

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**---------------------------------------**

**BÀI TẬP LỚN**

**Môn: An toàn và bảo mật thông tin**

**Tìm hiểu về chữ ký điện tử RSA và viết ứng dụng minh họa**

**CBHD: *ThS. Trần Phương Nhung***

**Nhóm: 3**

**Thành viên nhóm:**

**1.Nguyễn Thành Đạt\_2019602136 (Nhóm trưởng)**

**2.Nguyễn Bá Đông \_2019605704**

**3.Nguyễn Tiến Đại \_2019606524**

**4.Nguyễn Trọng Đạt\_2019604461**

**5.Nguyễn Đức Đăng\_2020603392**

**Hà Nội – Năm 2022**

## Mục Lục

Contents

[Mục Lục 2](#_Toc113510855)

[Chương 1. Tổng quan 4](#_Toc113510856)

[1.1. Giới thiệu 4](#_Toc113510857)

[1.2. Chữ ký số (Digital Signature) 5](#_Toc113510858)

[1.3. Mục đích nghiên cứu 6](#_Toc113510859)

[1.4. Phương pháp nghiên cứu. 6](#_Toc113510860)

[1.5. Đối tượng nghiên cứu 7](#_Toc113510861)

[1.6. Ý nghĩa khoa học và thực tiễn 7](#_Toc113510862)

[Chương 2. Kết quả nghiên cứu 8](#_Toc113510863)

[2.1. Giới thiệu 8](#_Toc113510864)

[2.2. Nội dung thuật toán 8](#_Toc113510865)

[2.2.1. Phương pháp mã hóa RSA – Chữ kí điện tử RSA 8](#_Toc113510866)

[a) Tổng quan 8](#_Toc113510867)

[b) Giải thuật và cài đặt giải thuật 10](#_Toc113510868)

[2.2.2. Phương pháp mã hóa bất đối xứng ứng dựng trong chữ ký điện tử 13](#_Toc113510869)

[2.2.3. Hàm băm MD5 15](#_Toc113510870)

[a) Tổng quan về hàm băm 15](#_Toc113510871)

[b)Hàm băm MD5 16](#_Toc113510872)

[2.3. Thiết kế, cài đặt chương trình demo thuật toán 22](#_Toc113510873)

[2.3.1. Giao diện chương trình demo 22](#_Toc113510874)

[2.3.1. Cài đặt chương trình demo 23](#_Toc113510875)

[a) Quá trình ký và xác thực chữ ký số 24](#_Toc113510876)

[b) Thuyết minh chương trình 24](#_Toc113510877)

[2.4. Cài đặt và triển khai 25](#_Toc113510878)

[a) Visual Studio 25](#_Toc113510879)

[b) Netbeans 25](#_Toc113510880)

[2.5. Thực hiện bài toán 26](#_Toc113510881)

[2.5.1. Phân công công việc 26](#_Toc113510882)

[2.5.2. Nguyễn Thành Đạt 28](#_Toc113510883)

[- Một số thư viện đã sử dụng: 29](#_Toc113510884)

[2.5.3. Nguyễn Bá Đông 31](#_Toc113510885)

[- Một số thư viện đã sử dụng: 32](#_Toc113510886)

[2.5.4. Nguyễn Tiến Đại 35](#_Toc113510887)

[2.5.5. Nguyễn Trọng Đạt 37](#_Toc113510888)

[BigInteger: cung cấp một lớp để quản lý các số nguyên có độ dài tùy ý. 37](#_Toc113510889)

[2.5.6. Nguyễn Đức Đăng 40](#_Toc113510890)

[Chương 3. Kiến thức lĩnh hội và bài học kinh nghiệm 42](#_Toc113510891)

[3.1. Nội dung đã thực hiện 42](#_Toc113510892)

[3.2. Hướng phát triển. 43](#_Toc113510893)

[TÀI LIỆU THAM KHẢO 44](#_Toc113510894)

**Lời nói đầu**

Ngày nay trong mọi hoạt động của con người thông tin đóng một vai trò quan trọng không thể thiếu. Xã hội càng phát triển nhu cầu trao đổi thông tin giữa các thành phần trong xã hội ngày càng lớn. Mạng máy tính ra đời đã mang lại cho con người rất nhiều lợi ích trong việc trao đổi và xử lý thông tin một cách nhanh chóng và chính xác. Chính từ những thuận lợi này đã đặt ra cho chúng ta một câu hỏi, liệu thông tin đi từ nơi gửi đến nơi nhận có đảm bảo tuyệt đối an toàn, ai có thể đảm bảm thông tin của ta không bị truy cập bất hợp pháp. Thông tin được lưu giữ, truyền dẫn, cùng sử dụng trên mạng lưới thông tin công cộng có thể bị nghe trộm, chiếm đoạt, xuyên tạc hoặc phá huỷ dẫn đến sự tổn thất không thể lường được. Đặc biệt là đối với những số liệu của hệ thống ngân hàng, hệ thống thương mại, cơ quan quản lý của chính phủ hoặc thuộc lĩnh vực quân sự được lưu giữ và truyền dẫn trên mạng. Nếu như vì nhân tố an toàn mà thông tin không dám đưa lên mạng thì hiệu suất làm việc cũng như hiệu suất lợi dụng nguồn dữ liệu đều sẽ bị ảnh hưởng. Trước các yêu cầu cần thiết đó, việc mã hoá thông tin sẽ đảm bảo an toàn cho thông tin tại nơi lưu trữ cũng như khi thông tin được truyền trên mạng.

Các phương thức tấn công thông qua mạng ngày càng tinh vi, phức tạp có thể dẫn đến mất mát thông tin, thậm chí có thể làm sụp đổ hoàn toàn hệ thống thông tin của tổ chức. Vì vậy an toàn thông tin là nhiệm vụ quan trọng, nặng nề và khó đoán trước đối với các hệ thống thông tin.

Một trong những ứng dụng của an toàn thông tin là chữ ký số. Với đặc điểm là đơn giản cho người sử dụng mà vẫn đảm bảo được tính bảo mật, kỹ thuật sử dụng chữ ký số là một trong những kỹ thuật được sử dụng phổ biến, đa dạng trong hầu hết các lĩnh vực, nhất là Tài chính, Ngân hang, Kế toán…Vì lý do đó, nhóm 3 chúng em đã nghiên cứu về “Giải thuật tạo chữ ký số sử dụng thuật toán RSA”.

# Chương 1. Tổng quan

## Giới thiệu

Mạng máy tính ra đời giúp con người có thể trao đổi thông tin một cách thuận lợi hơn, nhu cầu trao đổi thông tin ngày càng lớn và đa dạng, các tiến bộ về khoa học kỹ thuật không ngừng phát triển ứng dụng để nâng cao chất lượng và lưu lượng truyền tin thì các quan niệm ý tưởng về biện pháp bảo vệ an toàn thông tin cũng được đổi mới.

An toàn thông tin bao gồm các nội dung sau:

* + - Tính bí mật: tính kín đáo riêng tư của thông tin.
    - Tính xác thực: bao gồm xác thực đối tác, xác thực thông tin trao đổi.
    - Tính trách nhiệm: bảo đảm người gởi không thoái thác trách nhiệm về thông tin mình đã gởi.
    - Tính toàn vẹn thông điệp nhận được không bị thay đổi một cách ngẫu nhiên hay cố ý.

Bảo vệ an toàn thông tin là một chủ đề rộng liên quan đến nhiều lĩnh vực và trong thực tế có nhiều phương pháp để thực hiện bảo vệ an toàn thông tin. Phương pháp đạt hiệu quả và kinh tế nhất hiện nay là bảo vệ an toàn thông tin bằng biện pháp thuật toán (Mật mã thông tin).

Mật mã thông tin là việc biến đổi thông tin thành một dạng khác nhằm che dấu nội dung, ý nghĩa thông tin. Ngày nay, rất nhiều ứng dụng mã hóa và bảo mật thông tin được sử dụng trong nhiều lĩnh vực khác nhau trên thế giới như an ninh, quân sự, quốc phòng... cho đến các ứng dụng thương mại điện tử ngân hàng...

Trong những năm gần đây sự phát triển của khoa học máy tính và Internet càng cho thấy tầm quan trọng của mật mã thông tin. Ứng dụng của mật mã thông tin không chỉ đơn thuần là mã hóa và giải mã thông tin nó còn được dùng trong việc chứng thực nguồn gốc thông tin, xác thực người sở hữu mã hóa, các quy trình giúp trao đổi thông tin và thực hiện trao đổi an toàn trên mạng. Sự phát

triển nhanh chóng của Internet đã kéo theo một loạt các dịch vụ mới ra đời như trò chuyện, quảng cáo, đặt hàng, mua hàng... qua Internet. Trong số đó thì các dịch vụ thương mại điện tử là một bước nhảy vọt trong việc ứng dụng của Internet vào cuộc sống và kinh doanh.

* Ưu điểm: việc trao đổi thông tin, dữ liệu không cần qua giấy tờ, tiện lợi trong giao dịch.
* Nhược điểm: vấn đề xác thực nguồn gốc và nội dung thông tin.

## Chữ ký số (Digital Signature)

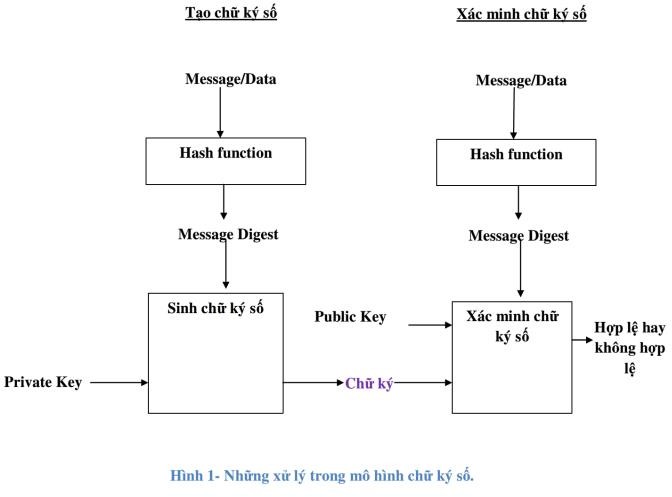
Là một kỹ thuật giúp sự phát triển của thương mại điện tử, giúp việc trao đổi dữ liệu được tin cậy hơn. Chữ ký số chứng minh rằng một thông điệp đã thực sự được gởi được chính người gởi mà không phải của kẻ khác giả mạo. Khi sử dụng chữ ký số ta có thể kiểm tra được tính xác thực của thông điệp. Nó không chỉ kiểm tra được thông tin về người gởi mà còn có thể kiểm tra được cả thông tin về thông điệp, giúp ta có thể biết được thông điệp có bị giả mạo hay can thiệp gì không trong quá trình vận chuyển.

Chữ ký số RSA rất phổ biến trong thương mại điện tử giao dịch trực tuyến vì nó đảm bảo an toàn khi điều kiện độ dài khóa đủ lớn. Dựa trên công nghệ mã hóa trên,chữ ký số RSA gồm một cặp khóa được mã hóa không đối xứng như một khóa công khai (public key) và khóa bí mật (private key).Cụ thể như sau :

* Khóa bí mật : dùng để tạo ra chữ ký số RSA
* Khóa công khai : tác dụng trong việc thẩm định,kiểm tra chữ ký số và xác thực về người dùng.Nó được tạo ra bởi khóa bí mật tương ứng với mỗi cặp khóa.
* Người ký : Đối tượng dùng khóa bí mật của mình để ký số vào một dữ liệu nào đó thể hiện tên mình
* Người nhận :Đối tượng nhận được thông điệp dữ liệu được ký số ,bằng việc sử dụng các chứng thư số để kiểm tra chữ ký số cho dữ liệu nhận được. Ngoài ra còn tiến hành các hoạt động,giao dịch điện tử…
* Ký số : Đưa khóa bí mật RSA vào phần mềm tự động tạo và gắn chữ ký số cho thông điệp dữ liệu nào đó.

Chữ ký số cung cấp các chức năng quan trọng như: tính xác thực (authentication), tính bí mật (confidentiatly), tính toàn vẹn dữ liệu (data integrity), tính không thoái thác (non-repudiation).

**\* Quy trình tạo và xác minh chữ ký số:**



## 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.

## 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 Java để xây dựng một ứng dụng về mã hóa dữ liệu và chữ ký số.

## Đối tượng nghiên cứu

* Hệ mật mã khóa công khai RSA
* Tìm hiểu về phương pháp mã hóa bất đối xứng ứng dụng trong chữ kí điện tử, tìm hiểu về hàm băm MD5.

## Ý nghĩa khoa học và thực tiễn

*Về mặt lý thuyết:*

* Trình bày trình bày khát quát về mật mã, khái niệm về hệ mật mã khoá bí mật và hệ mật mã khoá công khai.
* Trình bày lý thuyết chung về các phương pháp mã hoá: phương pháp mã hoá khóa bí mật và phương pháp mã hoá khóa công khai, nêu được các ưu điểm và nhược điểm của hai phương pháp này. Trình bày chi tiết hệ mật mã khóa công khai RSA.
* Trình bày lý thuyết chung về mô hình chữ ký số, các lược đồ chữ ký số cụ thể, được xây dựng trên mật mã hoá khoá công khai.

*Về mặt thực tiễn*

* Xây dựng được chương trình ứng dụng dựa vào hệ mật mã RSA có chức năng bảo mật nội dung cho các tập tin là các dữ liệu hoặc các tài liệu, tạo và kiểm tra chữ ký số cho các tập tin đó để xác định tính toàn vẹn nội dung và chủ nhân của tập tin khi thực hiện trao đổi qua mạng Internet.

# Chương 2. Kết quả nghiên cứu

## Giới thiệu

**Đề tài nghiên cứu**: Tìm hiểu về chữ ký điện tử RSA và viết ứng dụng minh họa.

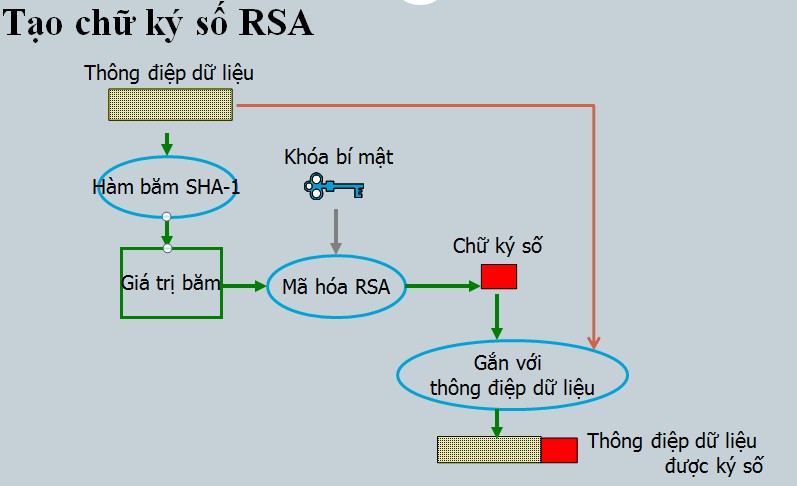
* Những nhiệm vụ, công việc chính khi thực hiện bài tập lớn:
  + Tìm hiểu về phương pháp mã hóa RSA
  + Phương pháp mã hóa bất đối xứng ứng dụng trong chữ ký điện tử
  + Hàm băm MD5
  + Ứng dụng để xây dựng chương trình, nắm vững một ngôn ngữ lập trình ( java, C#, C++,Javascript, python)
* Kết quả đạt được : Xây dựng chương trình demo thuật toán RSA xác thực chữ ký điện tử

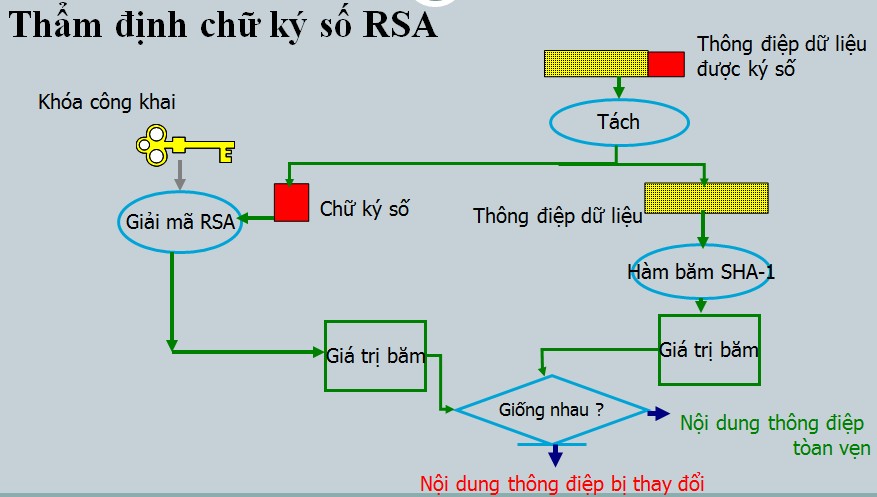
## Nội dung thuật toán

## Phương pháp mã hóa RSA – Chữ kí điện tử RSA

## Tổng quan

Quy trình tạo chữ ký số RSA:





**Quá trình ký (bên gửi):**

* Tính toán chuỗi đại diện (message digest/ hash value) của thông điệp sử dụng một giải thuật băm (Hashing algorithm) SHA-1.
* Chuỗi đại diện được ký sử dụng khóa riêng (Priavte key) của người gửi và giải thuật tạo chữ ký (Signature/ Encryption algorithm) RSA. Kết quả chữ ký số (Digital signature) của thông điệp hay còn gọi là chuỗi đại diện được mã hóa bởi giải thuật RSA (Encryted message digest).
* Thông điệp ban đầu (message) được ghép với chữ ký số (Digital signature) tạo thành thông điệp đã được ký (Signed message).
* Thông điệp đã được ký (Signed message) được gửi cho người nhận.

**Quá trình kiểm tra chữ ký (bên nhận):**

* Tách chữ ký số RSA và thông điệp gốc khỏi thông điệp đã ký để xử lý riêng;
* Tính toán chuỗi đại diện MD1 (message digest) của thông điệp gốc sử dụng giải thuật băm (là giải thuật sử dụng trong quá trình ký là SHA-1)
* Sử dụng khóa công khai (Public key) của người gửi để giải mã chữ ký số RSA -> chuỗi đại diện thông điệp MD2

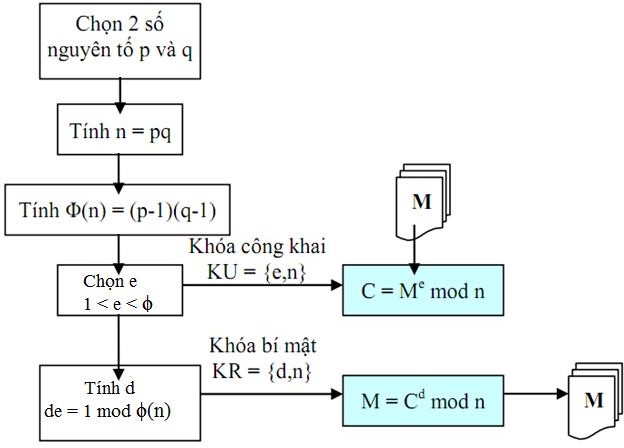
**So sánh MD1 và MD2:**

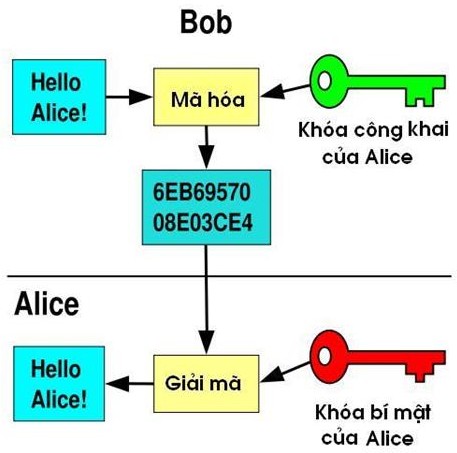
* Nếu MD1=MD2 -> chữ ký kiểm tra thành công. Thông điệp đảm bảo tính toàn vẹn và thực sự xuất phát từ người gửi (do khóa công khai được chứng thực).
* Nếu MD1<>MD2 -> chữ ký không hợp lệ. Thông điệp có thể đã bị sửa đổi hoặc không thực sự xuất phát từ người gửi.

## Giải thuật và cài đặt giải thuật

* 1. **Giải thuật RSA được dùng trong việc tạo khóa, mã hóa, giải mã.**

*Sơ đồ giải thuật :*



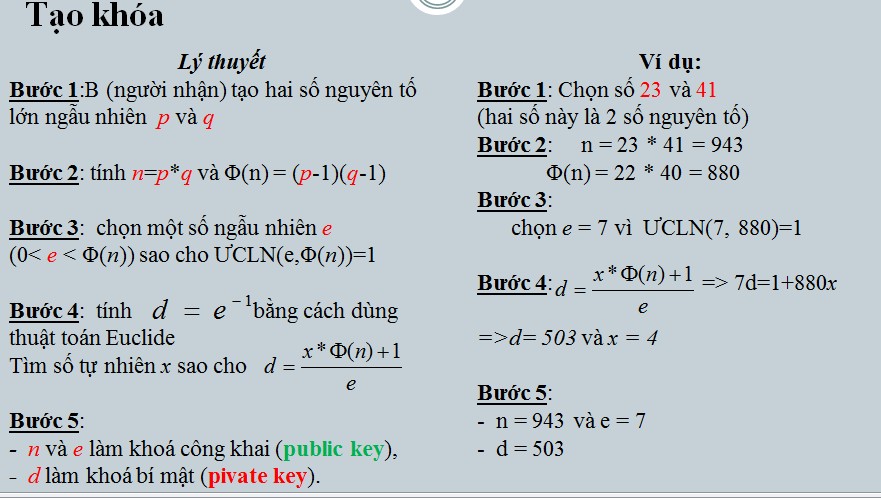


Thuật toán RSA có hai Khóa:

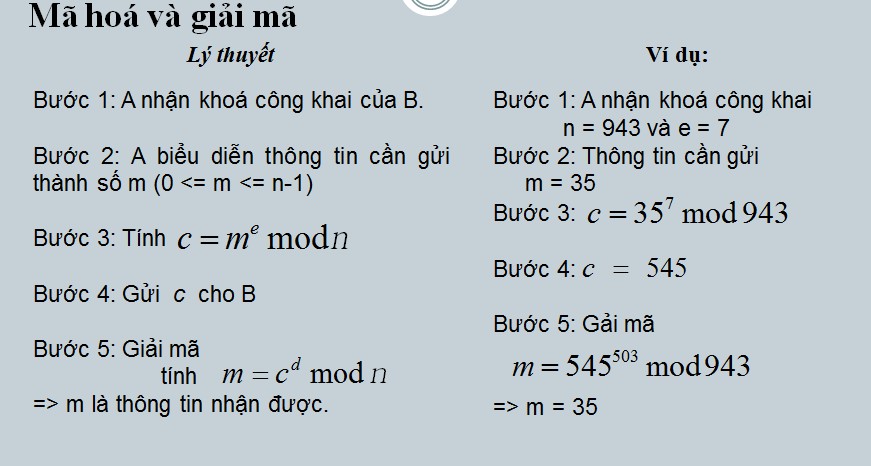
* ***Khóa công khai*** (**Public key**): được công bố rộng rãi cho mọi người và được dùng để mã hóa
* ***Khóa bí mật*** (**Private key**):

Những thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa bí mật tương ứng

**Các bước tạo khóa:**



***Mã hóa và giải mã***



## Phương pháp mã hóa bất đối xứng ứng dựng trong chữ ký điện tử

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

Thuật ngữ mật mã hóa khóa bất đối xứng thường được dùng đồng nghĩa với mật mã hóa khóa công khai mặc dù hai khái niệm không hoàn toàn tương đương. Có những thuật toán mật mã khóa bất đối xứng không có tính chất khóa công khai và bí mật như đề cập ở trên mà cả hai khóa (*cho mã hóa và giải mã*) đều cần phải giữ bí mật.

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

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

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

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

Không phải tất cả các thuật toán mật mã hóa khóa bất đối xứng đều hoạt động giống nhau nhưng phần lớn đều gồm 2 khóa có quan hệ toán học với nhau: một cho mã hóa và một để giải mã. Để thuật toán đảm bảo an toàn thì không thể tìm được khóa giải mã nếu chỉ biết khóa đã dùng mã hóa. Điều này còn được gọi là mã hóa công khai vì khóa dùng để mã hóa có thể công bố công khai mà không ảnh hưởng đến bí mật của văn bản mã hóa.

Các thông tin để mở khóa thì chỉ có người sở hữu mới biết. Tồn tại khả năng một người nào đó có thể tìm ra được khóa bí mật. Không giống với hệ thống mật mã sử dụng một lần (*one-time pad*) hoặc tương đương, chưa có thuật toán mã hóa khóa bất đối xứng nào được chứng minh là an toàn trước các tấn công dựa trên bản chất toán học của thuật toán. Khả năng một mối quan hệ nào đó giữa 2 khóa hay điểm yếu của thuật toán dẫn tới cho phép giải mã không cần tới khóa hay chỉ cần khóa mã hóa vẫn chưa được loại trừ. An toàn của các thuật toán này đều dựa trên các ước lượng về khối lượng tính toán để giải các bài toán gắn với chúng. Các ước lượng này lại luôn thay đổi tùy thuộc khả năng của máy tính và các phát hiện toán học mới.

**Ưu điểm :**

* Chỉ có Private key cần được giữ bí mật với điều kiện việc công khai Public key cần được đảm bảo.
* Cặp khóa riêng và công khai có thể sử dụng trong thời gian dài.
* Nhiều mô hình khóa công cộng được phát triển hình thành nên các kỹ thuật chữ ký số hiệu quả.
* Trong mạng lớn số lượng khóa quan tâm it hơn nhiều so với việc dùng khóa đối xứng.

**Nhược điểm:**

* Tốc độ cho các phương thức mã hóa công khai chậm hơn rất nhiều so với các mô hình mã hóa đối xứng.
* Kích thước khóa lớn hơn rất nhiều so với mã hóa đối xứng.
* Không có mô hình mã hóa công khai nào được chứng minh là an toàn.
* Phần lớn các mô hình mã hóa hiện nay có sự an toàn dựa trên các giả thuyết của của một tập nhỏ các vấn để lý thuyết số học .
* Không có bề dày lâu đời như hệ mã hóa đối xứng , nó chỉ được tìm ra vào những năm 1970.

## Hàm băm MD5

## Tổng quan về hàm băm

Trong ngành mật mã học, một hàm băm mật mã học (tiếng Anh: Cryptographic hash function) là một hàm băm với một số tính chất bảo mật nhất định để phù hợp việc sử dụng trong nhiều ứng dụng bảo mật thông tin đa dạng, chẳng hạn như chứng thực và kiểm tra tính nguyên vẹn của thông điệp. Một hàm băm nhận đầu vào là một xâu ký tự dài (hay thông điệp) có độ dài tùy ý và tạo ra kết quả là một xâu ký tự có độ dài cố định, đôi khi được gọi là tóm tắt thông điệp (message digest) hoặc chữ ký số (digital fingerprint)

Hàm băm là các thuật toán không sử dụng khóa để mã hóa (ở đây ta dùng thuật ngữ “băm" thay cho "mã hóa"), nó có nhiệm vụ “lọc" (băm) thông điệp được đưa vào theo một thuật toán h một chiều nào đó, rồi đưa ra một bản băm gọi là văn bản đại diện có kích thước cố định. Do đó người nhận không biết được nội dung hay độ dài ban đầu của thông điệp đã được băm bằng hàm băm. Giá trị của hàm băm là duy nhất, và không thể suy ngược lại được nội dung thông điệp từ giá trị băm này.

**Tính chất của hàm băm**

*Tính đụng độ:* Theo nguyên lý ***Diricle***: nếu có (*n+1)* con thỏ được bỏ vào n cái chuồng thì phải tồn tại ít nhất một cái chuồng mà trong đó có ít nhất là hai con thỏ ở chung. Rõ ràng với không gian giá trị Băm nhỏ hơn rất nhiều so với không gian tin về mặt kích thước thì chắc chắn sẽ tồn tại đụng độ, nghĩa là có hai tin *x # x’* mà giá trị Băm của chúng là giống nhau, tức *h(x) = h(x’)*

Sau đây chúng ta sẽ xét các dạng tấn công có thể có, từ đó rút ra các tính chất của hàm Băm:

**Tính chất 1: Hàm băm không va chạm yếu.**

Hàm băm h là không va chạm yếu nếu khi cho trước một bức điện x, không thể tiến hành về mặt tính toán để tìm ra một bức điện *x’  x* mà *h(x’) = h(x).*

Ví dụ: Người A gửi cho B (*x*, *y*) với *y = SigA(h(x)).* Nhưng trên đường truyền, tin bị lấy trộm. Tên trộm, bằng cách nào đó tìm được một bản thông điệp *x*’ có *h*(*x*’) = *h*(*x*) mà *x*’  *x*. Sau đó, tên trộm đưa *x*’ thay thế *x* rồi truyền tiếp cho người B. Người B nhận được và vẫn xác thực được thông tin đúng đắn.

Để tránh tấn công trên, hàm băm phải *không va chạm yếu.*

**Tính chất 2: Hàm băm không va chạm mạnh**

Hàm băm h là không va chạm mạnh nếu không có khả năng tính toán để tìm ra hai bức thông điệp x và x’ mà x  x’ và h(x) = h(x’).

Ví dụ: Đầu tiên, tên giả mạo tìm ra được hai bức thông điệp x’ và x (x’  x) mà có h(x’) = h(x) (ta coi bức thông điệp x là hợp lệ, còn x’ là giả mạo). Tiếp theo, tên trộm đưa cho ông A và thuyết phục ông này kí vào bản tóm lược h(x) để nhận được y. Khi đó (x’, y) là bức điện giả mạo nhưng hợp lệ.

Để tránh kiểu tấn công này, hàm h phải thỏa mãn tính không va chạm mạnh

**Tính chất 3: Hàm băm một chiều.**

Hàm băm *h* là một chiều nếu khi cho trước một bản tóm lược thông báo z, không thể thực hiện về mặt tính toán để tìm bức điện x sao cho *h(x) = z.*

Việc giả mạo các chữ kí trên bản tóm lược thông báo *z* ngẫu nhiên thường xảy ra với sơ đồ chữ kí. Giả sử tên giả mạo tính chữ kí trên bản tóm lược thông báo *z* ngẫu nhiên như vậy. Sau đó anh ta tìm *x* sao cho *z = h(x).* Nếu làm được như vậy thì *(x,y)* là bức điện giả mạo hợp lệ. Để tránh được tấn công này, h cần thoả mãn tính chất một chiều:

## b)Hàm băm MD5

Ronald Rivest là người đã phát minh ra các hàm Băm MD2, MD4 (1990) và MD5 (1991). Do tính chất tương tự của các hàm Băm này, sau đây chúng ta sẽ xem xét hàm Băm MD5, đây là một cải tiến của MD4 và là hàm Băm được sử dung rộng rãi nhất, nguyên tắc thiết kế của hàm băm này cũng là nguyên tắc chung cho rất nhiều các hàm băm khác

Đầu vào hàm băm là những khối 512 *bit*, được chia cho 16 khối con 32 *bit*. Đầu ra của thuật toán là một thiết lập của 4 khối 32 *bit* để tạo thành một hàm Băm 128 *bit* duy nhất.

Đầu tiên, ta chia bức điện thành các khối 512 *bit*, với khối cuối cùng (*đặt là x và x < 512bit*) của bức điện, chúng ta cộng thêm một bit 1 vào cuối của x, theo sau đó là các bit 0 để được độ dài cần thiết (512 *bit*). Kết quả là bức điện vào là một chuỗi M có độ dài chia hết cho 512, vì vậy ta có thể chia M ra thành các N khối con 32 *bit* (*N khối này sẽ chia hết cho 16*).

Bây giờ, ta bắt đầu tìm cốt của bức điện với 4 khối 32 *bit* A, B, C và D (*được xem như thanh ghi*) :

A = 0x01234567

B = 0x89abcdef C = 0xfedcba98 D = 0x76543210.

Người ta thường gọi A, B, C, D là các chuỗi biến số (*chaining variables*).

Bức điện được chia ra thành nhiều khối 512 *bit*, mỗi khối 512 *bit* lại được chia ra 16 khối 32 *bit* đi vào bốn vòng lặp của MD5. Giả sử ta đặt a, b, c và d thay cho A, B, C và D đối với khối 512 *bit* đầu tiên của bức điện. Bốn vòng lặp trong MD5 đều có cấu trúc giống nhau. Mỗi vòng thực hiện 16 lần biến đổi: thực hiện với một hàm phi tuyến của 3 trong 4 giá trị a, b, c và d; sau đó nó cộng kết quả đến giá trị thứ 4, tiếp đó cộng với một khối con 32 *bit* và một hằng số. Sau đó, nó dịch trái một lượng bit thay đổi và cộng kết quả vào một trong 4 giá trị a, b, c hay d. Kết quả cuối cùng là một giá trị mới được thay thế một trong 4 giá trị a, b, c hay d.

|  |
| --- |
| **A** |
| **B** |
| **C** |
| **D** |

**Khối của bức điện**

**Vòng 4**

**Vòng 3**

**Vòng 2**

**Vòng 1**

|  |
| --- |
| **A** |
| **B** |
| **C** |
| **D** |

*Hình 1.2 sơ đồ vòng lặp chính của MD5*

Có bốn hàm phi tuyến, mỗi hàm này được sử dụng cho mỗi vòng:

*F(X,Y,Z ) = (X and Y) or ((not X) and Z)*

*G(X,Y,Z ) = ((X and Z) or (Y and (not Z))) H(X,Y,Z ) = X xor Y xor Z*

*I(X,Y,Z ) = Y xor (X or (not Z)).*

Những hàm này được thiết kế sao cho các bit tương ứng của X, Y và Z là độc lập và không ưu tiên, và mỗi bit của kết quả cũng độc lập và ngang bằng nhau.

Nếu Mj là một biểu diễn của khối con thứ j (j = 16) và <<<s là phép dịch trái của s bit, thì các vòng lặp có thể biểu diễn như sau:

*FF(a,b,c,d,Mj,s,ti) được biểu diễn a = b + ((a + F(b,c,d) + Mj + ti) <<< s) GG(a,b,c,d,Mj,s,ti) được biểu diễn a = b + ((a + G(b,c,d) + Mj + ti) <<< s) HH(a,b,c,d,Mj,s,ti) được biểu diễn a = b + ((a + H(b,c,d) + Mj + ti) <<< s) II(a,b,c,d,Mj,s,ti) được biểu diễn a = b + ((a + I(b,c,d) + Mj + ti) <<< s).*

*Bốn vòng (64 bước) sẽ thực hiện như sau:*

*Vòng 1:*

*FF (a, b, c, d, M0, 7, 0x76aa478) FF (d, a, b, c, M1, 12, 0xe8c7b756) FF (c, d, a, b, M2, 17, 0x242070db)*

*FF (b, c, d, a, M3, 22, 0xc1bdceee) FF (a, b, c, d, M4, 7, 0xf57c0faf) FF (d, a, b, c, M5, 12, 0x4787c62a) FF (c, d, a, b, M6, 17, 0xa8304613) FF (b, c, d, a, M7, 22, 0xfd469501) FF (a, b, c, d, M8, 7, 0x698098d8)*

*FF (d, a, b, c, M9, 12, 0x8b44f7af) FF (c, d, a, b, M10, 17, 0xffff5bb1) FF (b, c, d, a, M11, 22, 0x895cd7be) FF (a, b, c, d, M12, 7, 0x6b901122) FF (d, a, b, c, M13, 12, 0xfd987193) FF (c, d, a, b, M14, 17, 0xa679438e) FF (b, c, d, a, M15, 22, 0x49b40821).*

*Vòng 2:*

*GG (a, b, c, d, M1, 5, 0x61e2562) GG (d, a, b, c, M6, 9, 0xc040b340) GG (c, d, a, b, M11, 14, 0x265e5a51)*

*GG (b, c, d, a, M0, 20, 0xe9b6c7aa) GG (a, b, c, d, M5, 5, 0xd62f105d) GG (d, a, b, c, M10, 9, 0x02441453) GG (c, d, a, b, M15, 14, 0xd8a1e681)*

*GG (b, c, d, a, M4, 20, 0xe7d3fbc8) GG (a, b, c, d, M9, 5, 0x21e1cde6) GG (d, a, b, c, M14, 9, 0xc33707d6) GG (c, d, a, b, M3, 14, 0xf4d50d87) GG (b, c, d, a, M8, 20, 0x455a14ed)*

*GG (a, b, c, d, M13, 5, 0xa9e3e905)*

*GG (d, a, b, c, M2, 9, 0xfcefa3f8) GG (c, d, a, b, M7, 14, 0x676f02d9)*

*GG (b, c, d, a, M12, 20, 0x8d2a4c8a). Vòng 3:*

*HH (a, b, c, d, M5, 4, 0xfffa3942) HH (d, a, b, c, M8, 11, 0x8771f681) HH (c, d, a, b, M11, 16, 0x6d9d6122)*

*HH (b, c, d, a, M14, 23, 0xfde5380c) HH (a, b, c, d, M1, 4, 0xa4beea44) HH (d, a, b, c, M4, 11, 0x4bdecfa9) HH (c, d, a, b, M7, 16, 0xf6bb4b60) HH (b, c, d, a, M10, 23, 0xbebfbc70) HH (a, b, c, d, M13, 4, 0x289b7ec6)*

*HH (d, a, b, c, M0, 11, 0xeaa127fa) HH (c, d, a, b, M3, 16, 0xd4ef3085) HH (b, c, d, a, M6, 23, 0x04881d05) HH (a, b, c, d, M9, 4, 0xd9d4d039)*

*HH (d, a, b, c, M12, 11, 0xe6db99e5) HH (c, d, a, b, M15, 16, 0x1fa27cf8) HH (b, c, d, a, M2, 23, 0xc4ac5665).*

*Vòng 4:*

*II (a, b, c, d, M0, 6, 0xf4292244) II (d, a, b, c, M7, 10, 0x432aff97)*

*II (c, d, a, b, M14, 15, 0xab9423a7) II (b, c, d, a, M5, 21, 0xfc93a039) II (a, b, c, d, M12, 6, 0x655b59c3)*

*II (d, a, b, c, M3, 10, 0x8f0ccc92) II (c, d, a, b, M10, 15, 0xffeff47d) II (b, c, d, a, M1, 21, 0x85845dd1)*

*II (a, b, c, d, M8, 6, 0x6fa87e4f)*

*II (d, a, b, c, M15, 10, 0xfe2ce6e0) II (c, d, a, b, M6, 15, 0xa3013414) II (b, c, d, a, M13, 21, 0x4e0811a1) II (a, b, c, d, M4, 6, 0xf7537e82)*

*II (d, a, b, c, M11, 10, 0xbd3af235) II (c, d, a, b, M2, 15, 0x2ad7d2bb) II (b, c, d, a, M9, 21, 0xeb86d391).*

Những hằng số ti được chọn theo quy luật sau: ở bước thứ i giá trị ti là phần nguyên của 232\*abs(sin(i)), trong đó i = [0..63] được tính theo *radian*.

Sau tất cả những bước này a, b, c và d lần lượt được cộng với A, B, C và D để cho kết quả đầu ra, và thuật toán tiếp tục với khối dữ liệu 512 *bit* tiếp theo cho đến hết bức điện. Đầu ra cuối cùng là một khối 128 *bit* của A, B, C và D, đây chính là hàm Băm nhận được.

**Tính bảo mật trong MD5:**

*Ron Rivest* đã phác hoạ những cải tiến của **MD5** so với **MD4** như sau: Vòng thứ 4 được thêm vào (*còn* ***MD4*** *chỉ có 3 vòng*).

Mỗi bước được cộng thêm một hằng số duy nhất.

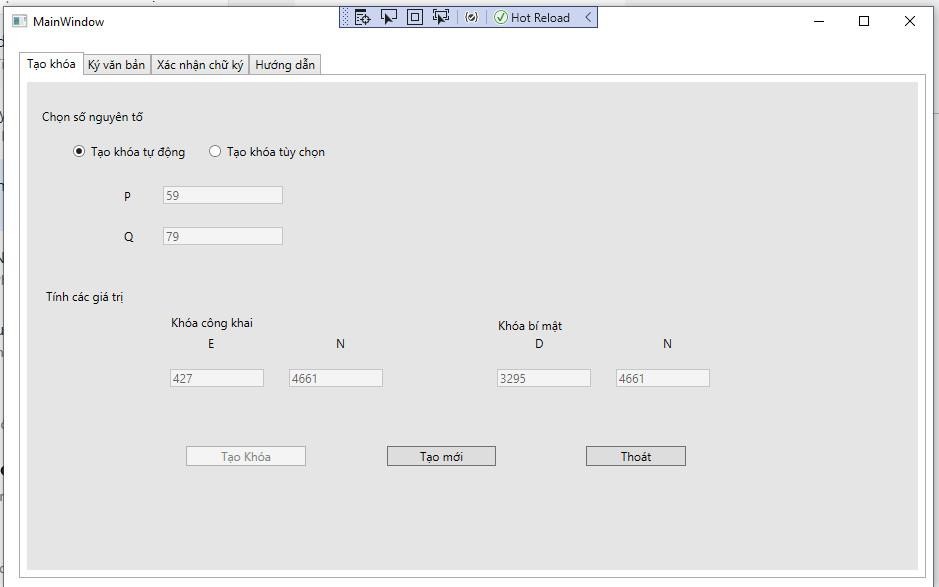
Hàm G ở vòng 2 thay đổi từ *((X and Y) or (X and Z) or (Y and Z))* thành *((X and*

*Z) or (Y and (not Z)))* nhằm giảm tính đối xứng của G (*giảm tính tuyến tính*). Mỗi bước được cộng kết quả của bước trước nó, làm các quá trình có tính liên

## Thiết kế, cài đặt chương trình demo thuật toán

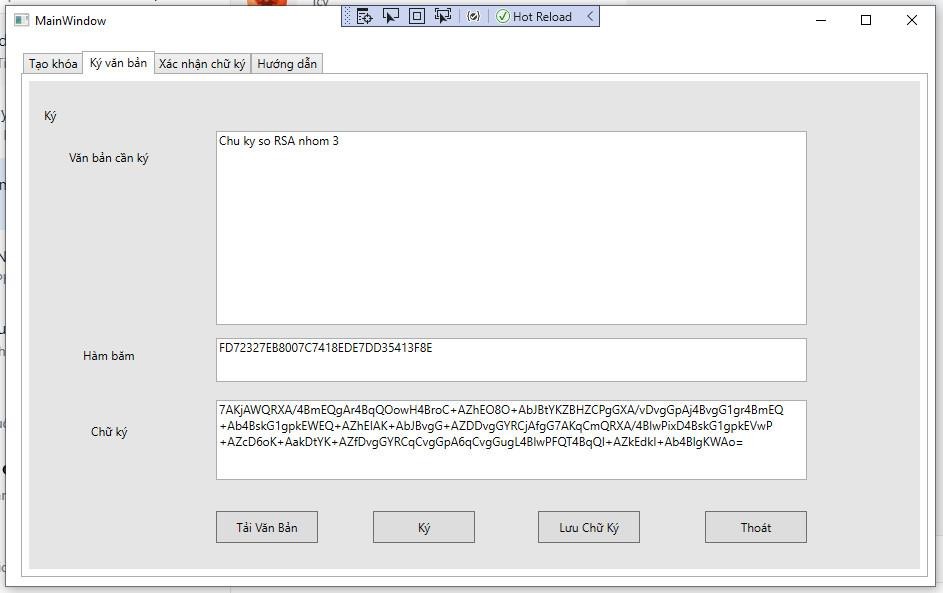
## Giao diện chương trình demo

**Bước 1 :** Tại giao diện khóa chúng ta nhấn vào phần tạo khóa để sinh ra khóa công khai và khóa bí mật (Có thể tạo khóa tự động hoặc tạo khóa tùy chọn)



**Bước 2 :** Tại giao diện ký văn bản ta lần lượt thực hiện các bước sau :

* Tải văn bản cần ký.
* Ký vào văn bản đó.
* Sau khi đã ký xong thì lưu lại chữ ký.

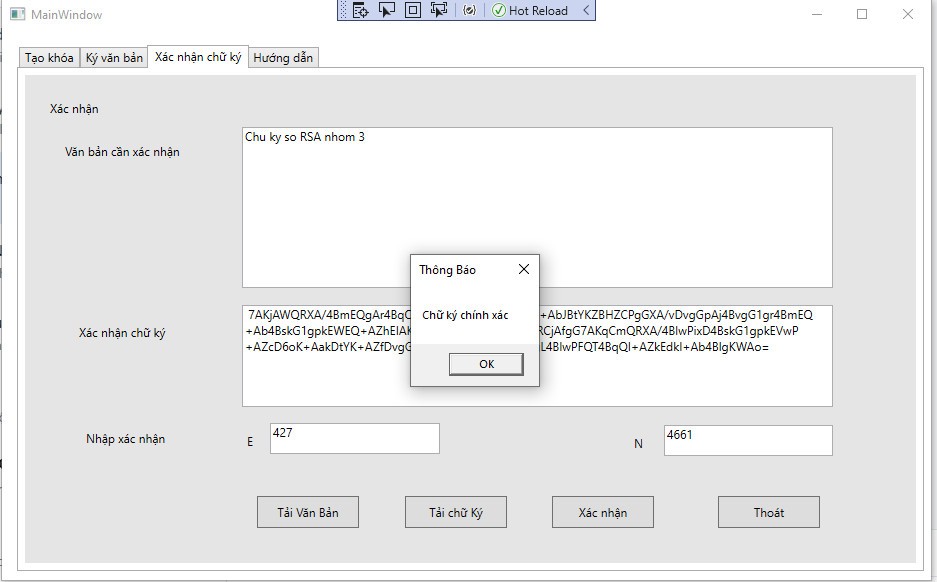


**Bước 3 :** Tại giao diện xác nhận văn bản ta thực hiện các bước

* Tải văn bản đã ký và cần xác nhận lên.
* Tải chữ ký kèm theo đã được ký ở văn bản.
* Xác nhận chữ ký.

+ Nếu văn bản đã được chỉnh sửa hoặc chữ ký kèm theo không chính xác sẽ xuất ra thông báo là văn bản đã được chỉnh sửa hoặc chữ ký không chính xác.

+ Nếu văn bản và chữ ký đều chính xác thì chương trình sẽ thông báo chữ ký đã chính xác.



## 2.3.1. Cài đặt chương trình demo

**Các yêu cầu của ứng dụng :**

* Tạo ra cặp khoá: khoá công khai và khoá bí mật bằng hệ mật mã khoá công khai RSA.
* Thực hiện ký chữ ký số lên thông điệp bằng cách dùng khoá bí mật của người ký.
* Thực hiện việc chứng thực chữ ký số bằng cách dùng khoá công khai của người đã ký lên thông điệp.
* Dùng khoá công khai của người nhận để thực hiện quá trình mã hoá thông điệp hoặc tập tin.
* Người nhận dùng khoá bí mật của mình để thực hiện quá trình giải mã thông điệp hoặc tập tin.

**Môi trường xây dựng ứng dụng**

* Sử dụng thuật toán băm MD5 để băm thông điệp trước khi thực hiện ký.
* Sử dụng hệ mật mã khoá công khai RSA để thực hiện sinh khoá cho hệ thống.
* Dùng ngôn ngữ lập trình Java để viết mã cho chương trình.

## Quá trình ký và xác thực chữ ký số

* Ký văn bản số
* Xác thực chữ ký số
* Mã hoá tập tin
* Giải mã tập tin

## Thuyết minh chương trình

*Quá trình tạo cặp khóa bí mật và khóa công khai*

* + Tính module n
  + Sinh khóa e
  + Tính khóa d

*Quá trình tạo chữ ký số*

* + Số hóa thông điệp:
  + Tạo và lưu chữ ký số:

*Quá trình xác thực chữ ký*

Khi cá thể B nhận được chữ ký từ A, B:

* Nếu đúng là chữ ký của A thì sẽ nhận được thông báo “Chữ ký đã được xác thực”.
* Nếu không đúng là chữ ký của A hoặc không đúng nội dung (Nội dung thông điệp hay Tập tin đính kèm) thì sẽ nhận được thông báo “Thông điệp hoặc tập tin đính kèm đã bị thay đổi”.

*Mã hóa tập tin bằng khóa công khai Giải mã tập tin băng khóa bí mật*

## Cài đặt và triển khai

Đề tài nghiên cứu cài đặt chương trình demo với 3 ngôn ngữ JavaScript, C#, Java, tương ứng với 2 công cụ: Visual Studio, Netbean.

## Visual Studio

Visual Studio là một môi trường phát triển tích hợp (IDE) từ Microsoft. Nó đóng vai trò như một công cụ hỗ trợ lập trình chuyên nghiệp. Tính năng chính của Visual Studio 2019 trên Microsoft Windows tạo nên các trang web, các ứng dụng web, và dịch web hay thiết kế cơ sở dữ liệu.

* + Yêu cầu của cấu hình máy tính:
* Hệ điều hành: Windows 7 trở lên.
* CPU: 1.8 GHz trở lên.
* RAM: Khuyến nghị RAM 4GB (tối thiểu 2,5 GB nếu chạy trên máy ảo).
* Dung lượng ổ cứng: 1GB đến 40GB tùy thuộc vào số tính năng cài đặt.

**Hướng dẫn cài đặt Visual Studio**

Để tải xuống và cài đặt Visual Studio, hãy làm theo các bước sau:

1. Tải file cài đặt Visual Studio tương ứng với thiết bị
2. Ấn vào bản vừa tải xuống > Chọn Continue.
3. Chọn tính năng mà bạn muốn cài (.NET Framework, Visual C++, C++ for Linux, .NET Core,…) > nhấn Install.
4. Chờ tiến trình hoàn tất nhấn Launch.

## Netbeans

NetBeans là một [môi trường phát triển tích hợp](https://vi.wikipedia.org/wiki/M%C3%B4i_tr%C6%B0%E1%BB%9Dng_ph%C3%A1t_tri%E1%BB%83n_t%C3%ADch_h%E1%BB%A3p) (IDE) cho [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)). NetBeans cho phép các ứng dụng được phát triển từ một tập hợp các thành phần phần mềm được gọi là *modules*. NetBeans chạy trên [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [macOS](https://vi.wikipedia.org/wiki/MacOS), [Linux](https://vi.wikipedia.org/wiki/Linux) và [Solaris](https://vi.wikipedia.org/wiki/Solaris_(h%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh)).

Ngoài việc phát triển Java, nó còn có các phần mở rộng cho các ngôn ngữ khácnhư [PHP](https://vi.wikipedia.org/wiki/PHP), [C](https://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [C++](https://vi.wikipedia.org/wiki/C%2B%2B), [HTML5](https://vi.wikipedia.org/wiki/HTML5) và [JavaScript](https://vi.wikipedia.org/wiki/JavaScript). Tuy nhiên phần mềm có dung lượng khá là nặng dành cho các máy cấu hình có RAM, CPU tương đối cao để vận hành.

**Hướng dẫn cài đặt Netbeans:**

Để tải xuống và cài đặt Netbeans, hãy làm theo các bước sau:

1. Tải file cài đặt Netbeans tương ứng với thiết bị.
2. Sau khi tải xong, nhấp chuột phải vào biểu tượng NetBeans IDE và chọn **Run as administrator.**
3. Giao diện cài đặt hiện lên hiển thị các ngôn ngữ lập trình được hỗ trợ. Các bạn có thể vào mục **Customize** để chọn thêm hoặc bớt ngôn ngữ. Chọn **Next** để tiếp tục quá trình cài đặ
4. Tích vào ô **I accept the terms in the license agreement** và chọn **Next**.
5. Bộ cài sẽ tự động tìm nơi cài đặt NetBeans IDE và tìm tới nơi lưu trữ JDK mới cài đặt. Nếu muốn thay đổi ổ đĩa cài đặt, bạn có thể đổi lại bằng cách chọn **Browse** và chọn nơi muốn cài. Sau đó chọn **Next** để tiếp tục quá trình cài đặt.
6. Tích vào ô kiểm **Check for Update** để kiểm tra NetBeans có phiên bản cập nhật tức thời. Chọn **Install** để tiến hành quá trình cài đặt.
7. Chọn **Finish** để kết thúc quá trình cài đặt NetBeans IDE.

## Thực hiện bài toán

### Phân công công việc

|  |  |
| --- | --- |
| Tên sinh viên | Tên công việc |
| 1.Nguyễn Thành Đạt | * Tìm hiểu về hàm băm MD5 (Chương 2) * Kết luận kiến thức lĩnh hội và bài học kinh nghiệm ( Chương 3) * Tìm hiều về công cụ Netbeans * Cài đặt và chạy demo chương trình bằng ngôn ngữ Java |
| 2. 2. Nguyễn Bá Đông | - Tìm hiểu về hàm băm  - Tìm hiểu về công cụ Visual Studio code  - Cài đặt và chạy demo chương trình bằng ngôn ngữ C# |
| 3. Nguyễn Tiến Đại | - Tìm hiểu phương pháp mã hóa RSA – chữ ký điện tử RSA  - Cài đặt và chạy demo chương trình bằng ngôn ngữ Python |
| 4. Nguyễn Trọng Đạt | - Tìm hiểu phương pháp mã hóa RSA – chữ ký điện tử RSA  - Cài đặt và chạy demo chương trình bằng ngôn ngữ PHP |
| 5.Nguyễn Đức Đăng | - Tìm hiểu phương pháp mã hóa RSA – chữ ký điện tử RSA  - Cài đặt và chạy demo chương trình bằng ngôn ngữ JavaScript |

### 2.5.2. Nguyễn Thành Đạt

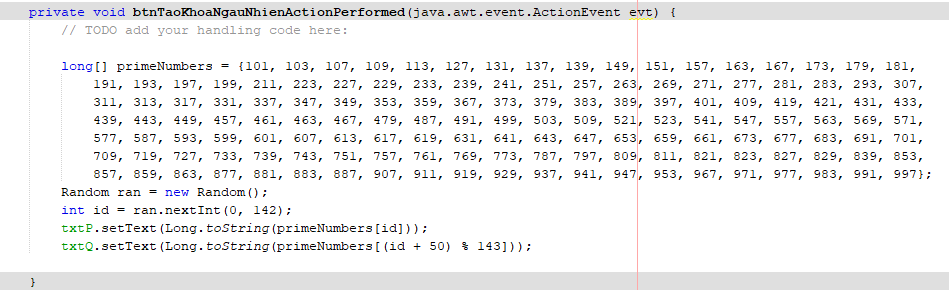
* Tìm hiểu về hàm băm MD5 (Chương 2)
* Kết luận kiến thức lĩnh hội và bài học kinh nghiệm ( Chương 3)
* Tìm hiều về công cụ NetBeans
* Cài đặt và chạy demo chương trình bằng ngôn ngữ Java.

**Một số thư viện đã sử dụng:**

* Jaxb-api là kiến trúc java cho liên kết XML, việc truy cập các tài liệu XML từ các ứng dụng được bằng ngôn ngữ lập trình Java sẽ dễ dàng hơn.

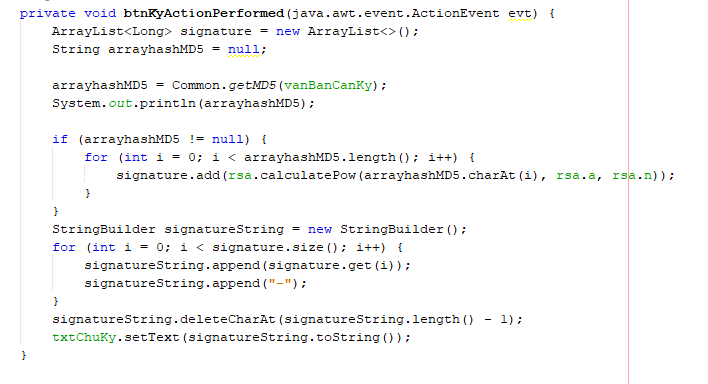
**Demo code**

*Tạo khóa*

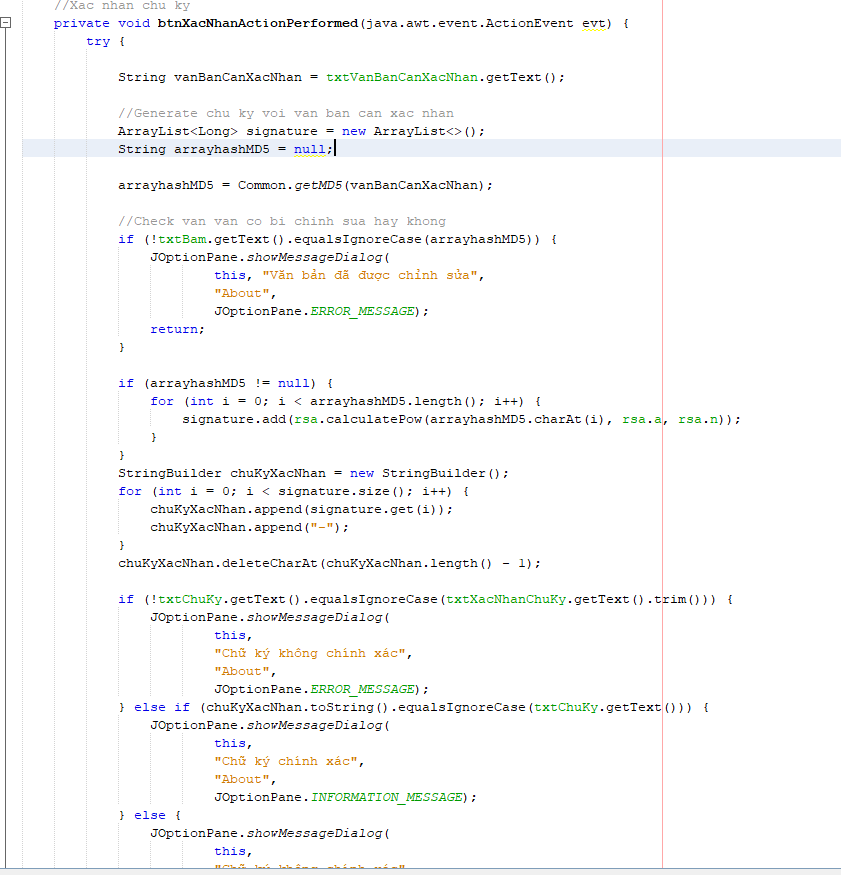
**

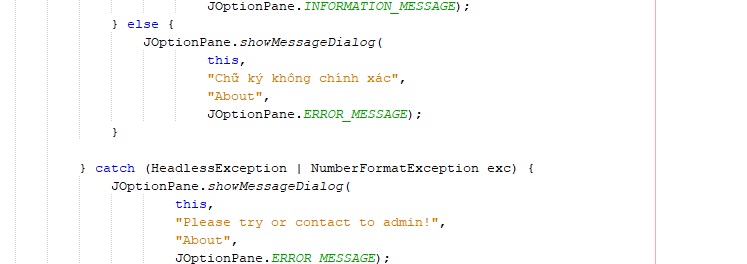


* + *Ký chữ ký, mã hóa*

**

* + *Xác thực chữ ký, giải mã*





### 2.5.3. Nguyễn Bá Đông

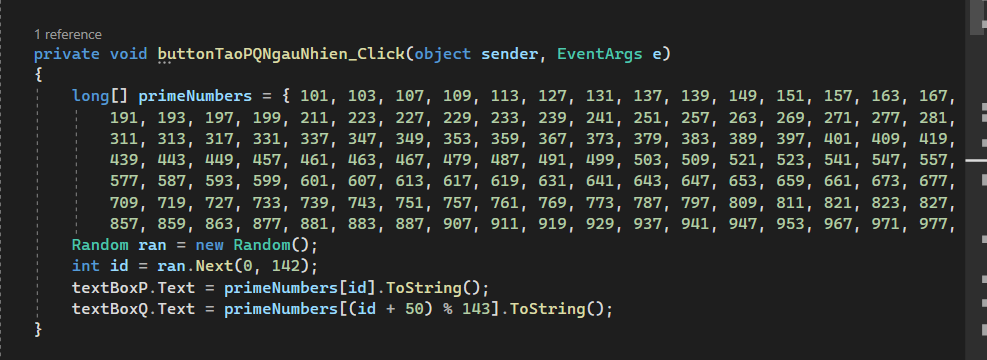
* Tìm hiểu về hàm băm MD5 (Chương 2)
* Cài đặt và chạy demo chương trình bằng ngôn ngữ C#

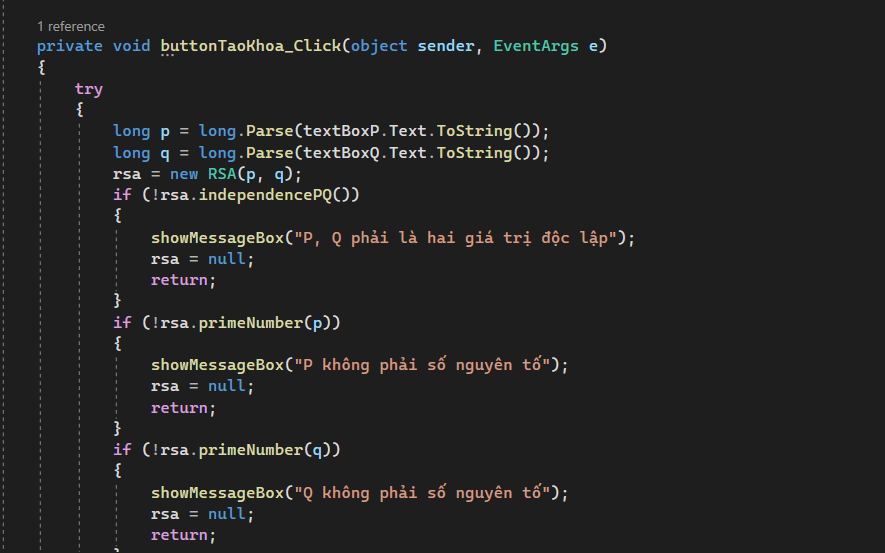
**Một số thư viện đã sử dụng:**

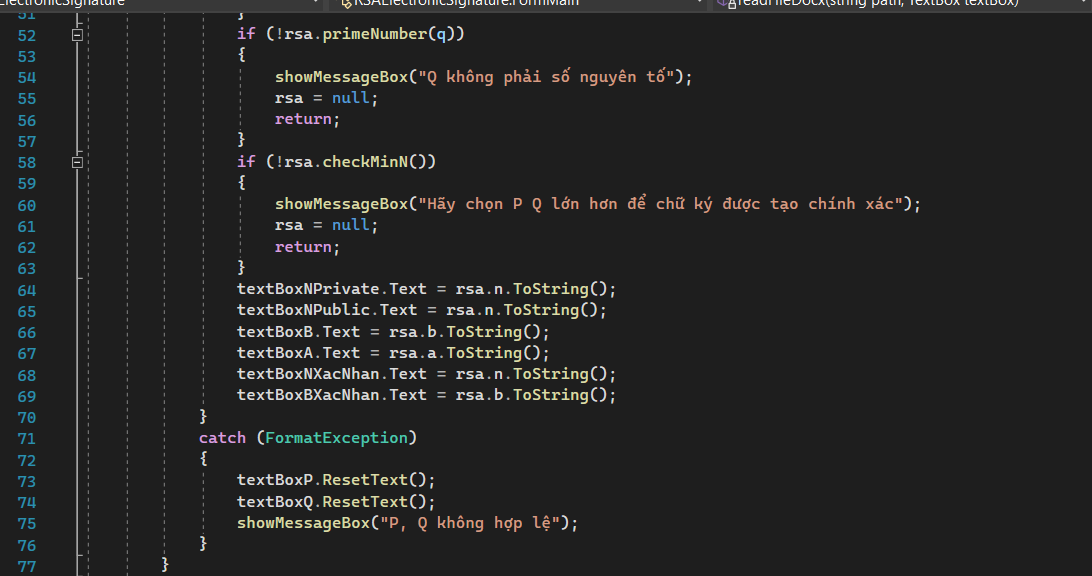
* Jaxb-api là kiến trúc java cho liên kết XML, việc truy cập các tài liệu XML từ các ứng dụng được bằng ngôn ngữ lập trình Java sẽ dễ dàng hơn.

**Demo code**

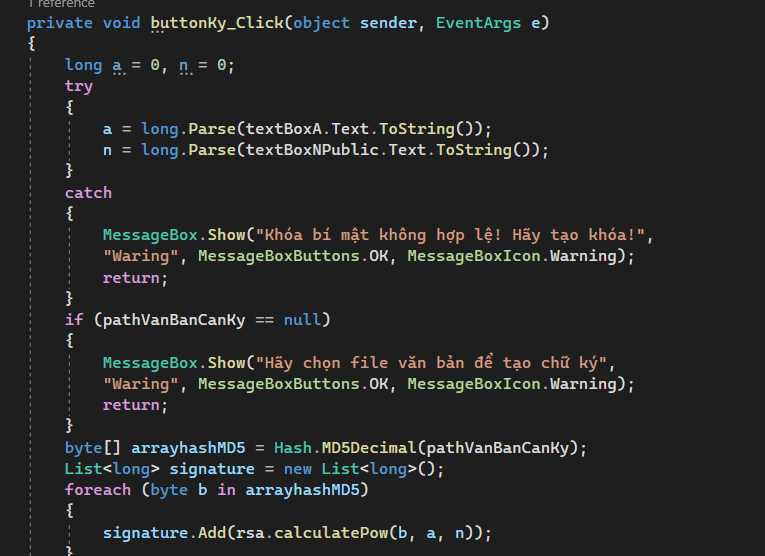
*Tạo khóa*

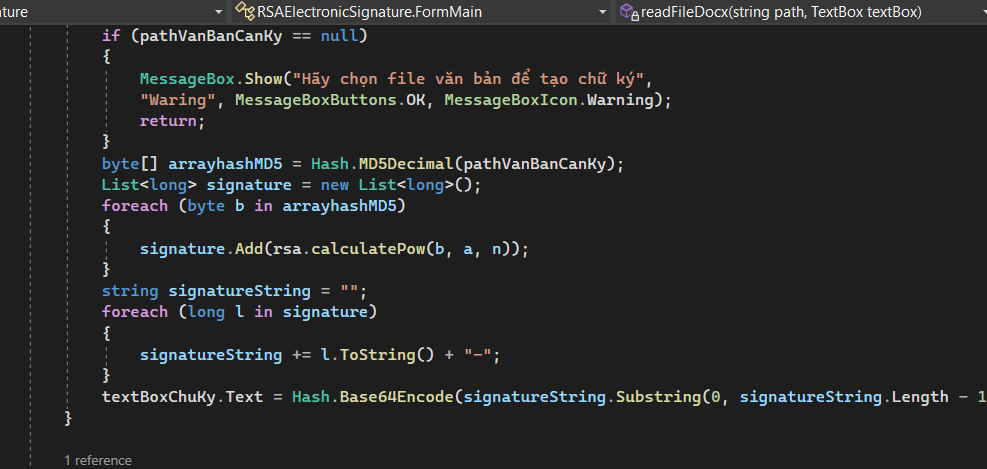
**



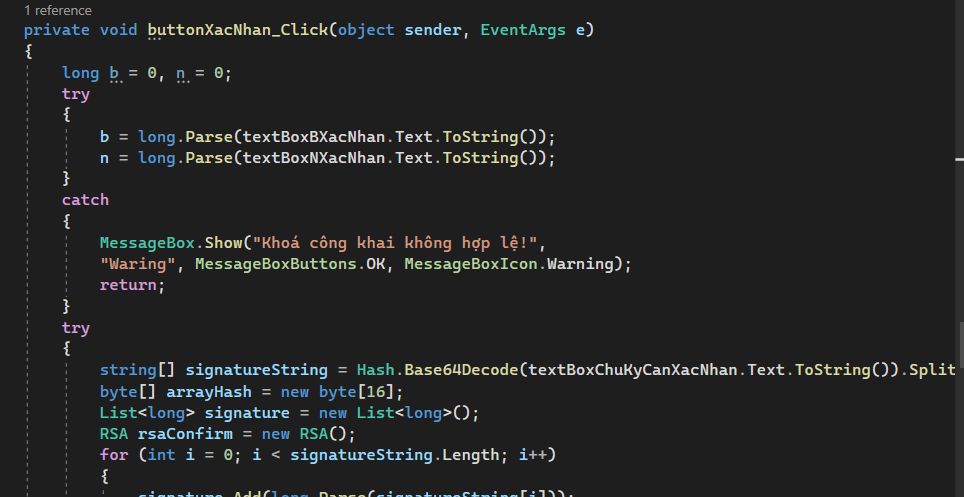


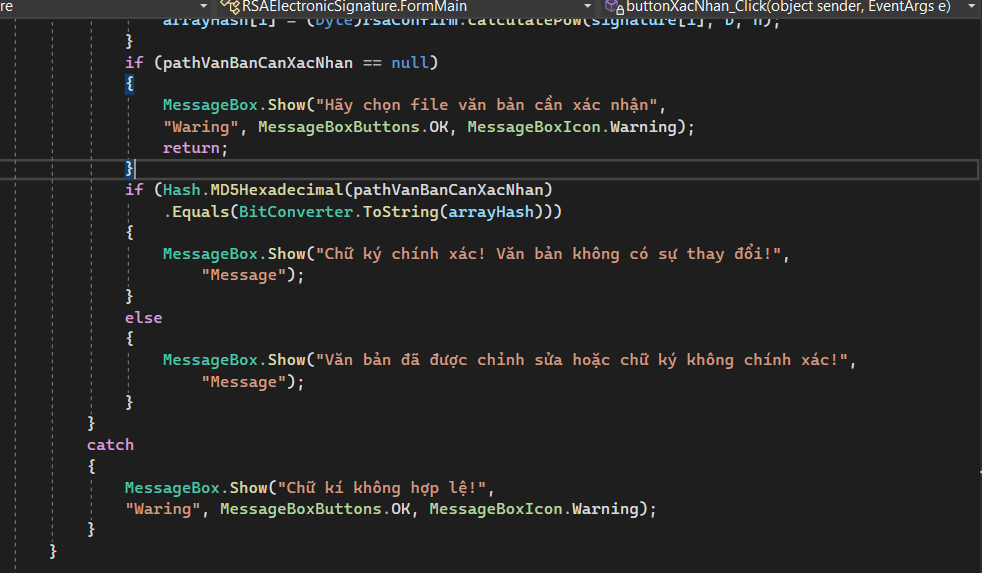
* + *Ký chữ ký, mã hóa*

**

**

* + *Xác thực chữ ký, giải mã*



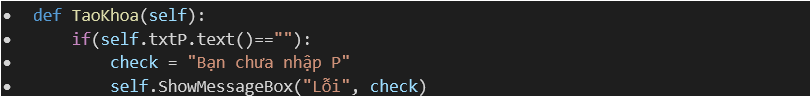


### 2.5.4. Nguyễn Tiến Đại

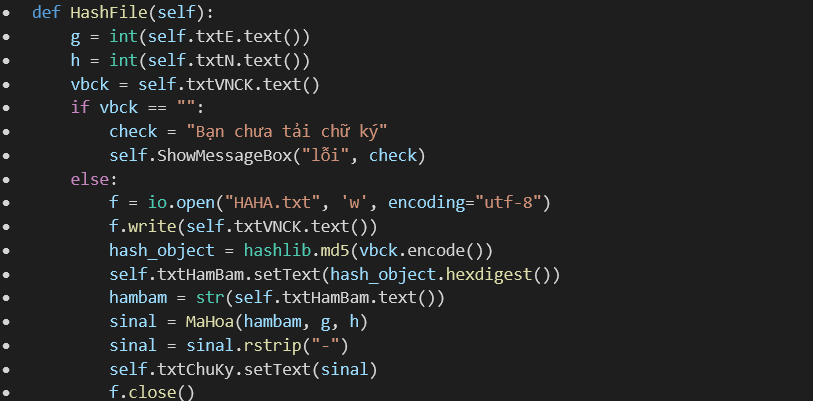
- Tìm hiểu phương pháp mã hóa RSA – chữ ký điện tử RSA.

- Cài đặt và chạy demo chương trình bằng ngôn ngữ Python.

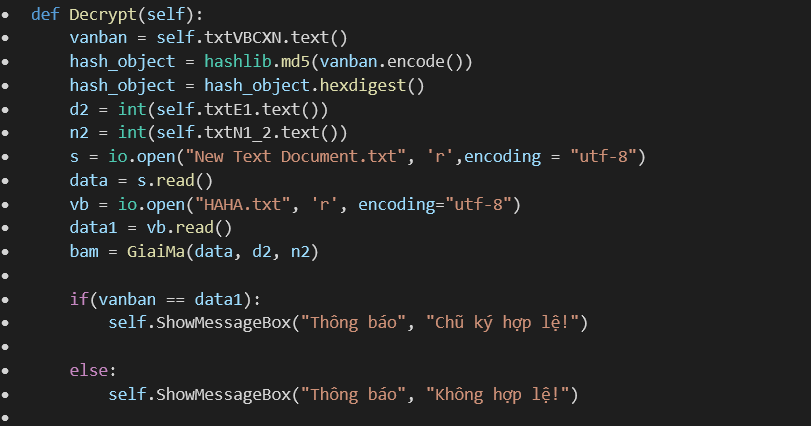
* *Demo code tạo khóa:*



* *Chữ ký, mã hóa:*

**

* *Xác thực chữ ký, giải mã:*



### 2.5.5. Nguyễn Trọng Đạt

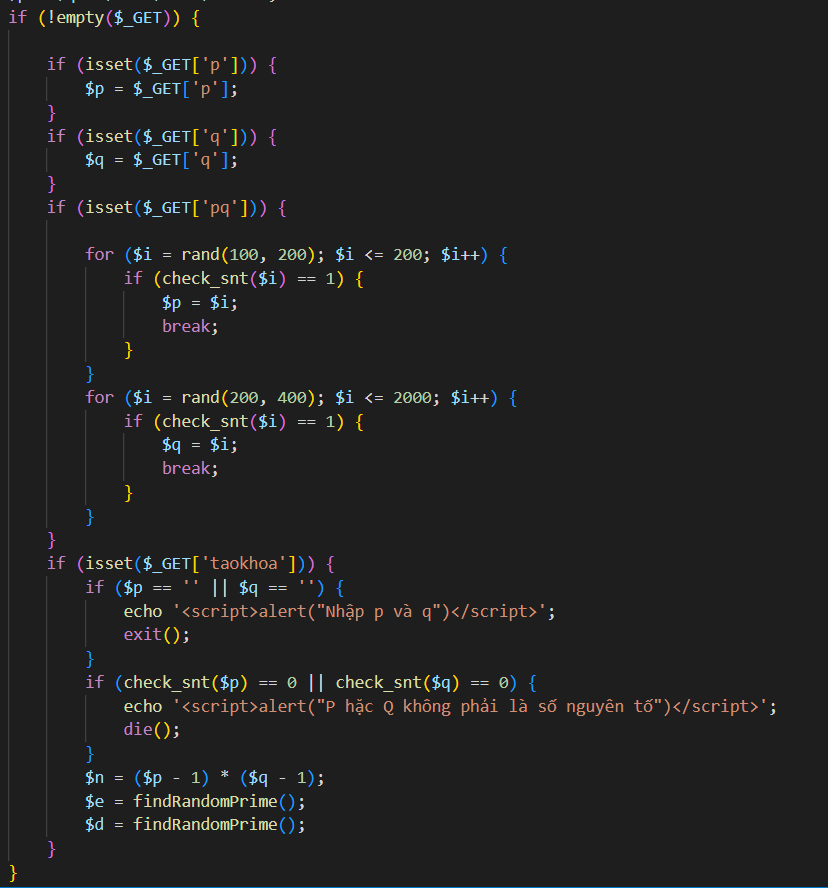
- Tìm hiểu phương pháp mã hóa RSA – chữ ký điện tử RSA.

- Cài đặt và chạy demo chương trình bằng ngôn ngữ PHP

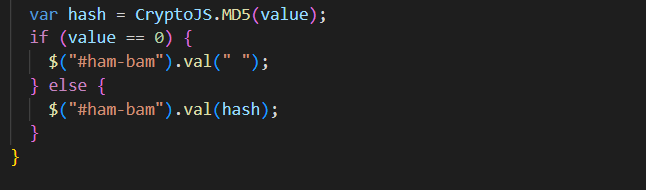
- Thư viện được sử dụng

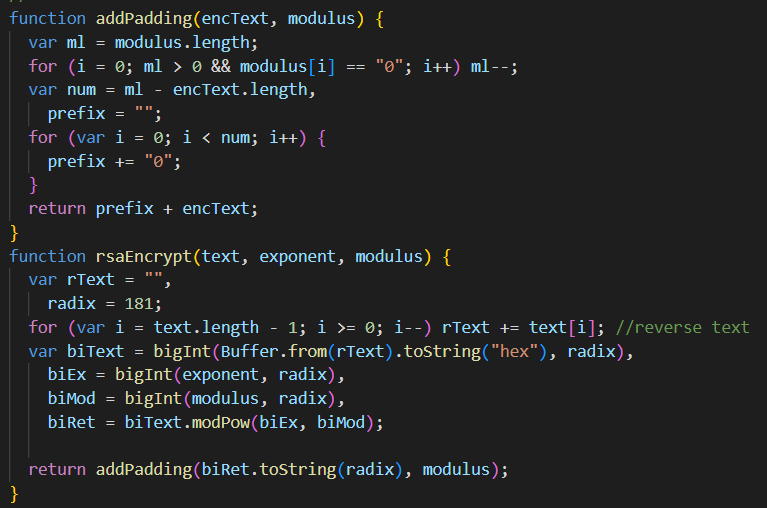
BigInteger : cung cấp một lớp để quản lý các số nguyên có độ dài tù

* *Demo code tạo khoá*:

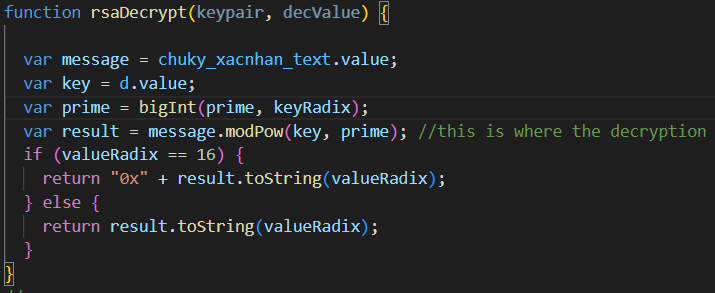


* *Chữ ký, mã hoá:*





* *Xác thực chữ ký, giải mã:*



### 2.5.6. Nguyễn Đức Đăng

- Tìm hiểu phương pháp mã hóa RSA – chữ ký điện tử RSA.

- Cài đặt và chạy demo chương trình bằng ngôn ngữ JavaScript.

* *Demo code tạo khoá*:
* const crypto = require("crypto");
* *Chữ ký, mã hoá:*

const data = "my secret data";

const encryptedData = crypto.publicEncrypt(

{

key: publicKey,

padding: crypto.constants.RSA\_PKCS1\_OAEP\_PADDING,

oaepHash: "sha256",

},

// We convert the data string to a buffer using `Buffer.from`

Buffer.from(data)

);

* *Xác thực chữ ký, giải mã:*

const decryptedData = crypto.privateDecrypt(

{

key: privateKey,

padding: crypto.constants.RSA\_PKCS1\_OAEP\_PADDING,

oaepHash: "sha256",

},

);

const verifiableData = "this need to be verified";

const signature = crypto.sign("sha256", Buffer.from(verifiableData), {

key: privateKey,

padding: crypto.constants.RSA\_PKCS1\_PSS\_PADDING,

});

console.log(signature.toString("base64"));

const isVerified = crypto.verify(

"sha256",

Buffer.from(verifiableData),

{

key: publicKey,

padding: crypto.constants.RSA\_PKCS1\_PSS\_PADDING,

},

signature

);

.

# Chương 3. Kiến thức lĩnh hội và bài học kinh nghiệm

## Nội dung đã thực hiện

Đề tài nghiên cứu về chữ kí điện tử sử dụng thuật toán RSA và viết ví dụ minh họa, trong đó hệ mã RSA được tập trung tìm hiểu bao gồm cách thức hoạt động, quy trình kí, quy trình kiểm tra chữ kí. Bên cạnh đó, nghiên cứu về phương pháp mã hóa bất đối xứng ứng dụng trong chữ kí điện tử, đưa ra được những ưu điểm, nhược điểm của phương pháp khi ứng dụng vào bài toán.

Một mặt quan trọng khác là hàm băm(MD5), chúng em đã được lĩnh hội thêm kiếm thức về các thuật toán không sử đụng khóa để mã hóa, tìm hiểu được tính chất của hàm băm và cách thức hoạt động của nó.

Từ đó có được những kiến thức cần thiết để xây dựng chương trình ứng dụng chữ kí điện tử đã đạt được một số kết quả như sau: Giới thiệu một cách khái quát ứng dụng của thuật toán RSA, phương pháp mã hóa bất đối xứng, kiến thức về hàm băm và cài đặt chương trình ứng dụng. “Chữ ký số và ứng dụng trong giao dịch hành chính điện tử ” để thực hiện các quá trình: Ký và xác thực chữ ký, mã hóa và giải mã các tập tin… giao dịch qua mạng.

**Những bài học học kinh nghiệm sau khi hoàn thành đề tài nhóm:**

* + Nắm rõ kỹ năng xác định vấn đề, kĩ năng phân tích vấn đề và sàng lọc ý kiến.
  + Nhóm trưởng cần xác định vai trò của từng thành viên, kiểm soát công việc tối ưu nhất và đưa ra những quyết định đúng đắn.
  + Xây dựng tiêu chí đánh giá công việc và thái độ làm việc của từng thành viên trong nhóm. Vừa đảm bảo tính khích lệ động viên, vừa đảm bảo tính ràng buộc về trách nhiệm đối với từng thành viên.
  + Biết lắng nghe, tôn trọng ý kiến của từng thành viên.
  + Đặt tinh thần trách nhiệm trong công việc thành ưu tiên hàng đầu.
  + Phát huy sự gắn kết, tạo sự đồng thuận, phân chia công việc hiệu quả, đưa ra cơ chế giải quyết những mâu thuẫn phát sinh trong nhóm.

## Hướng phát triển.

Do thời gian nghiên cứu có hạn, nên chương trình mới chỉ mô phỏng được các thao tác: ký, xác thực chữ ký, mã hóa và giải mã tập tin mà chưa thiết kế một cách hoàn chỉnh để có thể kết nối trực tiếp vào một số phần mềm : gửi nhận email, phần mềm quản lý văn bản … Hướng phát triển của đề tài là xây dựng chương trình để có thể kết nối trực tiếp vào một số phần mềm gửi nhận email và phần mềm quản lý văn bản. Đồng thời xây dựng một hệ thống chứng thực khóa công khai cho các thành viên, nhằm tránh trường hợp bị người khác giả mạo khóa công khai của người nhận khi thực hiện trao đổi thông tin.

Cuối cùng, với những kết quả đạt được của đề tài nghiên cứu, tuy còn có những hạn chế, nhưng đã giúp chúng em có được khả năng nghiên cứu cơ bản về bảo mật và xác thực thông tin. Từ đó có thể xây dựng các ứng dụng về bảo mật và xác thực thông tin ở những cấp độ an toàn khác nhau

## TÀI LIỆU THAM KHẢO

1. <https://vi.wikipedia.org/wiki/RSA_(m%C3%A3_h%C3%B3a)>
2. <https://en.wikipedia.org/wiki/RSA>
3. <https://vi.wikipedia.org/wiki/Ch%E1%BB%AF_k%C3%BD_s%E1%BB%91>
4. <https://en.wikipedia.org/wiki/Digital_signature>
5. <https://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html>
6. A. MENEZES, P. VAN OORSCHOT, AND S. VANSTONE (1997), Handbook of Applied Cryptography, CRC Press.
7. Rolf Oppliger (2005), Contemporary Cryptography, Artech House.
8. DAN BONEH (1999), Twenty Year of Attacks on RSA, Stanford University.
9. WHITFIELD DIFFIE AND MARTIN E.HELLMAN (1976), New Direction in Cryptography, Invited Paper.