc cả người gửi và người nhận phải dùng chung một khóa, và cả hai người đều phải giữ bí mật và ghi nhớ khóa này cho từng giao dịch, điều này dẫn đến nhiều bất tiện trong việc ghi nhớ và lưu trữ khoá. Chính vì thế, một lần nữa hệ mã hoá hiện đại đã chuyển mình sang một chương mới đó là hệ mã hoá dùng khoá công khai *(hệ mã hoá bất đối xứng).* Về hệ mã hoá bất đối xứng, chúng ta tạo ra hai khoá bí mật và công khai, có quan hệ toán học để dùng trong thuật toán, một dùng để mã hóa và một dùng để giải mã. Người ta chỉ còn cần phải nhớ một khoá bí mật của họ và khoá còn lại sẽ được công khai cho mọi người. Phổ biến nhất trong hệ mật mã này là hệ mã hoá RSA, chúng ta cùng tìm hiểu trong mục tiếp theo.

### Hệ mã hoá bất đối xứng

Hệ mã hóa bất đối xứng *(asymmetric cryptography)* hay còn gọi là hệ mã hóa khoá công khai, là một hệ mã hóa sử dụng một cặp key để mã hóa và giải mã: public key (khóa công khai) dùng để mã hóa và private key (khóa bí mật) để giải mã.

Khi chúng ta sử dụng, bất cứ người gửi nào cũng có thể sử dụng khoá công khai của người nhận để mã hóa bản tin và gửi cho người nhận. Nhưng một điều hiển nhiên là người sở hữu khoá bí mật sẽ giữ nó cho riêng mình, và do đó, chỉ chủ sở hữu nó mới có thể giải mã và đọc được thông điệp.

Thông thường thì cặp khóa được sinh này sẽ cố gắng đảm bảo rằng từ public key rất khó (gần như là không thể) truy ra được private key. Vì vậy, bất cứ kẻ tấn công nào nếu có được public key (điều này khá dễ dàng) cũng không thể có được private key để giải mã.

#### Một vài hệ mã hoá khoá công khai tiêu biểu

#### Hệ mật mã RSA

Đây là hệ mã hoá đã đánh dấu sự tiến bộ vượt bậc trong lĩnh vực mật mã khoá công khai, được ứng dụng nhiều trong thực tiễn. Chúng ta sẽ tìm hiểu sâu hơn về hệ mã hoá này ở mục 1.2.5.

#### Hệ mật mã xếp ba lô Merkle - Hellman

Hệ mã hoá Merkle - Hellman và các hệ liên quan dựa trên tính phức tạp của bài toán tổng hợp các tập con (bài toán này là bài toán NP (nondeterministic polynomial time) đầy đủ - là một lớp khá lớn các bài toán không có giải thuật được biết trong thời gian đa thức). Tuy nhiên cho tới nay tất cả các hệ mật mã xếp ba lô khác nhau đều đã bị chứng tỏ là không bảo mật (ngoại trừ hệ Chor-Rivest).

#### Hệ mật mã McEliece

McEliece là hệ mã hoá dựa trên lý thuyết về ma trận sinh và không gian chiều vector gồm ba quá trình là tạo khoá, mã hoá và giải mã. Hệ mật mã McEliece dựa trên bài toán giải mã cho các mã tuyến tính (cũng là một bài toán NP đầy đủ) và có chưa thể thay thế được cho các hệ mã hoá công khai hiện nay như RSA vì tính chất mã hoá chỉ dựa trên một vài vấn đề trong lý thuyết số.

#### Hệ mật mã Elgamal

Hệ mật mã Elgamal là hệ mật mã khoá công khai do ông Taher Elgamal người Ai Cập đề xuất vào năm 1984. Đặc điểm của hệ mã hoá này là phụ thuộc vào độ phức tạp của bài toán logarit và là biến thể sơ đồ phân phối của giao thức trao đổi khoá Diffie-Hellman.

### Hệ mã hoá RSA

Trong phần này, chúng ta sẽ tìm hiểu về *hệ mã hoá RSA* - một hệ mã hoá đóng góp vai trò không hề nhỏ trong việc tạo chữ ký số. Để nắm bắt, giải được bài toán RSA, chúng ta cần phải hiểu biết nhiều công thức toán học, trong đó quan trọng nhất là phải nắm vững các lý thuyết về toán học sau đây.

#### Định nghĩa số nguyên tố

Số nguyên tố là các số tự nhiên chỉ chia hết cho 1 và chính nó. Hay nói cách khác, số nguyên tố là một số tự nhiên lớn hơn 1, nếu như ngoài bản thân nó và 1 ra, nó không chia hết cho số nào khác nữa thì nó là số nguyên tố. Có một lưu ý là số 0 và 1 không được coi là số nguyên tố.

Chúng ta có một số ví dụ về số nguyên tố như: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29,…

#### Định lý Euler

Định lý Euler là định lý cơ bản trong các hệ thống mã hoá RSA, tuy nhiên được cho là không đủ và không thật sự cần thiết đối với việc kiểm tra tính hợp lệ trong RSA. Nội dung của định lý như sau:

Cho p là một số nguyên tố:

(1) Định lý Euler: Nếu a ∈ Zn\* thì (mod n).

(2) Nếu n là tích của các số nguyên khác nhau và nếu r ≡ s (mod Φ(n)) thì (mod n) đối với mọi số nguyên a. Nói một cách khác khi làm việc với modulo n thì các số mũ có thể được rút gọn theo modulo Φ(n).

#### Định lý Euclid mở rộng

Định lý Euclid mở rộng có nhiệm vụ giải phương trình vô định nguyên có dạng

ax + by = c

Nếu cho trước 2 số nguyên x, y thì tồn tại hai số nguyên a,b thoả

a.x + b.y = ƯCLN(x, y) = 1.

Nếu ƯCLN(x, y)=1 thì x, y là số nguyên tố cùng nhau (ƯCLN = 1)

*Ví dụ 1.6:*

Cho khoá công khai e = 7, Φ = 160, n = 187, tính d?

Ta có: e.d = 1 mod 160

* Phương trình xuôi: 160 x + 7 y = 1

⬄ 160 = 7(10) + 90

⬄ 7 = 90(0) +7

⬄ 90 = 7(10) + 20

⬄ 7 = 20(0) + 7

⬄ 20 = 7(2) + 6

⬄ 7 = 6(1) + 1 => dừng

* Phương trình đảo:

1 = 7 - 6(1)

1 = 7 - 1(20 - 7(2))

1 = 3(7) - 1(20) (xem 7 là x)(x + 2x = 3x)

1 = 3(7) - 1(90 - 7(10))

1 = 13(7) - 1(90)

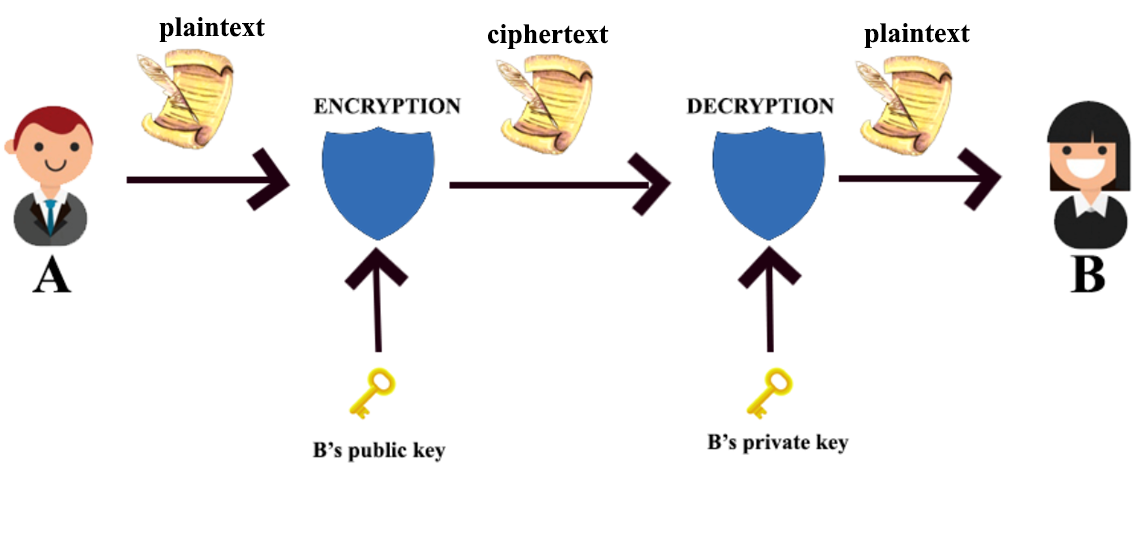
1 = 13(7) - 1(160 - 7(10))

1 = 23(7) - 1(160) (đề bài cho e = 7, Φ = 160)

Từ đó chúng ta suy ra khoá bí mật d = 23

#### Mã hoá RSA - Rivest Shamir Adleman

RSA là một hệ mã hóa bất đối xứng mang đầy đủ các tính chất, ứng dụng của hệ mật mã này. Cha đẻ của hệ mã hoá RSA là Ron Rivest, Adi Shamir và Leonard Adleman (tên của nó cũng chính là tên viết tắt của 3 tác giả này) và được ứng dụng đặc biệt vào công tác mã hoá thông tin và chữ ký điện tử (chữ ký số). Đây là hệ mã hoá đầu tiên phù hợp trong việc tạo chữ ký số đồng thời với việc mã hoá và đang sử dụng phổ biến trong giao dịch điện tử, được cho là đảm bảo an toàn với điều kiện độ dài khoá đủ lớn. RSA được xây dựng dựa trên độ khó của bài toán phân tích thừa số nguyên tố (bài toán RSA) và định lý Euclid mở rộng. Trong hệ mã hóa này, khoá công khai có thể chia sẻ công khai cho tất cả mọi người, còn khoá bí mật sẽ được người chủ sở hữu giữ kín, vì thế không ai khác có thể đọc được dữ liệu đã được mã hoá từ khoá công khai trên ngoại trừ dùng khoá bí mật để giải mã. Hoạt động của RSA dựa trên ba bước chính: sinh khóa, mã hóa và giải mã.



Hình 1.9 Quy trình mã hoá RSA

#### *Sinh khóa*

Mấu chốt cơ bản của việc sinh khóa trong RSA là tìm được bộ 3 số tự nhiên e, d và n sao cho: và một điểm không thể bỏ qua là cần bảo mật cho d sao cho dù biết e, n hay thậm chí cả m cũng không thể tìm ra d được.

Cụ thể, khóa của RSA được sinh như sau:

1. Tạo 2 số nguyên tố lớn ngẫu nhiên và khác nhau p và q, p và q có độ lớn xấp xỉ nhau (số nguyên tố yêu cầu tối thiểu 10 chữ số để đảm bảo tính bảo mật)
2. Tính n = p \* q và Φ(n) = (p −1) \* (q −1).
3. Chọn một số nguyên ngẫu nhiên e, 1 < e < Φ, sao cho ƯCLN(e, Φ) = 1.
4. Sử dụng thuật toán Euclide mở rộng để tính một số nguyên d duy nhất, 1 < d < Φ thoả mãn e.d ≡ 1(mod Φ).
5. Khoá công khai là cặp số (n, e). Khoá riêng bí mật là d.

Trong đó các số nguyên d và e trong thuật toán khoá RSA được gọi là số mũ mã hoá và số mũ giải mã. Số n được gọi là số modulus. Chúng ta cần giữ private key thật cẩn thận cũng như các số nguyên tố p và q vì từ đó có thể tính toán các khóa rất dễ dàng.

Mức độ bảo mật của RSA phụ thuộc rất lớn vào khả năng phân tích thừa số nguyên tố của các số lớn. Bởi vì chúng ta cung cấp public một cách rộng rãi, nếu việc phân tích thừa số nguyên tố đơn giản, thì việc bị lộ private là không thể tránh khỏi.

Vì vậy, khi sinh khóa, chúng ta cần chọn các số nguyên tố p và q một cách ngẫu nhiên. Bản thân hai số nguyên tố này cũng rất lớn, và để việc phân tích thừa số nguyên tố khó khăn hơn, hai số nguyên tố này sẽ không có cùng độ dài. Trong tương lai gần, có lẽ vẫn chưa có một phương pháp hiệu quả nào cho phép thực hiện điều này với các máy tính cá nhân.

*Ví dụ 1.7:*

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

Lấy:

* + p = 29: Số nguyên thứ nhất (giữ bí mật hoặc phải huỷ sau khi tạo khoá)
  + q = 71: Số nguyên thứ hai (giữ bí mật hoặc phải huỷ sau khi tạo khoá)
  + Tính n = p.q = 2059 và Φ(n) = (p −1)(q −1) = 1960
  + Chọn ngẫu nhiên e sao cho ước chung lớn nhất (e, Φ) = 1, ví dụ lấy e = 3
  + Dựa vào định lý Euclid mở rộng, Φ và khoá công khai e ở trên, chúng ta tìm ra khoá bí mật d = 1307

#### Mã hóa và giải mã

Trong phần này, chúng ta sẽ tìm hiểu cách mã hóa với public key (n, e) và giải mã với private key (n, d).

*Ví dụ 1.8:*

Bob muốn mã hoá và gửi thông báo m cho Alice.

Bob cần thực hiện:

1. Thu nhận khoá công khai (n, e) của Alice.
2. Biểu diễn bản tin dưới dạng một số nguyên m trong khoảng [0 , n - 1]
3. Tính c = mod n.
4. Gửi bản mã c cho Alice.

Để khôi phục bản rõ m từ c, Alice phải thực hiện phép tính sau bằng cách dùng khoá riêng m = mod n

Chứng minh hoạt động giải mã:

Vì e.d = 1 (mod Φ) nên luôn tồn tại một số nguyên k sao cho

e.d = 1 + k Φ

Bây giờ nếu (m, p) = 1 theo định lý Fermat ta có:

= 1 (mod p).

Lũy thừa cả hai vế của đồng dư thức trên với số mũ k (q - 1) và rồi nhân cả hai vế với m ta có:

Mặt khác nếu ƯCLN (m, p) = p thì đồng dư thức cuối cùng ở trên vẫn đúng vì mỗi vế đều đồng dư với 0 mod p. Bởi vậy, trong mọi trường hợp ta đều có:

Bằng lập luận tương tự ta lại có: = m (mod p)

Cuối cùng vì p và q là các số nguyên tố khác nhau nên = m (mod n) và bởi vậy .

*Ví dụ 1.9:* Tạo khóa

* Alice chọn các số nguyên tố p = 2357, q = 2551 và tính n = p.q = 6012707 và Φ = (p - 1)(q - 1)= 6007800.
* Alice chọn e = 3674911 và dùng thuật toán Euclide mở rộng để tìm được d = 422191 thỏa mãn ed = 1(mod Φ). Khóa công khai của Alice là cặp số (n = 6012707, e = 3674911), khóa bí mật của Alice là d = 422191
* Để mã hóa thông báo m = 5234673, Bob sử dụng thuật toán lấy lũy thừa theo modulo để tính c = mod 6012707 = 3650502 rồi gửi c cho Alice.
* Để giải mã bản mã c, Alice tính mod n = mod 6012707 = m

Cả hai phép tính trên đều có thể được thực hiện hiệu quả nhờ giải thuật bình phương và nhân. Vấn đề cốt lõi của hệ mã hoá RSA đó là việc chọn số nguyên tố p, q đủ lớn để đảm bảo an toàn cho bản mã, nếu để kẻ thám mã mà biết được số nguyên tố p, q thì dễ dàng tính được khoá bí mật d từ khoá công khai (e, n) do đó bản mã sẽ bị lộ, bởi vì với sự phát triển của công nghệ, các siêu máy tính xuất hiện ngày càng nhiều. Cùng với chúng ta máy tính lượng tử cho phép tính toán với tốc độ cao hơn rất nhiều có thể sẽ phá vỡ sự bảo mật của RSA.

# HÀM BĂM (HASH)

## Sơ lược về hàm băm

Các hàm băm đóng vai trò cơ bản trong mật mã hiện đại. Hàm băm sẽ tạo ra một đầu ra từ bản tin đầu vào. Đầu ra này được định nghĩa là mã băm (mã hash, kết quả băm, giá trị băm). Ý tưởng cơ bản của việc sử dụng các hàm băm trong mật mã là sử dụng chúng như một ảnh biểu diễn rút gọn (đôi khi còn được gọi là vết, dấu tay số hay tóm lược thông báo) của một xâu vào và có thể được dùng như thể nó chính là xâu vào đó.

Một lớp các hàm băm được gọi là các mã xác thực thông báo (MAC - Message Authentication Codes) sẽ cho phép xác thực thông báo bằng kỹ thuật đối xứng (mật mã cổ điển). Trong chương này chúng ta sẽ tìm hiểu về hàm băm và các ứng dụng của hàm băm trong đời sống chúng ta.

## Khái niệm hàm băm

*Hàm băm là gì?*

Là quá trình biến đổi dữ liệu có độ dài (bit) bất kỳ thành một chuỗi đầu ra có độ dài (bit) cố định tùy theo kiểu hàm băm được sử dụng.

*Ví dụ 2.1:* Chúng ta tải một file mp3 từ trên mạng về, sau đó cho nó chạy qua hàm băm SHA-256 sẽ cho ra kết quả là một chuỗi 256 bit, hoặc nếu chúng ta cho nó chạy qua hàm băm SHA-512 sẽ cho ra một chuỗi 512 bit, tuỳ theo loại hàm băm chúng ta sử dụng.

## Tính chất của hàm băm

*Hàm băm mang những tính chất đặc trưng như sau:*

* **Không thể đảo ngược:** chúng ta không thể truy xuất dữ liệu ban đầu từ các giá trị đã được băm.
* **Tính toàn vẹn của dữ liệu:** vì chỉ có một giá trị băm cho một dữ liệu đầu vào nên nếu chỉ cần một ký tự trong dữ liệu vào bị thay đổi thì giá trị băm sẽ thanh đổi theo.Từ đó ta biết được dữ liệu đó có toàn vẹn hay không.
* **Giá trị băm là duy nhất:** hai dữ liệu đầu vào khác nhau thì cho ra hai giá trị băm khác nhau.

## Thuộc tính cần thiết

*Một hàm băm được xem là mạnh và tốt thì cần phải có những thuộc tính sau đây*

1. **Tính xác định:** Dữ liệu đầu vào bất kể ở kích thước nào đều chỉ có ra giá trị băm có độ dài nhất quán.
2. **Tốc độ tính toán:** Tốc độ băm rất quan trọng và thay đổi tùy theo trường hợp sử dụng. Ví dụ như đôi lúc chúng ta sẽ cần một hàm băm nhanh và đôi lúc chúng ta cần hàm băm chậm.
3. **Tính chống đảo ngược:** Hàm băm không được phép đảo ngược ,tức là nó chỉ có một chiều chúng ta sẽ không thể khôi phục được giá trị ban đầu từ các giá trị băm.

## Cách hoạt động của hàm băm

* Đầu tiên, hàm băm sẽ chia dữ liệu vào (Data input) thành các phần bằng nhau (về kích thước).
* Tiếp theo, thuật toán băm sẽ được thực hiện trên từng phần dữ liệu vào riêng biệt.
* Sau đó, giá trị băm của phần đầu tiên sẽ được thêm vào phần dữ liệu thứ hai và được áp dụng thuật toán băm đề cho ra giá trị băm của phần thứ hai .
* Tiếp tục như vậy cho đến khi không còn phần dữ liệu nào nữa thì chúng ta sẽ được kết quả của một hàm băm.

Hình bên dưới mô tả chi tiếp quá trình này:

**Dữ liệu vào**

**Giá trị băm**

**Khối 1 (K1)**

**Giá trị băm**

**G(K1)**

**G(K1)+Khối 2­ (K2)**

**Giá trị băm**

**G(K2)**

**G(K2)+Khối n­ (Kn)**

**Giá trị băm**

**G(Kn)**

**Kết quả băm**

Băm

Băm

Băm

Hình 2.1 Sơ đồ hoạt động hàm băm

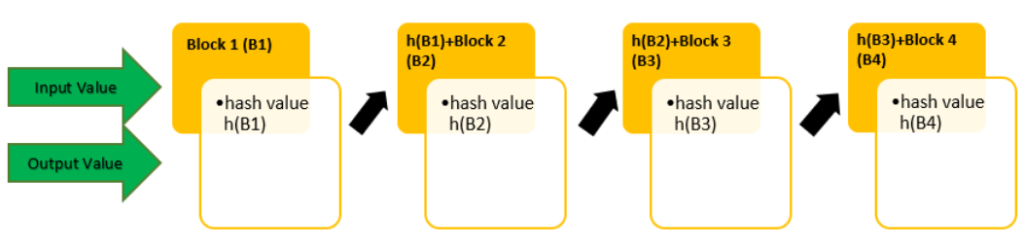
Trong đó:

K: Phần dữ liệu đã được chia.

Kn: Phần dữ liệu đã được chia cuối cùng.

G(K): Giá trị băm của phần dữ liệu K.

Ví dụ 2.2:



Hình 2.2 Ví dụ về hàm băm

Ở hình trên giá trị đầu vào được chia làm 4 phần

* Phần dữ liệu đầu tiên (B1) được băm thành giá trị băm h(B1).
* Giá trị băm h(B1) kết hợp với phần dư liệu thứ hai (B2) và được băm thành h(B2).
* Giá trị băm h(B2) kết hợp với phần dư liệu thứ ba (B3) và được băm thành h(B3).
* Giá trị băm h(B3) kết hợp với phần dư liệu thứ tư (B4) và được băm thành h(B4).

🡺Giá trị băm cuối cùng h(B4) cũng là kết quả cuối cùng của hàm băm.

Hàm băm được thực hiện như vậy là để có thể phát hiện bất khi giá trị giả mạo nào vì nếu có bất cứ giá trị nào bị thay đổi nó sẽ làm thay đổi cả giá trị băm.

## Các loại hàm băm phổ biến

* SHA-1, SHA-2.
* MD5.
* RIPEMD -160.
* Whirlpool.

Có nhiều hàm băm khác nhau nhưng trong bài này chúng ta sẽ nghiên cứu sơ lược về MD5 và tập trung vào thuật toán của hàm băm SHA.

### MD5 (Message - Digest Algorithm 5)

#### Tổng quan về hàm băm MD5

MD5 được phát hành vào năm 1992 , là một trong 3 thuật toán được Ronald Rivest tạo ra,trong số đó phiên bản đầu tiên được phát hành năm 1989 là MD2 và sau đó, năm 1990 MD4 được phát hành để thay đế cho MD2.

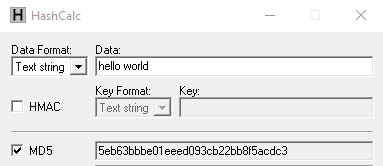
MD5 được phát triên cho các hệ máy 32 bit và nó không nhanh như MD4 nhưng được cho là phiên bản MD an toàn nhất.

Tuy nhiên hiện tại nó được cho là không an toàn vì lỗi hổng bảo mật.

#### Chi tiết về MD5

Mã MD5 có độ dài 128 bit và có giá trị băm là chuỗi hệ thập lục phân 32 ký tự

Ví dụ 2.3: Sử dụng công cụ HashCalc để kiểm tra. Cho dữ liệu đầu vào là “*hello world*” thì giá trị băm đầu ra sẽ là “*5eb63bbbe01eeed093cb22bb8f5acdc3*”



Hình 2.3 Ví dụ về hàm băm MD5 bằng công cụ HashCalc

Thay vì kiểm tra và xác nhận giữa 2 file dữ liệu không có bất kỳ thay đổi nào bằng cách so sánh các dữ liệu ban đầu, MD5 sẽ thực hiện điều này bằng cách tổng kiểm giá trị trên 2 file dữ liệu, sau đó so sánh để xác minh tính toàn vẹn của các file.

### SHA (Secure Hashing Algorithm)

Là một thuật toán băm an toàn. Họ hàm băm SHA gồm: SHA-0, SHA-1, SHA-2, SHA-3

* SHA-0: Ít được sử dụng trên thực tế;
* SHA-1: Tương tự SHA0, nhưng đã khắc phục một số lỗi, chuỗi đầu ra của SHA-1 có kích thước 160 bít và thương được biểu diễn thành 40 số hexa
* SHA-2: Khắc phục lỗi của SHA1 và có nhiều thay đổi. Kích thước chuỗi đầu ra có thể là 224, 256, 384 và 512 bít;
* SHA-3: Cho phép chuỗi đầu ra có kích thước không cố định.

Sau đây chúng ta sẽ tiến hành tìm hiểu về các loại hàm băm trên:

#### SHA-0:

Là một thuật toán băm có độ dài giá trị băm là 160 bit là phiên bản đầu tiên của SHA. Giá trị băm của nó có độ dài 40 ký tự Hex tương đương với 20 byte. Được phát hành đầu tiên vào năm 1993 nhưng lại ngay lập tức bị thay thế bởi SHA-1 do các lỗi bảo mật của nó.

#### SHA-1:

Là phiên bản thứ hai của SHA. Nó có độ dài giá trị băm tương tự SHA-0 nhưng đã khắc phục các điểm yếu về bảo mật của phiên bản trước. Tuy nhiên nó cũng không thoát khỏi số phận như phiên bản tiền nhiệm, vào năm 2005 nó đã cũng được phát hiện là không an toàn. Mặc dù vậy nó đã trở thành tiền đề để phát triển phiên bản sau.

#### SHA-2:

Là tiêu chuẩn hàm băm mật mã của hầu hết phần mềm và phần cứng hiện nay.

SHA-2 bao gồm: SHA-224, SHA-256, SHA-284, SHA-512.

SHA-2 được coi là hàm băm an toàn nhất hiện tại vì:

* + Không thể tìm ra dữ liệu từ giá trị băm bởi giải thuật SHA.
  + Chỉ cần một sự thay đổi nhỏ cũng có thể khiến giá trị băm thay đổi khác hoàn toàn.

#### SHA-3:

Hàm keccak hay còn được gọi là SHA-3 được phát triển vào năm 2007 bởi Guido Bertoni, Joan Daemen, Michaël Peeters và Gilles Van Assche, nhằm khắc phục những sai sót tiềm ẩn của SHA-1, SHA-2.

Được kế thừa từ SHA-2 nên SHA-3 cũng có các phiên bản SHA3-224, SHA3-256, SHA3-384, SHA3-512 với kích thước đầu ra tương ứng.

SHA-3 được thiết kế rất hiệu quả về phần cứng nhưng chậm về phần mềm. Nó mất gấp đôi thời gian so với phiên bản trước là SHA-2 để chạy trong phần mềm nhưng chỉ cần một phần tư thời gian để chạy trên các phần cứng.

## Ứng dụng của hàm băm

Hàm băm đã không quá xa lạ với chúng ta bởi vì nó được ứng dụng rất rộng rãi trong đời sống. Dưới đây là một vài ứng dụng quan trọng của hàm băm

#### Kiểm tra sự toàn vẹn của tệp tin

Như đặc điểm của hàm băm, cùng một giá trị sẽ cho ra cùng một giá trị băm. Vậy nên ta có thể đối chiếu tệp tin ta tải trên mạng về với bản gốc bằng cách so sánh giá trị băm của chúng với nhau. Nếu chúng có chung giá trị băm tức là tệp tin của chúng ta trùng với bản gốc, nếu không tệp tin của chúng ta đã bị sửa đổi hoặc bị hỏng. Một số trường hợp, tệp tin của chúng ta tải về bị can thiệp bởi bên thứ ba trước khi đến thiết bị của chúng ta và chúng có thể cài mã độc vào tệp tin đó. Việc kiểm tra giá trị băm giúp đảm bảo tệp tin của chúng ta an toàn.

Cũng nhờ ứng dụng này nên hàm băm cũng được sử dụng trong các chứng chỉ SSL / TLS để kiểm tra dữ liệu đầu vào trong quá trình truyền tin.

#### Xác minh mật khẩu

Có một điều rất hay chúng ta nên biết, trong những thiết kế cơ sở dữ liệu hiện đại, thứ lưu trong đó không phải mật khẩu của chúng ta dưới dạng văn bản đơn thuần mà là giá trị hash của chúng. Khi chúng ta nhập mật khẩu, mật khẩu của chúng ta sẽ được chạy qua hàm hash, sau đó sẽ được so sánh với giá trị băm trong cơ sở dữ liệu để quyết định chúng ta có được chứng thực để sử dụng dịch vụ không. Điều này làm giảm đáng kể thiệt hại khi cơ sở dữ liệu bị tấn công, khi những gì bị lộ ra ngoài là những giá trị băm chứ không phải mật khẩu của chúng ta. Để an toàn hơn, hệ thống còn thêm giá trị muối (salt) vào mật khẩu gốc của chúng ta, rồi cho chạy qua hàm băm, sau đó mới lưu vào cơ sở dữ liệu. Vậy nên kể cả khi giá trị băm của mật khẩu chúng ta bị lộ và bị giải mã, kẻ tấn công vẫn chưa thể có được mật khẩu thực sự của chúng ta do nó đã được thêm vào giá trị "salt".

#### Chia nhỏ công việc

Hàm băm được sử dụng để xử lý một lượng lớn dữ liệu, thay vì phải xử lý một lượng lớn dữ liệu lớn với nhiều độ dài khác nhau thì ta có thể xử lý các chuỗi có độ dài nhỏ bằng nhau là các giá trị băm.

#### Ứng dụng vào chữ ký số (digital signatures).

Hàm băm kết hợp với hệ mã hoá khoá công khai RSA để tạo ra và xác thực chữ ký số. Bằng các kỹ thuật băm, chúng ta băm thông điệp thành các đoạn nhỏ rồi dùng mã hoá RSA để tạo ra một khối dữ liệu mang thông tin xác thực của chủ sở hữu. Đây cũng là ứng dụng quan trọng nhất của hàm băm mà chúng ta sử dụng trong bài đồ án này.

# LÝ THUYẾT VỀ CHỮ KÝ SỐ

## Giới thiệu về chữ ký số

Từ thời xa xưa, ông cha chúng ta thường sử dụng con dấu của cá nhân hoặc những chữ ký tay để sử dụng cho các loại văn bản giấy tờ thực, thánh chỉ, các hiệp ước, hiệp định hoặc các bản cam kết. Những chữ ký tay này mang vai trò quyết định giá trị của văn bản và đại diện cho trách nhiệm của các bên tham gia ký kết văn bản.

Nhưng vào thời đại phát triển hiện nay, các công ty doanh nghiệp sử dụng dịch vụ kinh doanh giao dịch điện tử ngày càng nhiều, dẫn đến thanh toán, xác thực online trở nên cực kỳ quan trọng . Việc các công ty di chuyển hợp đồng lẫn nhân sự đến gặp tận nơi để ký kết theo cách thủ công thời xưa khá là bất tiện, tốn khá nhiều thời gian, công sức, nhân lực. Đặc biệt dưới sự bùng phát của dịch bệnh COVID-19 càng khiến việc này trở nên khó khăn và nguy hiểm. Chưa kể các loại máy móc hiện đại phát triển, chữ ký tay có thể bị sao chép và sử dụng khắp nơi, đơn giản nhất là việc học sinh giả chữ ký phụ huynh để ký vào bản kiểm điểm.

Chính vì những bất cập trên, chữ ký điện tử ra đời, mở ra một kỷ nguyên mới về thời đại công nghệ chữ ký số.

## Khái niệm chữ ký số

*Vậy chữ ký số là gì?*

Chữ ký số (digital signature) là một tập con của chữ ký điện tử (chữ ký điện tử là một khái niệm bao hàm chữ ký số), được biểu thị ở dạng một đoạn mật mã, tạo ra bằng các phương pháp mã hoá bí mật dùng để đính kèm (ký) vào các văn bản gốc, hoá đơn, tài liệu trong các giao dịch điện tử nhằm xác minh, chứng thực tác giả (chủ sở hữu) của dữ liệu đó và giúp người nhận kiểm tra được tính toàn vẹn nội dung của dữ liệu.

## Các ưu điểm của chữ ký số

*Chữ ký số có các đặc tính nổi bật dưới đây:*

* **Tính xác thực**

Mỗi chữ ký số được xây dựng bằng các hệ thống mã hoá bất đối xứng và hàm băm. Hệ mã hoá bất đối xứng tạo ra 2 khoá riêng biệt: 1 khoá công khai (public key) được share cho tất cả mọi người biết, có công dụng dùng để xác minh văn bản và chữ ký đi kèm là của đúng người chủ sở hữu gửi; 1 khoá bí mật (private key) được giữ kín bí mật chỉ có người gửi biết. Quy trình xác thực chữ ký số sẽ phân tích và so sánh 2 chuỗi mã hoá của 2 chữ ký (được tạo ra bằng hàm băm) để xác thực chủ sở hữu của văn bản đã được ký đó là người gửi chứ không phải ai khác. Cũng khó có thể tránh khỏi việc chữ ký bị giả mạo, thế nhưng nếu chúng ta quản lý tốt chữ ký số thì tỉ lệ giả mạo sẽ rất thấp.

Việc xác thực người gửi cực kỳ quan trọng trong cuộc sống, đặc biệt những giao dịch liên quan đến tiền và tài sản.

*Ví dụ 3.1:* Trong một cuộc giao dịch điện tử, người A (người bán) gửi cho người B (người mua) số tài khoản ngân hàng của mình thông qua tin nhắn để B chuyển tiền qua số tài khoản đó. Nhưng có một kẻ lừa đảo C nào đó bắt được tin nhắn đó và sửa đổi số tài khoản của A trong tin nhắn đó thành số tài khoản của C và gửi lại tin nhắn đó cho B với danh nghĩa là A. Cuối cùng B nghĩ rằng số tài khoản đó là của A nên chuyển tiền vào đó, đây là một ví dụ cơ bản của hình thức tấn công the - man - in - the - middle nếu như không xác thực kỹ càng chủ sở hữu của tin nhắn.

* **Tính toàn vẹn và bảo mật**

Giả sử A cần gửi cho B một thông điệp mật kèm chữ ký điện tử, A sẽ sử dụng khoá công khai của B để mã hoá thông điệp rồi dùng khoá cá nhân của mình để mã hoá chữ ký, sau đó gửi cả thông điệp lẫn chữ ký cho B. B sẽ dùng khoá công khai của A để giải mã chữ ký, rồi dùng khoá cá nhân của mình để giải mã thông điệp của A. Văn bản được bảo vệ và xác thực qua nhiều lớp, đảm bảo và xác nhận nội dung của văn bản được ký không bị sửa đổi, đem lại sự tin tưởng cho các bên tham gia giao dịch.

Việc tạo chữ ký và kiểm chứng chữ ký thường được thực hiện nhờ hàm băm nên các mã hash tạo ra không trùng nhau và chỉ được sử dụng duy nhất trong 1 giao dịch, ngăn việc sử dụng lại chữ ký số nhằm mục đích khác.

* **Tính trách nhiệm và duy nhất**

Vì chỉ có người gửi mới có khoá bí mật để tạo chữ ký số, giảm thiểu việc một bên giao dịch trốn tránh và từ chối văn bản đã gửi là của mình, chữ ký số bảo vệ quyền lợi của bên hợp tác còn lại và cũng có thể được dùng làm chứng cứ để bên thứ ba giải quyết.

* **Tiết kiệm thời gian, công sức**

Thay thế cho việc vận chuyển văn bản, thực hiện ký tay theo cách thủ công, chữ ký số giúp giảm thiểu thời gian, chi phí hành chính và phương tiện để các bên giao dịch có thể ký số một cách nhanh chóng, chính xác và an toàn, mọi lúc mọi nơi, không cần chuẩn bị hồ sơ bản cứng lẫn sắp xếp nhân sự để thực hiện ký kết.

* **Tính linh động**

Dựa theo nhu cầu sử dụng và phù hợp với mọi đối tượng sử dụng từ văn phòng đến cá nhân, từ công nhân đến doanh nhân, chữ ký số chia thành hai dạng:

+ Sử dụng USB Token chứa chữ ký số: Loại chữ ký số thông dụng nhất dành cho các công ty văn phòng sử dụng với lưu lượng vừa phải, gồm một chiếc USB token có chứa các thông tin của khách hàng, lưu giữ khoá bí mật và chứng thư số của thuê bao, có khả năng lưu trữ lớn tốc độ cao. Đi kèm theo token là một mã pin duy nhất gồm 10 ký tự cố định được tạo ra riêng dành cho mỗi doanh nghiệp.

+ Chữ ký số không cần sử dụng USB token: Thường gọi là chữ ký số online, sử dụng công nghệ đám mây (cloud - based) giúp việc ký văn bản trở nên đơn giản, tiết kiệm chi phí và có thể thực hiện mọi lúc mọi nơi.

## Quy trình tạo và xác thực chữ ký số

Chữ ký điện tử được ứng dụng và thừa kế từ các thành tựu khoa học về mã hoá khoá công khai và hàm băm (hash). Trong chương này ở các mục sau chúng ta sẽ tìm hiểu kỹ hơn về các thức tạo và xác thực chữ ký số.

Quy trình gồm ba bước sau:

* Tạo khoá
* Tạo chữ ký và ký vào thông điệp
* Giải mã

### Thuật toán tạo khoá

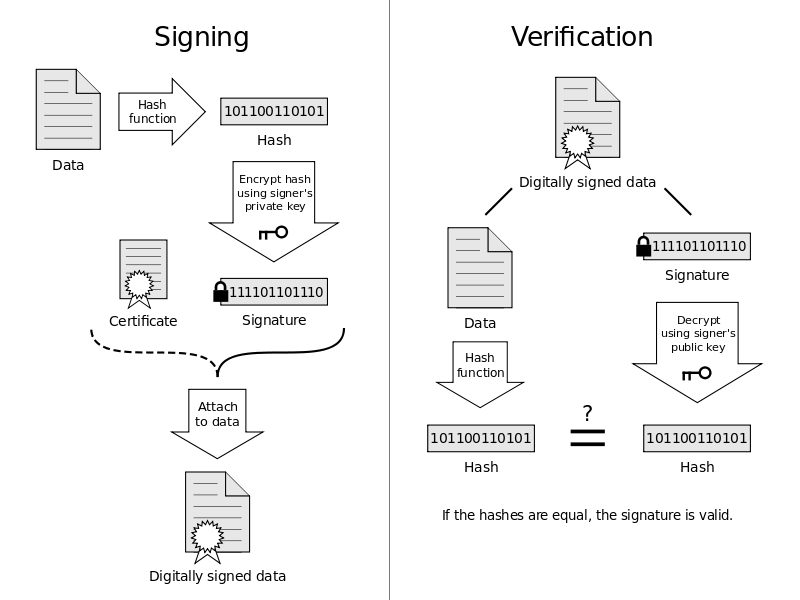
Sử dụng thuật toán của hệ mã hoá khoá công khai để khởi tạo một cặp khoá bao gồm khoá công khai và khoá bí mật. Có nhiều thuật toán để tạo khoá có thể ứng dụng vào chữ ký số như: Chữ ký số ABA, chữ ký số Elgamal, chữ ký số RSA, … Ở bài nghiên cứu này chúng ta sử dụng thuật toán mã hoá khoá công khai RSA để tạo và xác thực chữ ký số.

### Tạo chữ ký và ký vào thông điệp

* Đầu tiên chúng ta cần có một file cần được ký xác nhận (hoá đơn, hợp đồng, các văn bản nhận giao dịch,…), có thể ở dạng \*.doc, \*.pdf, \*.mp3, … Trong bài nghiên cứu này chúng ta sẽ tiến hành thực hiện ký chữ ký số và xác thực nó trên file văn bản \*.doc.
* Dùng giải thuật hàm băm để thay đổi thông điệp cần truyền đi để được một message digest (MD5 thu được digest có chiều dài 128 bit hoặc SHA thu được digest 160 bit).
* Sử dụng khóa private key của người gửi để mã hóa message digest thu được ở bước trên. Bước này thường dùng giải thuật RSA. Kết quả thu được gọi là digital signature của thông điệp ban đầu.
* Gộp digital signature vào thông điệp ban đầu (“ký nhận” vào thông điệp). Sau đó, mọi sự thay đổi trên message sẽ bị phát hiện. Việc ký nhận này đảm bảo người nhận tin tưởng thông điệp này xuất phát từ người gửi chứ không phải là ai khác.

### Kiểm tra và xác thực chữ ký số

* Dùng public key của người gửi (khóa này được thông báo đến mọi người) để giải mã chữ ký số của thông điệp.
* Dùng giải thuật (MD5 hoặc SHA) băm thông điệp nhận được.
* So sánh kết quả thu được ở bước 1 và 2. Nếu trùng nhau, ta kết luận thông điệp này không bị thay đổi trong quá trình truyền và thông điệp này là của người gửi.



Hình 3.1 Quy trình tạo và xác thực chữ ký số

*Ví dụ 3.2:*

Chúng ta có một file văn bản \*.doc cần được ký và chuyển giao qua cho bên đối tác.

*Tạo khoá*

* Đầu tiên chúng ta sử dụng thuật toán RSA để tạo khoá công khai và khoá bí mật, lấy Ví dụ 1.7 ở mục 1.2.5.3 Mã hoá RSA, chúng ta tạo ra hai cặp khoá bí mật (n, d) = (2059, 1307) và khoá công khai (n, e) = (2059, e)

*Tạo chữ ký và ký*

* Kế tiếp chúng ta sử dụng hàm băm, ở ví dụ này chúng ta dùng hàm băm MD5, băm file văn bản và tạo ra một mã hash code hệ thập lục phân 32 ký tự, ký hiệu là m.
* Sau đó dùng khoá bí mật (2059, 1307) , tạo ra bản mã hoá

c = mod 2059 và gửi cho bên đối tác bản mã c đính kèm trong file \*.doc

*Giải mã và xác thực*

* Đối tác sau khi nhận được file \*.doc và bản mã c, họ thực hiện giảm mã c để tìm được mã hash code:

m = mod 2059

* Song song đó đối tác thực hiện băm file \*.doc ra và tạo được một mã hash code h
* Đem so sánh m và mã hash code h vừa băm, nếu nó trùng nhau chứng tỏ file văn bản \*.doc này là do chính chủ sở hữu gửi và thông điệp được bảo toàn không bị thay đổi.

Ví dụ trên cũng đã mô tả quy trình cơ bản được áp dụng trong việc xây dựng ứng dụng tạo và xác thực chữ ký số.

# MỘT SỐ QUY ĐỊNH, THÔNG TƯ VỀ CHỮ KÝ SỐ

Trong chương này chúng ta sẽ đề cập đến những quy định, Nghị Định phổ biến nhất về chữ ký số trong các văn bản luật giao dịch điện tử được ban hành.

## Quy định về ký số, kiểm tra chữ ký số trên văn bản điện tử trong cơ quan nhà nước việt nam

Các quy định về chữ ký số được quy định rõ ràng trong Nghị Định Số: 130/2018/NĐ - CP quy định chi tiết thi hành luật giao dịch điện tử về chữ ký số và dịch vụ chứng thực chữ ký số.

### Quy định về giá trị pháp lý của chữ ký số

Việc xác định giá trị pháp lý của chữ ký số vô cùng quan trọng, nó đảm bảo tính minh bạch, hợp pháp và an toàn những điều cần thiết trong các giao dịch điện tử và để chống giả mạo chữ ký. Những vấn đề về giá trị pháp lý của chữ ký số được quy định tại Điều 8 của Nghị Định 130/2018/NĐ - CP như sau:

* Đối với trường hợp văn bản cần có chữ ký số thì yêu cầu đối với một văn bản dữ liệu được xem là hợp pháp nền được ký bằng chữ ký số đảm bảo an theo quy định của pháp luật tức là Điều 9 của Nghị Định này.
* Đối với trường hợp văn bản cần đáp ứng được yêu cầu đóng dấu của cơ quan tổ chức thì văn bản dữ liệu đó phải được ký bởi chữ ký của cơ quan tổ chức đảm an toàn theo quy định của pháp luật tức Điều 9 của Nghị Định này.
* Đối với trường hợp chữ ký số nước ngoài được cấp giấy phép sử dụng tại Việt Nam sẽ có giá trị pháp lý và hiệu lực tương đương như chữ ký số cấp phép bởi tổ chức cung cấp dịch vụ chứng thực chữ ký số công cộng của Việt Nam.

### Quy định về điều kiện tạo chữ ký số đảm bảo an toàn đối với các đơn vị cung cấp

Đối với các đơn vị cung cấp chữ ký số thì phải đáp ứng đầy đủ ba điều kiện tạo chữ ký số đảm bảo an toàn được quy định rõ tại Điều 9 của Nghị Định 130/2018/NĐ - CP như sau:

* Đầu tiên, chữ ký số phải được tạo ra trong khoản thời gian chứng thư số có hiệu lực và được kiểm tra bằng khóa công khai được ghi trên chứng thư số đó.
* Tiếp theo, chữ ký số phải được tạo ra bằng việc sử dụng khóa bí mật tương ứng với khóa công khai được ghi trên chứng thư số do một trong bốn tổ chức sau đây cấp:

+ Tổ chức cung cấp dịch vụ chứng thực chữ ký số quốc gia.

+ Tổ chức cung cấp dịch vụ chứng thực chữ ký số chuyên dùng cho chính phủ.

+ Tổ chức cung cấp dịch vụ chứng thực chữ ký số công cộng.

+ Tổ chức cung cấp dịch vụ chứng thực chữ ký số chuyên dùng của các cơ quan, tổ chức được cấp giấy chứng nhận đủ điều kiện đảm bảo an toàn cho chữ ký số chuyên dùng.

* Cuối cùng, đảm bảo chắc chắn rằng khóa bí mật chỉ thuộc sự kiểm soát của người ký tại thời điểm đó.

## Chứng thư số, chữ ký số nước ngoài tại việt nam

### Đối tượng sử dụng chứng thư số nước ngoài

Đối với điều kiện sử dụng chứng thư số nước ngoài tại Việt Nam đã được quy định tại Điều 43 của Nghị Định 130/2018/NĐ - CP như sau:

* Chứng thư số còn hiệu lực sử dụng.
* Được Bộ Thông tin và Truyền thông cấp giấy phép sử dụng tại Việt Nam hoặc chấp nhận trong giao dịch quốc tế. Trường hợp sử dụng chứng thư số nước ngoài cho máy chủ và phần mềm không cần giấy phép.

### Đối tượng sử dụng chứng thư số nước ngoài

Được quy định tại Điều 44 của bộ luật này:

* Tổ chức, cá nhân nước ngoài tại Việt Nam.
* Tổ chức, cá nhân Việt Nam có nhu cầu giao dịch điện tử với đối tác nước ngoài mà chứng thư số của các tổ chức cung cấp dịch vụ chứng thực chữ ký số trong nước chưa được công nhận tại nước đó.

### Phạm vi hoạt động và thời hạn giấy phép sử dụng chứng thư số nước ngoài tại Việt Nam

Những quy định này được ghi rõ trong Điều 45 của bộ luật này như sau:

* Phạm vi hoạt động là các giao dịch điện tử của đối tượng sử dụng chứng thư số nước ngoài quy định tại Điều 44 Nghị định này.
* Thời hạn giấy phép sử dụng chứng thư số nước ngoài tại Việt Nam là 05 năm nhưng không quá thời gian hiệu lực của chứng thư số.

**Điều kiện cấp giấy phép sử dụng**

Những quy định này được ghi rõ trong Điều 46 của bộ luật này như sau:

### Đối với thuê bao sử dụng chứng thư số nước ngoài tại Việt Nam:

* Thuộc đối tượng quy định tại Điều 44 Nghị định này;
* Có một trong các văn bản sau đây để xác thực thông tin trên chứng thư số:
* Giấy chứng nhận đăng ký doanh nghiệp hoặc giấy chứng nhận đầu tư hoặc quyết định thành lập hoặc quyết định quy định chức năng, nhiệm vụ, quyền hạn đối với tổ chức; chứng minh nhân dân hoặc căn cước công dân hoặc hộ chiếu đối với cá nhân;
* Văn bản của cơ quan có thẩm quyền cho phép tổ chức, cá nhân nước ngoài hoạt động hợp pháp tại Việt Nam đối với thuê bao là tổ chức, cá nhân nước ngoài;
* Trường hợp được ủy quyền sử dụng chứng thư số phải có ủy quyền cho phép hợp pháp sử dụng chứng thư số và thông tin thuê bao được cấp chứng thư số phải phù hợp với thông tin trong văn bản ủy quyền, cho phép.

### Đối với tổ chức cung cấp dịch vụ chứng thực chữ ký số nước ngoài có chứng thư số được công nhận tại Việt Nam

* Thành lập và hoạt động hợp pháp tại quốc gia mà tổ chức cung cấp dịch vụ chứng thực chữ ký số nước ngoài đăng ký hoạt động;
* Đáp ứng danh mục tiêu chuẩn bắt buộc áp dụng về chữ ký số và dịch vụ chứng thực chữ ký số do Bộ Thông tin và Truyền thông ban hành hoặc các tiêu chuẩn quốc tế về chữ ký số được Bộ Thông tin và Truyền thông xác định có độ an toàn thông tin tương đương;
* Được doanh nghiệp kiểm toán chứng nhận hoạt động nghiệp vụ tuân thủ các tiêu chuẩn quốc tế có uy tín về dịch vụ chứng thực chữ ký số.

# XÂY DỰNG ỨNG DỤNG TẠO VÀ XÁC THỰC CHỮ KÝ SỐ

Từ những kiến thức chúng em đã tìm hiểu được ở các chương phía trên, kết hợp với những kiến thức có sẵn và được dạy dỗ ở trường, chúng em đã áp dụng thành công chúng vào một ứng dụng có thể sử dụng trong thực tiễn. Bằng tất cả tâm huyết, chúng em đã tạo ra một ứng dụng có thể tạo và xác thực chữ ký số như quy trình chuẩn có tên là “RSA Digital Signature”, được viết bằng ngôn ngữ C# trên nền tảng Winform có giao diện đồ hoạ thân thiện với người dùng. Trong chương này chúng ta sẽ tìm hiểu sâu về ứng dụng này.

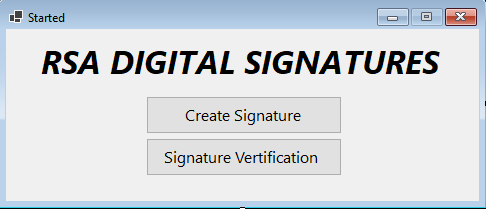
## Giới thiệu về phần mềm

**Phần mềm này được tạo ra để thực hiện các chức năng**

* Thực hiện tạo khóa bí mật và khóa công khai.
* Thực hiện tạo chữ ký số bằng giải thuật RSA.
* Thực hiện giải mã để xác thực cho chữ ký được tạo bằng giải thuật RSA.

Bao gồm 3 form chính: Started, Tạo chữ ký và Xác thực chữ ký

Khi khởi động ứng dụng, người dùng sẽ tiếp xúc đầu tiên với form Started, trong form này gồm 2 option để người dùng lựa chọn: Tạo chữ ký và Xác thực chữ ký. Tuỳ theo chọn lựa của người dùng mà các form có chức năng như tên gọi được hiện ra.



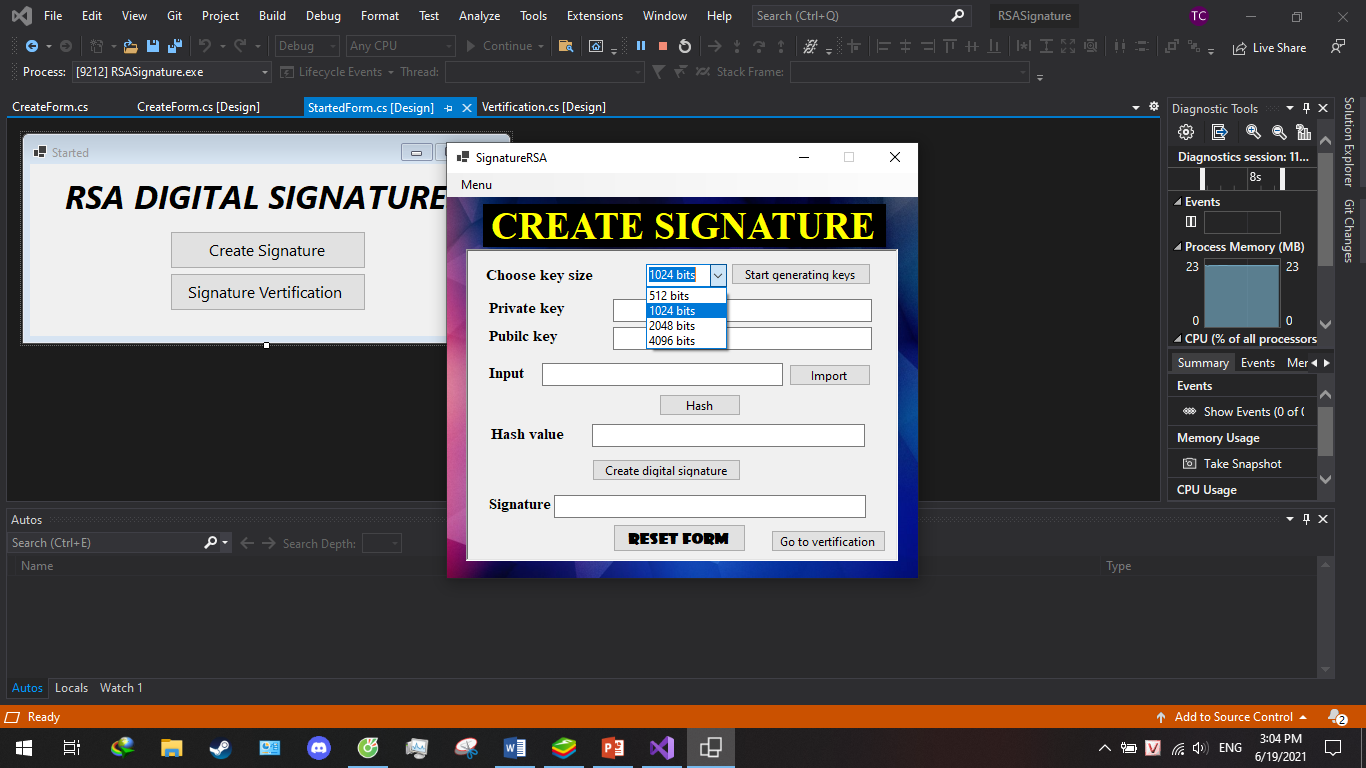
Hình 5.1 Giao diện started của phần mềm.

## Thực hiện tạo khóa bí mật và khóa công khai

Form Tạo chữ ký số có công dụng tạo khoá cung cấp cho hệ mã hoá RSA và sử dụng các khoá đã tạo cho quá trình tạo chữ ký số.

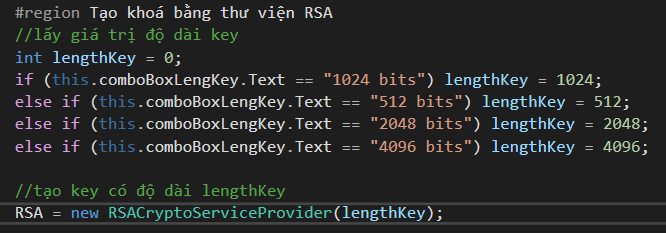
Phần mềm thực hiện việc tạo và lưu các khóa tại vị trí được chỉ định bởi người dùng. Nó sử dụng gói thư viện mã hoá *System.Security.Cryptography* được hỗ trợ trên C# để thực hiện quá trình này.

Phần mềm hỗ trợ tạo khóa với 4 độ dài khác nhau là 512 bit, 1024 bit, 2048 bit, 4096 bit. Trong đó khóa có độ dài càng lớn thì độ bảo mật càng cao, nhưng bù lại thời gian tạo chữ ký sẽ càng lâu vì độ phức tạp của khoá.



Hình 5.2 Các độ dài khóa mà chương trình hỗ trợ.

Độ dài khoá này sẽ được lưu vào biến lengthKey, tuỳ vào kích thước người dùng đã chọn. Kế tiếp truyền kích thước khoá vào đối tượng RSA trên.



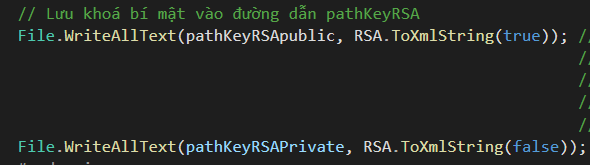
Hình 5.3 Code truyền giá trị độ dài khoá vào đối tượng RSA

Ở đây chúng ta sẽ tạo khoá dựa trên độ dài khoá đã gán. Câu lệnh để tạo cả 2 khoá trong đối tượng RSA là

*RSA.ToXmlString(option);*

Trong đó khoá bí mật sẽ có giá trị option là “true”, trong khi đó khoá công khai sẽ có giá trị option là “false”. Trong mã hoá RSA theo đúng quy trình thì người gửi sẽ sử dụng khoá công khai của người nhận để mã hoá và người nhận sẽ giải mã bằng khoá bí mật của mình. Thế nhưng vì đây là ứng dụng của RSA vào chữ ký số, người gửi chính là chúng ta, nên khi mã hoá chữ ký số và gửi cho người nhận mà vẫn giữ được các tính chất của chữ ký số, chúng ta sử dụng khoá bí mật của người gửi để mã hoá và người nhận sẽ dùng khoá công khai của họ để giải mã. Chính vì thế chức năng của hai khoá sẽ đổi cho nhau và option tạo khoá ở phần chữ ký số sẽ ngược lại với RSA thông thường, cụ thể ở đây là chúng ta tạo khoá bí mật với option là “false” và khoá công khai sẽ là option “true”.

Và chúng ta cần lưu khoá đã tạo vào một file, ở đây chúng ta sẽ tạo một file xml để lưu trữ khoá và gọi hàm ghi file với đường dẫn đến file xml và hàm tạo khoá RSA trên.



Hình 5.4 Code tạo khoá công khai và bí mật

Trong đó

*File.WriteAllText()*: hàm ghi file trong winform

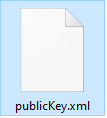
*pathKeyRSApublic*: đường dẫn lưu file xml chứa khoá công khai

*pathKeyRSAPrivate*: đường dẫn lưu file xml chứa khoá bí mật

*RSA.ToXmlString(true)*: tạo khoá công khai và trả về giá trị khoá ở dạng XML

*RSA.ToXmlString(false)*: tạo khoá bí mật và trả về giá trị khoá ở dạng XML

Sau khi tạo thì phần mềm sẽ xuất ra khóa dưới định dạng “.xml”. và dữ liệu của khoá sẽ được xuất



Hình 5.5 Khóa bí mật và khoá công khai

Giá trị của khoá này sẽ được đọc và xuất ra màn hình ở textbox Publickey và PrivateKey để dễ dàng theo dõi.

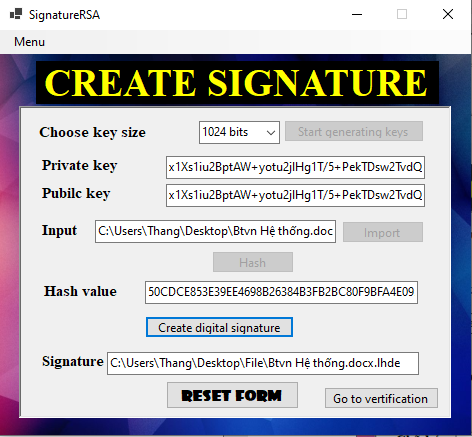
## Thực hiện tạo chữ ký số bằng giải thuật RSA

Việc tạo chữ ký số được thực hiện bằng cách mã hóa dữ liệu đầu vào đã được chuyển thành chuỗi bit (chuỗi nhị phân) sử dụng khóa bí mật đã được tạo trước đó.

Dữ liệu đầu vào có thể là *bất kỳ* dạng file nào mà chúng ta muốn ký: word, pdf, video, … Chọn vào button Import ở mục Input để đưa vào file đầu vào.

Sau khi người dùng đưa vào file dữ liệu đầu vào, ô textbox Input sẽ mang giá trị là đường dẫn dẫn đến file dữ liệu đầu vào. Theo quy trình tạo chữ ký số thì chúng ta sẽ băm file input ra bằng các công thức hàm băm. Ở chương trình này chúng ta sử dụng hàm băm SHA256 với độ dài ký tự là 65 chữ số có hỗ trợ sẵn của thư viện C#. Button hash khi được nhấn vào sẽ gọi hàm thực hiện băm SHA256 để thực hiện băm với đường dẫn truyền vào là file input ở trên.

Sau đó để tạo chữ ký số, nhấn vào button Create digital signature



Hình 5.6 Giao diện tạo chữ ký số.

Chương trình sẽ chạy hàm tạo chữ ký số *btnEncryptClick()*, hàm này có chức năng kiểm tra điều kiện các file cần có đã đủ chưa và chuẩn bị các bước để mã hoá. Câu lệnh:

*RSA.FromXmlString(File.ReadAllText(this.pathKeysXMLPrivate));*

Có tác dụng gán khoá bí mật từ đường dẫn input đã tạo ở trên vào đối tượng RSA, sau đó gọi hàm *RSA\_Algorithm()* để thực hiện quá trình mã hoá

*RSA\_Algorithm(inputFileName, outputFileName, RSA.ExportParameters(false),true);*

Trong đó

*inputFileName, outputFileName:* là đường dẫn đến file dữ liệu đầu vào và ra

*RSA.ExportParameters(false)*: là câu lệnh xuất ra khoá bí mật của đối tượng RSA để thực hiện mã hoá.

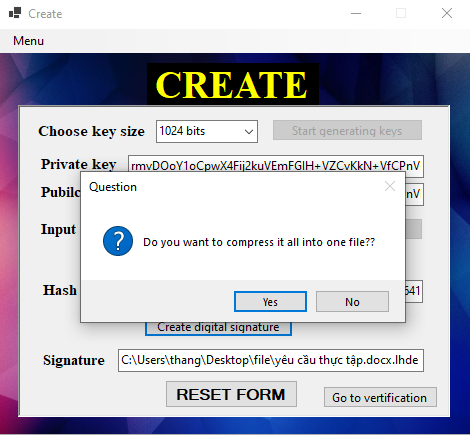
Trong hàm *RSA\_Algorithm(),* sau khi các bước chuẩn bị hoàn tất, dữ liệu chữ ký số sau khi mã hoá bằng câu lệnh *encryptedData = RSA.Encrypt(bin, false)* sẽ được ghi vào file output người dùng đã lưu.

Sau khi tạo thì chữ ký sẽ được lưu dưới định dạng *“tên file được sử dụng để ký .lhde”.* Trong đó “.lhde” là do lập trình viên quy định để có thể phân biệt file chữ ký với các file khác. Vị trí lưu của chữ ký sẽ do người sử dụng chỉ định.

File được dùng làm chữ ký : 

File chữ ký : 

Ngoài ra chương trình còn có chức năng nén file khóa công khai và file chữ ký vào chung 1 tệp để tiện lợi hơn cho người gửi dễ dàng gửi đi, một tệp nén “secret.zip” sẽ được tạo chứa chữ ký và khoá công khai bên trong.



Hình 5.7 Tuỳ chọn nén file sau khi tạo thành công chữ ký

Chương trình sẽ hỏi chúng ta vị trí lưu file nén trên .Sau khi nén, file sẽ có định dạng như sau: , chúng ta chỉ cần gửi file nén này và file hợp đồng (input) cho đối tác.

## Thực hiện giải mã để xác thực cho chữ ký được tạo bằng giải thuật RSA

Việc giải mã để xác thực chữ ký được thực hiện bằng cách giải mã file chữ ký sử dụng khóa công khai do người gửi tạo gửi tới để cho ra dữ liệu ban đầu là một chuỗi bit.Với mã lệnh:

*encryptedData = RSA.Decrypt(bin, false);* dùng để giải mã chữ ký.

Trong đó:

*“bin”* là số bit có thể mã hóa đã được tính trong hàm *RSA\_Algorithm();*

*“false”* ở đây nếu bạn muốn sử dụng đệm mã OAEP thì để là “*true*”.

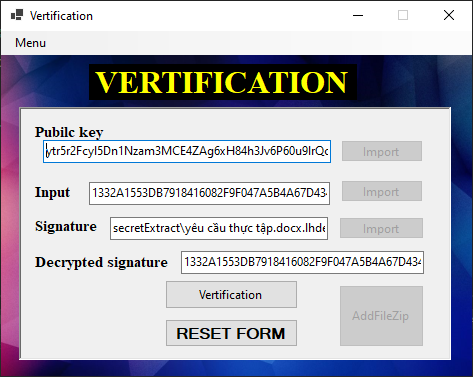
Sau đó đem so sánh chuỗi bit đã được giải mã với chuỗi bit của dữ liệu được ký. Nếu 2 chuỗi giống nhau thì chương trình sẽ kết luận file được ký không bị thay đổi và cho phép chúng ta mở file còn nếu khác nhau thì chương trình sẽ kết luận file được ký đã bị thay đổi và chương trình sẽ không mở file cho chúng ta.

Chúng ta có 2 cách để nhập dữ liệu để xác thực:

* Add vào file nén zip nhận được từ bên gửi và file hợp đồng gốc (input)
* Add từng file chữ ký, khoá công khai và file hợp đồng gốc vào

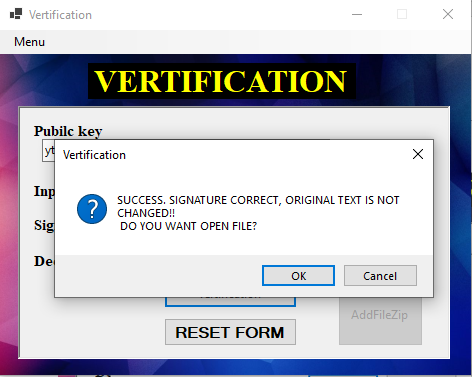
Sau đó chúng ta ấn vào Verification, ứng dụng sẽ tự động kiểm tra và xác thực.

Nếu chữ ký xác thực đúng và dữ liệu không đổi, chúng ta có thể mở trực tiếp file hợp đồng và thực hiện ký và gửi ngược lại cho đối tác. Còn nếu xác thực chữ ký sai, tức nội dung văn bản đã bị thay đổi bất hợp pháp, ứng dụng sẽ cảnh báo nguy hiểm và không cho mở file hợp đồng.



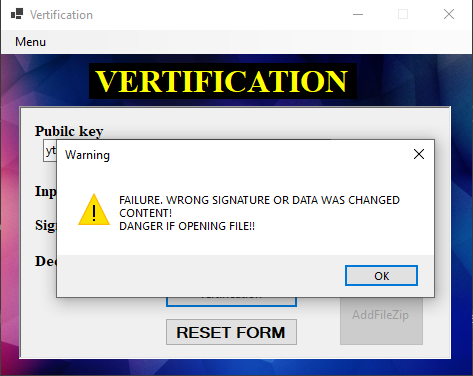
Hình 5.8 Giao diện xác thực chữ ký số.

* Nếu chữ ký xác thực đúng



Hình 5.9 Chữ ký giống với văn bản.

* Nếu chữ ký được xác thực là sai



Hình 5.10 Chữ ký không giống với văn bản.

# KẾT LUẬN

Trong quá trình nghiên cứu thực hiện đồ án môn học “**TÌM HIỂU VỀ CHỮ KÝ ĐIỆN TỬ VÀ XÂY DỰNG ỨNG DỤNG TẠO VÀ XÁC THỰC CHỮ KÝ ĐIỆN TỬ**”, dưới sự hướng dẫn chỉ bảo tận tình của cô Mạnh Thiên Lý, chúng em đã nghiên cứu và nắm vững được khái niệm, quy trình tạo và xác thực, cách thức hoạt động của chữ ký số và đã tạo thành công một chương trình để áp dụng những kiến thức đã tìm hiểu được vào ứng dụng trong đời sống. Do điều kiện về thời gian và nguồn kiến thức còn thiếu nên việc tạo ứng dụng để xây dựng và xác thực chữ ký số của chúng em chỉ giới hạn ở phạm vi thừa kế các thư viện về hàm băm và mã hoá RSA để áp dụng vào quy trình thực hiện. Chương trình này được chúng em xây nhằm mục đích thực hiện các hoạt động tạo và xác thực đơn giản chữ ký số, nếu được nâng cấp chỉnh chu thì chương trình của chúng em có khả năng hoạt động như các phần mềm tạo và xác thực chữ ký số thương mại hiện nay. Mặc dù đã rất cố gắng nhưng đồ án này không thể tránh khỏi những sai sót. Em rất mong được sự góp ý, giúp đỡ nhiệt tình của các thầy cô và các chúng ta để đề tài này của em được hoàn thiện. Em xin chân thành cảm ơn.

# PHỤ LỤC

**DANH MỤC TIÊU CHUẨN** **VỀ CHỮ KÝ SỐ VÀ ĐỊNH DẠNG VĂN BẢN ĐIỆN TỬ KÝ SỐ**  
*(Ban hành kèm theo Thông tư số 41/2017/TT - BTTTT ngày 19 tháng 12 năm 2017 của Bộ trưởng Bộ Thông tin và Truyền thông)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SSố TT** | **Loại tiêu chuẩn** | **Ký hiệu tiêu chuẩn** | **Tên đầy đủ của tiêu chuẩn** | **Quy định áp dụng** |
| **1** | **Tiêu chuẩn về định dạng văn bản điện tử** | | | |
| 1.1 | Văn bản điện tử ký số (đáp ứng yêu cầu tại Điều 6 Thông tư này) | (.pdf) | Định dạng Portable Document (.pdf) - Version 1.4 trở lên | Bắt buộc áp dụng |
| 1.2 | Định dạng văn bản điện tử ký số khác bao gồm: văn bản, bảng tính, trình diễn, ảnh đồ họa | Tiêu chuẩn về văn bản, bảng tính, trình diễn, ảnh đồ họa thuộc danh mục tiêu chuẩn kỹ thuật về ứng dụng công nghệ thông tin trong cơ quan nhà nước. | | Khuyến khích áp dụng |
| **2** | **Tiêu chuẩn về chữ ký số** | | | |
| 2.1 | Tiêu chuẩn mật mã chữ ký số | PKCS#1 | RSA Cryptography Standard (Phiên bản 2.1 trở lên) | Bắt buộc áp dụng |
| TCVN 7635:2007 | Các kỹ thuật mật mã - Chữ ký số. |
| 2.2 | Tiêu chuẩn hàm băm an toàn | FIPS PUB 180-4 | Secure Hash Standard | Bắt buộc áp dụng hàm băm SHA- 256, 384, 512. |
| 2.3 | An toàn trao đổi bản tin XML | XML Encryption Syntax and Processing | XML Encryption Syntax and Processing | Bắt buộc áp dụng |
| XML Signature Syntax and Processing | XML Signature Syntax and Processing | Bắt buộc áp dụng |
| 2.4 | Quản lý khóa công khai bản tin XML | XKMS v2.0 | XML Key Management Specification version 2.0 | Bắt buộc áp dụng |
| 2.5 | Chuẩn cú pháp thông điệp mật mã cho ký số và mã hóa | PKCS#7 v1.5 (RFC 2315) | Cryptographic message syntax for file - based signing and encrypting | Bắt buộc áp dụng |
| **3** | **Tiêu chuẩn dịch vụ cấp dấu thời gian** | | | |
| 3.1 | Giao thức cấp dấu thời gian | RFC 3161 | Internet X.509 Public Key Infrastructure - Time stample Prototol | Bắt buộc áp dụng |
| 3.2 | Dịch vụ cấp dấu thời gian | ISO/IEC 18014 - 1:2008 | Information technology- Security techniques - Time Stamping services - Part 1: Framework | Bắt buộc áp dụng  - Áp dụng bộ ba tiêu chuẩn: ISO/IEC 18014- 1:2008); ISO/IEC 18014- 2:2009); ISO/IEC 18014- 3:2009. |
| ISO/IEC 18014 - 2:2009 | Information technology - Security techniques - Time Stamping services - Part 2: Mechanisms producing independent tokens |
| ISO/IEC 18014 - 3:2009 | Information technology - Security techniques - Time-stamping services - Part 3: Mechanisms producing linked tokens |

# TÀI LIỆU THAM KHẢO

Tiếng Việt

[1]. PGS.TS. Nguyễn Bình, giáo trình Mật mã học (Chủ biên), 2003.

[2]. TS. Vũ Đức Thịnh, slide bài giảng môn Mật mã học, trường Đại học Công nghiệp Thực Phẩm TP. HCM.

[3]. [https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/Thong-tu-41-2017-TT-BTTTT-su-dung-chu-ky-so-cho-van-ban-dien-tu-trong-co-quan-nha-nuoc 370856.aspx](https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/Thong-tu-41-2017-TT-BTTTT-su-dung-chu-ky-so-cho-van-ban-dien-tu-trong-co-quan-nha-nuoc%20370856.aspx)

[4]. [https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/Nghi-dinh-130-2018-ND-CP-huong-dan-Luat-Giao-dịch-dien-tu-ve-chu-ky-so-358259.aspx](https://thuvienphapluat.vn/van-ban/Cong-nghe-thong-tin/Nghi-dinh-130-2018-ND-CP-huong-dan-Luat-Giao-dich-dien-tu-ve-chu-ky-so-358259.aspx)

[5]. <https://aita.gov.vn/tieu-chuan-sha-2>

[6]. <https://stackjava.com/demo/sha-la-gi-code-vi-du-sha1-sha2-voi-java.html>

**Tiếng Anh**

[7]. <https://en.wikipedia.org/wiki/Digital_signature>

[8]. <https://sectigostore.com/blog/hash-function-in-cryptography-how-does-it-work/>