TRƯỜNG ĐẠI HỌC TRẦN ĐẠI NGHĨA

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỌC

MÔN HỌC: AN TOÀN VÀ BẢO MẬT HỆ THỐNG THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG CHƯƠNG TRÌNH CHỮ KÝ SỐ ĐỂ XÁC THỰC NỘI DUNG CỦA MỘT VĂN BẢN BẰNG MÃ HÓA BẤT ĐỐI XỨNG

TP. HỒ CHÍ MINH, THÁNG 11 NĂM 2020

TRƯỜNG ĐẠI HỌC TRẦN ĐẠI NGHĨA

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỌC

MÔN HỌC: AN TOÀN VÀ BẢO MẬT HỆ THỐNG THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG CHƯƠNG TRÌNH CHỮ KÝ SỐ ĐỂ XÁC THỰC NỘI DUNG CỦA MỘT VĂN BẢN BẰNG MÃ HÓA BẤT ĐỐI XỨNG

Nhóm báo cáo:

Nguyễn Tiểu Phụng

Huỳnh Đức Anh Tuấn

Giảng viên hướng dẫn:

Th.s Đặng Thế Hùng

TP. HỒ CHÍ MINH, THÁNG 11 NĂM 2020

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến thầy cô giảng viên trong khoa Công nghệ thông tin trường Đại học Trần Đại Nghĩa. Và đặc biệt là thầy Thạc sĩ Đặng Thế Hùng – giảng viên học phần "An toàn bảo mật hệ thống thông tin" đã tận tình hướng dẫn, truyền đạt kiến thức và kỹ năng cần thiết để chúng em có thể hoàn thành đồ án môn học này.

Tuy nhiên, trong quá trình tìm hiểu và nghiên cứu đề tài, do kiến thức chuyên ngành và thời gian còn hạn chế chúng em vẫn còn nhiều thiếu sót trong quá trình tìm hiểu, thực hiện, đánh giá và trình bày về đề tài. Rất mong được sự quan tâm, góp ý của các thầy cô và giảng viên bộ môn để đồ án môn học của chúng em được hoàn chỉnh hơn.

Xin chân thành cảm ơn!

MỞ ĐẦU

1. Lý do chọn đề tài

- Mật mã học là một trong những vấn đề quan trọng trong lĩnh vực bảo mật và an toàn thông tin. Với sự bùng nổ mạnh mẽ của internet hiện nay, mạng máy tính đang ngày càng đóng vai trò thiết yếu trong mọi lĩnh vực hoạt động của toàn xã hội, và khi nó trở thành phương tiện điều hành các hệ thống thì nhu cầu bảo mật thông tin được đặt lên hàng đầu.
- Việc sử dụng chữ ký số là một giải pháp hữu hiệu, ngày càng được ứng dụng nhiều trong thực tế, không chỉ giới hạn trong lĩnh vực công nghệ thông tin, mật mã học mà còn được áp dụng trong nhiều lĩnh vực khác như tài chính ngân hàng, viễn thông,...
- Mật mã học khóa công khai tạo ra chữ ký số và úng dụng vào các tài liệu. Hệ mã hóa RSA – hệ mã hóa điểm hình của mật mã công khai cùng với hàm băm mật mã SHA chính là những công cụ chính tạo ra chữ ký số.

2. Cấu trúc đồ án

- Chương 1: Tổng quan
- Chương 2: Hệ mã RSA – Hàm băm SHA và Chữ ký số
- Chương 3: Xây dựng ứng dụng

MỤC LỤC

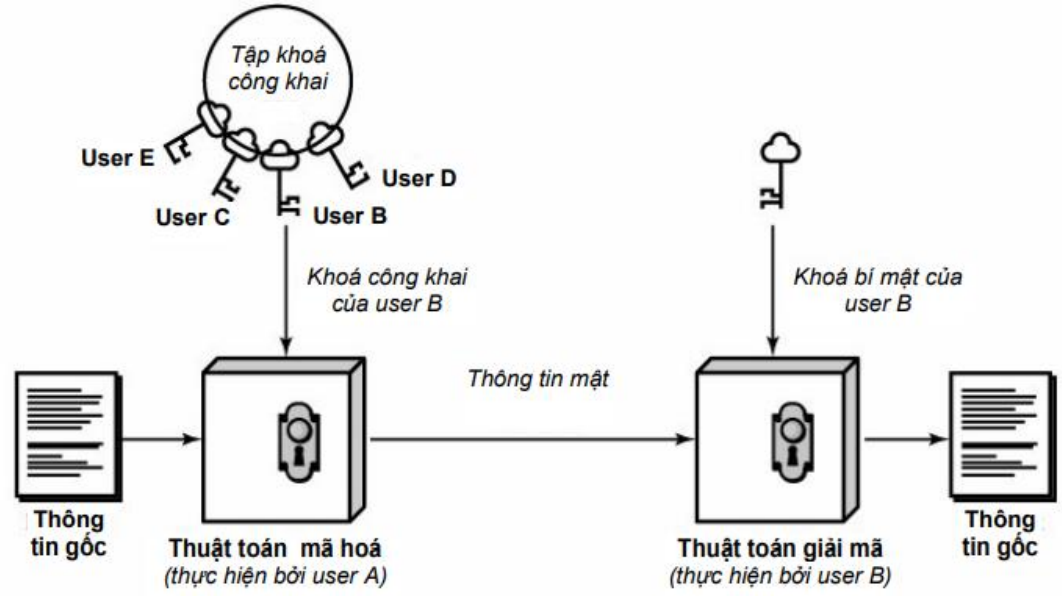
LỜI CẢM ƠN **3**
MỞ ĐẦU **4**
CHƯƠNG 1 HỆ MÃ RSA – HÀM BĂM SHA VÀ CHỮ KÝ SỐ **1**
1.1. Hệ mã RSA 1
1.1.1. Cấu trúc hệ thống mật mã bất đối xứng 1
1.1.2. Thuật toán mật mã RSA 2
1.2. Thuật toán băm SHA 4
1.2.1. Hàm băm 4
1.2.2. Thuật toán SHA 5
1.3. Chữ ký số 6
1.3.1. Nguyên lý hoạt động của chữ ký số 6
1.3.2. Chuẩn chữ ký DSS 8
CHƯƠNG 2 XÂY DỰNG ỨNG DỤNG **9**
2.1. Xác định mô hình 9
2.2. Cài đặt 10
2.2.1. Modul tạo khóa. 10
2.2.2. Modul tạo chữ ký cho file văn bản. 11
2.2.3. Modul kiểm tra xác thực. 12
TÀI LIỆU THAM KHẢO **16**

CHƯƠNG 1 HỆ MÃ RSA – HÀM BĂM SHA VÀ CHỮ KÝ SỐ

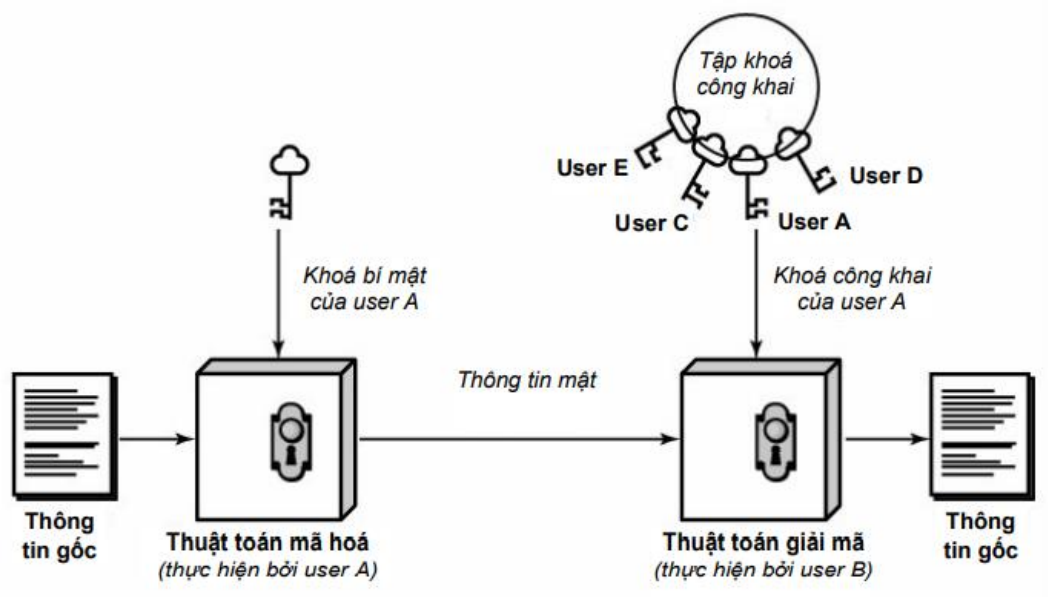
1.1.Hệ mã RSA

1.1.1. Cấu trúc hệ thống mật mã bất đối xứng

- Đặc trưng của kỹ thuật mật mã bất đối xứng là dùng 2 khóa riêng biệt cho hai quá trình mã hóa và giải mã, trong đó có một khóa được phổ biến công khai (public key hay PU) và khóa còn lại được giữ bí mật (private key hay PR). Cả hai khoá đều có thể được dùng để mã hoá hoặc giải mã. Việc chọn khoá công khai hay khoá bí mật cho quá trình mã hoá sẽ tạo ra hai ứng dụng khác nhau của kỹ thuật mật mã bất đối xứng:
  + Nếu dùng khoá công khai để mã hoá và khoá bí mật để giải mã, ta có ứng dụng bảo mật trên thông tin (confidentiality).
  + Nếu dùng khoá bí mật để mã hoá và khoá công khai để giải mã, ta có ứng dụng xác thực nội dung và nguồn gốc thông tin (authentication).



Hình a- ứng dụng trong bảo mật thông tin



Hình b- ứng dụng trong xác thực thông tin

Hình 2.1 Cấu trúc hệ thông mật mã bất đối xứng

- Cấu trúc một hệ thống mật mã bất đối xứng được trình bày trong hình 2.22.
- Mật mã hóa bất đối xứng được sử dụng trong các ứng dụng: che giấu thông tin, tạo chữ ký số (digital signature) và trao đổi khóa trong các thuật tóan mật mã đối xứng (key exchange).

1.1.2. Thuật toán mật mã RSA

- Cũng như các thuật toán mật mã bất đối xứng khác, nguyên lý của RSA dựa chủ yếu trên lý thuyết số chứ không dựa trên các thao tác xử lý bit.
- RSA là một thuật toán mật mã khối, kích thước khối thông thường là 1024 hoặc 2048 bit.
- Thông tin gốc của RSA được xử lý như các số nguyên. Ví dụ, khi chọn kích thước khối của thuật toán là 1024 bit thì số nguyên này có giá trị từ 0 đến $2^{1024} – 1$, tương đương với số thập phân có 309 chữ số. Chú ý rằng đây là những số nguyên cực lớn, không thể xử lý được bằng cách sử dụng các cấu trúc dữ liệu có sẵn của các ngôn ngữ lập trình phổ biến.

- Tóm lại, thuật toán mật mã RSA được thực hiện gồm 3 quá trình tách rời: tạo khoá, mã hoá và giải mã được tóm tắt như sau:

1-Tạo khoá:

- ***Chọn p, q*** *(p và q là số nguyên tố, $p \neq q$)*
- ***Tính N = p.q***
- ***Tính $\phi(N) = (p - 1)(q - 1)$***
- ***Chọn e sao ước số chung lớn nhất của e và $\phi(N)$ là 1***
- ***Chọn d sao cho $e.d \bmod \phi(N) = 1$***
- ***Cặp khoá RSA được tạo ra là PU = (N, e), PR = (N, d)***

2- Mã hoá:

- ***$C = M^e \bmod N$*** *(M là số nguyên nhỏ hơn N)*

3- Giải mã:

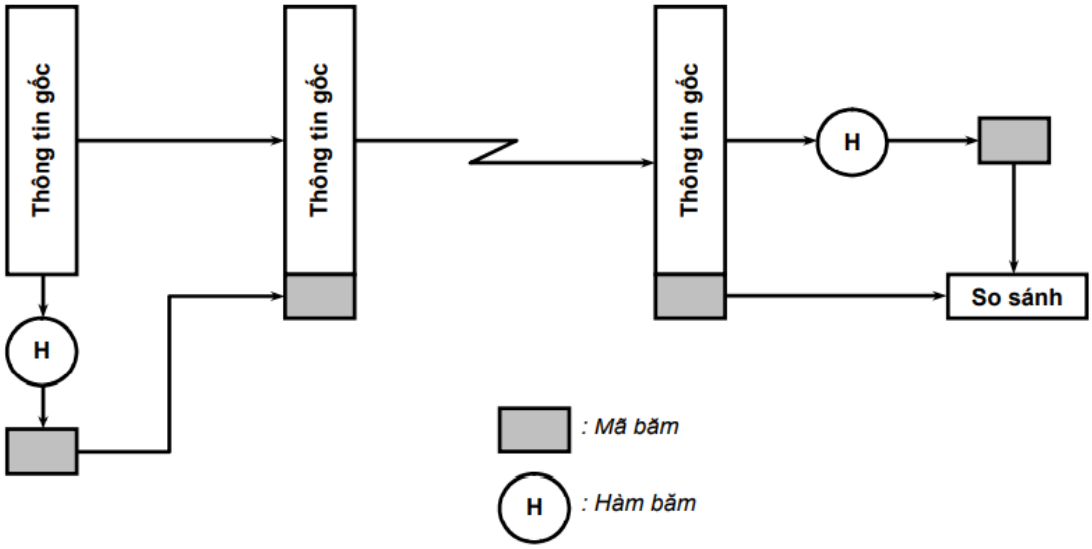
- ***$M = C^d \bmod N$***

- Trong thực tế, để đạt được độ an tòan cao, cặp khóa phải được chọn trên các số p và q đủ lớn (N nhỏ nhất phải là 1024 bit), do vậy, vấn đề thực thi RSA bao gồm các phép tóan lũy thừa trên các số rất lớn.
- Vấn đề giảm chi phí tính tóan và tăng tốc độ thực hiện thuật tóan RSA là một trong những vấn đề quan trọng cần phải giải quyết. Trên các hệ thống máy tính hiện nay, hiệu suất thực hiện giải thuật RSA là chấp nhận được.
- Độ an toàn của RSA:
- Theo lý thuyết, hệ thống RSA có thể bị tấn công bằng những phương thức sau đây:
  + Brute-force attack: tìm lần lượt khoá riêng PR
  + Mathematical attack: xác định p và q bằng cách phân tích N thành tích của các thừa số nguyên tố rồi từ đó xác định e và d.
  + Timing attack: dựa trên thời gian thực thi của thuật toán giải mã.
  + Chosen ciphertext attack: sử dụng các đoạn thông tin mật (ciphertext) đặc biệt để khôi phục thông tin gốc.

1.2. Thuật toán băm SHA

1.2.1. Hàm băm

- Các hàm băm bảo mật (secure hash functions) hay gọi tắt là hàm băm là một trong những kỹ thuật cơ bản để thực hiện các cơ chế xác thực thông tin (message authentication). Ngoài ra, hàm băm cũng còn được sử dụng trong nhiều thuật toán mật mã, trong chữ ký số (digital signature) và nhiều ứng dụng khác.
- Nguyên tắc của hàm băm là biến đổi khối thông tin gốc có độ dài bất kỳ thành một đoạn thông tin ngắn hơn có độ dài cố định gọi là mã băm (hash code hay message digest). Mã băm được dùng để kiểm tra tính chính xác của thông tin nhận được. Thông thường, mã băm được gởi kèm với thông tin gốc. Ở phía nhận, hàm băm lại được áp dụng đối với thông tin gốc để tìm ra mã băm mới, giá trị này được so sánh với mã băm đi kèm với thông tin gốc. Nếu hai mã băm giống nhau, nghĩa là thông tin gởi đi không bị thay đổi.



Hình 2.2 Một ứng dụng điểm hình của hàm băm

- Hình 2.2 mô tả nguyên lý hoạt động của một giải thuật xác thực thông tin sử dụng hàm băm đơn giản.
- Các yêu cầu của một hàm băm bảo mật H:

+ H có thể được áp dụng cho khối thông tin với chiều dài bất kỳ.
+ Kết quả của hàm H luôn có chiều dài cố định.
+ Việc tính giá trị của H(x) với một giá trị x cho trước phải đơn giản, có thể thực hiện được bằng cả phần cứng hoặc phần mềm.
+ Cho trước một giá trị h, không thể tìm được một giá trị x sao cho H(x) = h, đây được gọi là thuộc tính một chiều của hàm băm (one-way property).
+ Cho trước khối thông tin x, không thể tìm được một khối thông tin y khác x sao cho H(y) = H(x). Thuộc tính này được gọi là weak collision resistance.
+ Không thể tìm được hai khối thông tin x và y khác nhau sao cho H(x) = H(y).

- Thuộc tính này được gọi là strong collision resistance.

1.2.2. Thuật toán SHA

- SHA-1 tạo ra mã băm có chiều dài cố định là 160 bit. Về sau, có nhiều nâng cấp đối với SHA, chủ yếu là tăng chiều dài mã băm, từ đó xuất hiện các phiên bản khác nhau của SHA, bao gồm: SHA-256 (mã băm dài 256 bit), SHA-384 (mã băm dài 385 bit) và SHA-512 (mã băm dài 512 bit).

Thông số	SHA-1	SHA-256	SHA-384	SHA-512
Kích thước mã băm (bit)	160	256	384	512
Kích thước thông tin gốc (bit)	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$<2^{128}$
Kích thước khối (bit)	512	512	1024	1024
Độ dài từ (bit)	32	32	64	64
Số bước thực hiện (bước)	80	64	80	80

Hình 2.3 Các phiên bản SHA

- Phần này chỉ mô tả thuật toán băm SHA-1, các phiên bản khác của SHA cũng được thiết kế theo nguyên lý tương tự.
- SHA-1 chấp nhận các khối thông tin có kích thước tối đa là 264 bit để tạo ra mã băm với độ dài cố định 160 bit. Toàn bộ khối thông tin được xử lý theo từng khối 512 bit, qua 5 công đoạn.

1.3.Chữ ký số

1.3.1. Nguyên lý hoạt động của chữ ký số

- Chữ ký số là một cơ chế xác thực cho phép người tạo ra thông tin (message creator) gắn thêm một đoạn mã đặc biệt vào thông tin có tác dụng như một chữ ký. Chữ ký được tạo ra bằng cách áp dụng một hàm băm lên thông gốc, sau đó mã hóa thông tin gốc dùng khóa riêng của người gởi. Chữ ký số có mục đích đảm bảo tính tòan vẹn về nguồn gốc và nội dung của thông tin.
- Tại sao phải dùng chữ ký số trong khi các cơ chế xác thực thông tin (message authentication) đã thực hiện chức năng xác thực nguồn gốc thông tin? Các cơ chế xác thực thông tin sử dụng các hàm băm một chiều có tác dụng bảo vệ thông tin trao đổi giữa hai thực thể thông tin khỏi sự xâm phạm của một thực thể thứ 3, tuy nhiên nó không có tác dụng ngăn chặn được sự xâm phạm của chính hai thực thể.
- Các yêu cầu đối với chữ ký số:
  + Là một chuỗi bit phát sinh từ khối thông tin cần được xác nhận (thông tin gốc).
  + Chữ ký phải chứa thông tin nhận dạng riêng của người ký để tránh giả mạo và tránh phủ nhận.
  + Quy trình tạo ra chữ ký cũng như xác minh chữ ký phải đơn giản, nhanh chóng
  + Chữ ký thông thể bị giả mạo bằng bất cứ cách nào.
  + Có thể sao chép một bản sao của chữ ký dành cho mục đích lưu trữ.
- **Phân loại chữ ký số**: Có nhiều thuật toán phát sinh chữ ký số khác nhau. Có thể phân loại các thuật toán này theo các cách như sau:
  + Chữ ký cố định và chữ ký ngẫu nhiên: thuật toán tạo chữ ký cố định (deterministic) tạo ra một chữ ký duy nhất ứng với một khối thông tin gốc xác định, nghĩa là nếu thực hiện nhiều lần thuật toán tạo chữ ký trên một bản tin thì vẫn cho ra một kết quả duy nhất. Ngược lại, chữ ký ngẫu nhiên (probabilistic) tạo ra những chữ ký khác nhau đối với cùng một bản tin.

+ Chữ ký phục hồi được và chữ ký không phục hồi được: cơ chế tạo chữ ký phục hồi được (reversible signature) cho phép người nhận phục hồi lại thông tin gốc từ chữ ký, điều này cũng có nghĩa là chữ ký phải có chứa thông tin gốc trong nó dưới một dạng mã hoá nào đó, và kết quả là chữ ký số sẽ có kích thước lớn hơn thông tin gốc. Khi đó, người gởi chỉ cần gởi đi chữ ký là đủ. **Các phương pháp thực hiện chữ ký số**: Có hai phương pháp thực hiện chữ ký số là ký trực tiếp (direct signature) và ký thông qua trọng tài (arbitrated signature).

+ *Ký trực tiếp (direct signature):* Ở phương pháp này, giả thiết rằng phía nhận biết được khóa công khai của phía gởi. Do đó, chữ ký có thể được tạo ra bằng cách mã hóa tòan bộ bản tin bằng khóa riêng của người tạo ra thông tin, hoặc là chỉ mã hóa phần mã băm (kết quả tạo ta từ hàm băm đối với thông tin gốc) dùng khóa riêng của người tạo thông tin.

    - Để đạt được tính bảo mật của thông tin thì thông tin gốc cùng với chữ ký vừa được tạo ra sẽ được mã hóa sử dụng khóa công khai của thực thể nhận chữ ký (trong trường hợp dùng mật mã bất đối xứng) hoặc dùng khóa bí mật (trong trường hợp dùng mật mã đối xứng).

    - Một nhược điểm rất dễ thấy của phương thức ký trực tiếp đó là độ an tòan của chữ ký phụ thuộc cao độ vào khóa riêng của người tạo ra chữ ký. Do vậy, nếu khóa riêng này bị mất hoặc bị tiết lộ thì ý nghĩa của chữ ký số sẽ không còn.

+ Ký thông qua trọng tài (arbitrated signature): đây là một giải pháp được xây dựng để khắc phục nhược điểm của chữ ký trực tiếp. Khi thực thể A muốn gởi một bản tin cho thực thể B, quá trình tạo ra một chữ ký được thực hiện bình thường như đối với chữ ký trực tiếp.

+ Tuy nhiên, trước khi bản tin này được gởi đến B, nó phải được gởi đến một thực thể thứ 3 gọi là trọng tài (arbiter). Trọng tài thực hiện việc kiểm tra, xác nhận tính chính xác của thông tin và chữ ký, sau đó ghi lại ngày giờ rồi mới

gởi cho thực thể B, kèm theo thông tin xác nhận của trọng tài. Sự xuất hiện của trọng tài trong quy trình đảm bảo được thực thể A sẽ không phủ nhận được thông tin mình đã gởi.

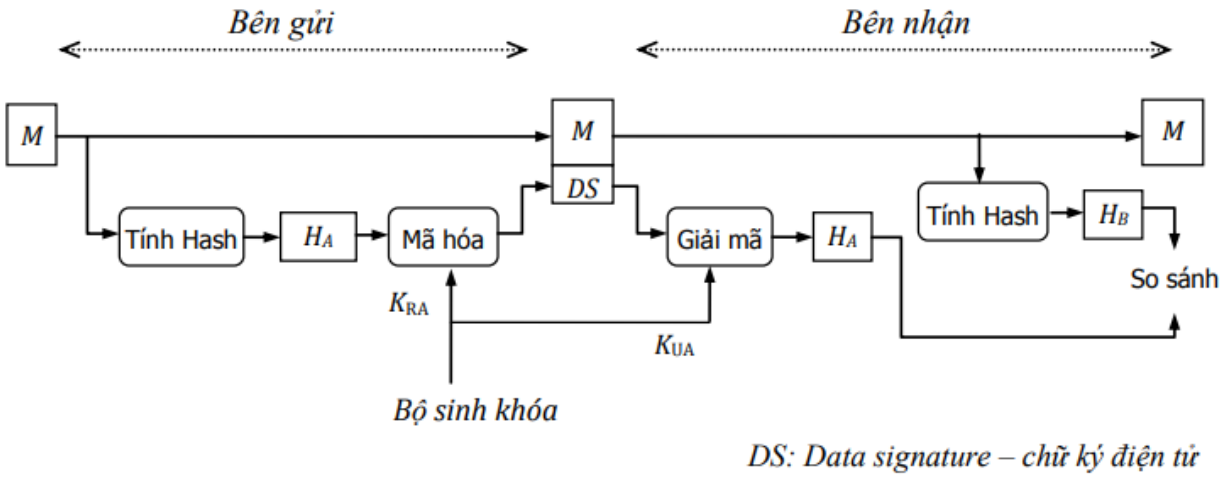
1.3.2. Chuẩn chữ ký DSS

- Trong thuật toán xác thực thông tin dùng mật mã RSA, thông tin gốc được đưa vào hàm băm SHA để tạo ra mã băm (tức message digest) có kích thước cố định. Mã băm này sau đó được mã hóa (bằng thuật toán RSA) dùng khóa riêng của thực thể tạo thông tin (phía gởi).
- Kết quả của phép mã hóa được gắn vào thông tin gốc và gởi đi. Phía thu nhận được thông tin, tách phần mã băm ra khỏi thông tin gốc và giải mã nó bằng khóa công khai của phía gởi.
- Chú ý rằng khóa công khai là thông tin được công bố rộng rãi cho bất kỳ thực thể nào có quan tâm. Đồng thời, thông tin gốc cũng được đưa vào hàm băm để tính mã băm, sau đó đem so sánh với mã băm vừa nhận được. Nếu hai mã này giống nhau thì thông tin vừa nhận được chấp nhận như là thông tin hợp lệ.
- Hoạt động của DSS cũng bao gồm việc đưa thông tin gốc vào hàm băm để tạo ra mã băm có kích thước cố định. Tuy nhiên, mã băm này sẽ không được mã hóa trực tiếp bằng một giải thuật mã hóa mà được sử dụng làm ngõ vào của một hàm tạo chữ ký S (Signature function). Các thông tin đưa vào hàm tạo chữ ký bao gồm:
  + Mã băm của thông tin gốc
  + Một số ngẫu nhiên k
  + Khóa riêng của người ký (PR_a)
  + Khóa công khai của nhóm các thực thể liên quan đến giao dịch chữ ký (PU_G).

CHƯƠNG 2 XÂY DỰNG ỨNG DỤNG

2.1.Xác định mô hình

Mô hình chữ ký số RSA trong các hệ thống quản lý: Quá trình gửi và nhận các tệp văn bản phục vụ quản lý dựa vào thuật toán SHA-256 và thuật toán RSA.



Hình 3.1 Mô hình chữ ký số

- Quán trình ký và gửi các tệp văn bản.
  + Từ file cần gửi ban đầu, chương trình sẽ sử dụng hàm băm SHA-256 để mã hóa chuỗi ký tự dài 256 bit. Chương trình sử dụng thuật toán RSA để mã hóa giá trị băm thu được với khóa bí mật của người gửi để nhận được một giá trị gọi là chữ ký điện tử. Kết hợp file ban đầu với chữ ký điễn tử thành một thông điệp đã ký và gửi đi cho người nhận.
- Quá trình nhận tệp văn bản.
  + Sau khi người nhận nhận được văn bản. Hệ thống sẽ tách thông điệp đã ký ra thành file văn bản và chữ ký điện tử. Đến giai đoạn này có 2 quá trình kiểm tra:
  + Kiểm tra file văn bản có đúng người gửi hay không. Sử dụng thuật toán RSA để giải mã chữ ký điện tử bằng khóa công khai của người gửi. Nếu giải mã không được file thì file nhận được là không đúng người gửi. Nếu giải mã

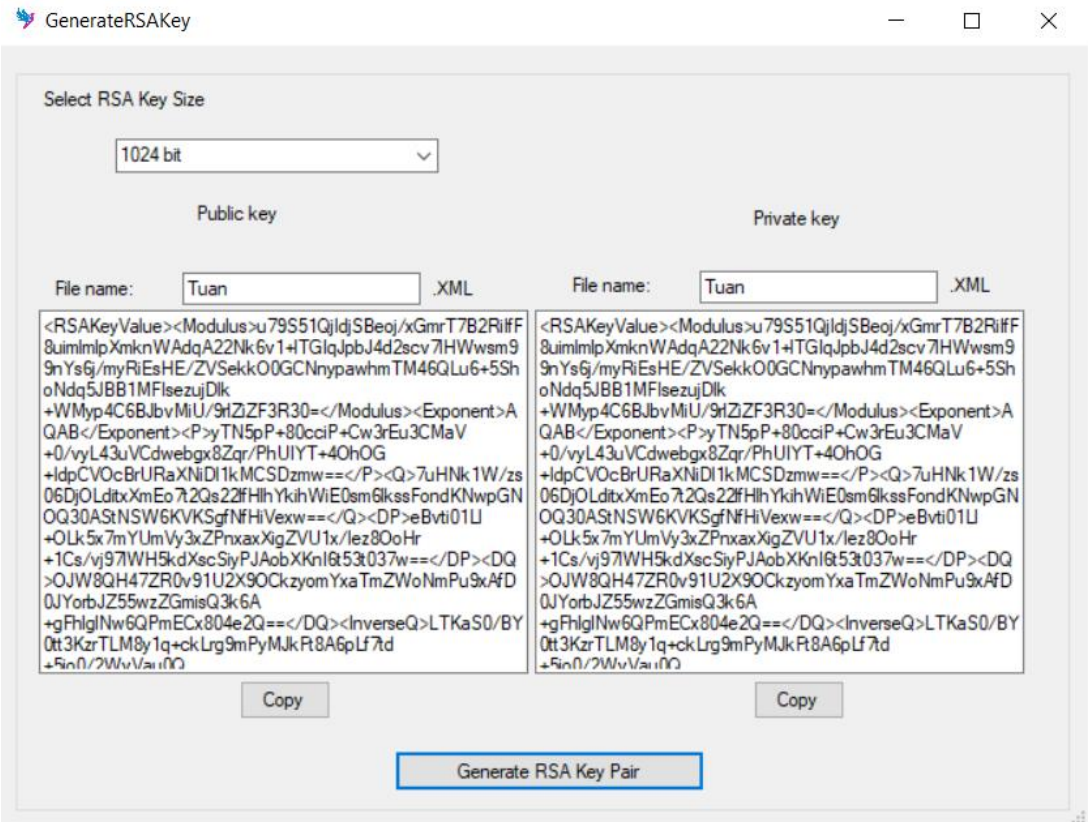
thành công thì file nhận được là đúng người gửi và ta nhận được giá trị băm 1.

+ Kiểm tra file văn bản có bị thay đổi hay không: Từ file văn bản ban đầu ta sử dụng hàm băm SHA-256 mã hóa thành giá trị băm 2. Kiểm tra giá trị băm 1 và giá trị băm 2 có giống nhau hay không? Nếu giống nhau thì file nhận được là toàn vẹn không bị thay đổi, nếu ngược lại thì file văn bản đã bị thay đổi.

2.2. Cài đặt

2.2.1. Modul tạo khóa.

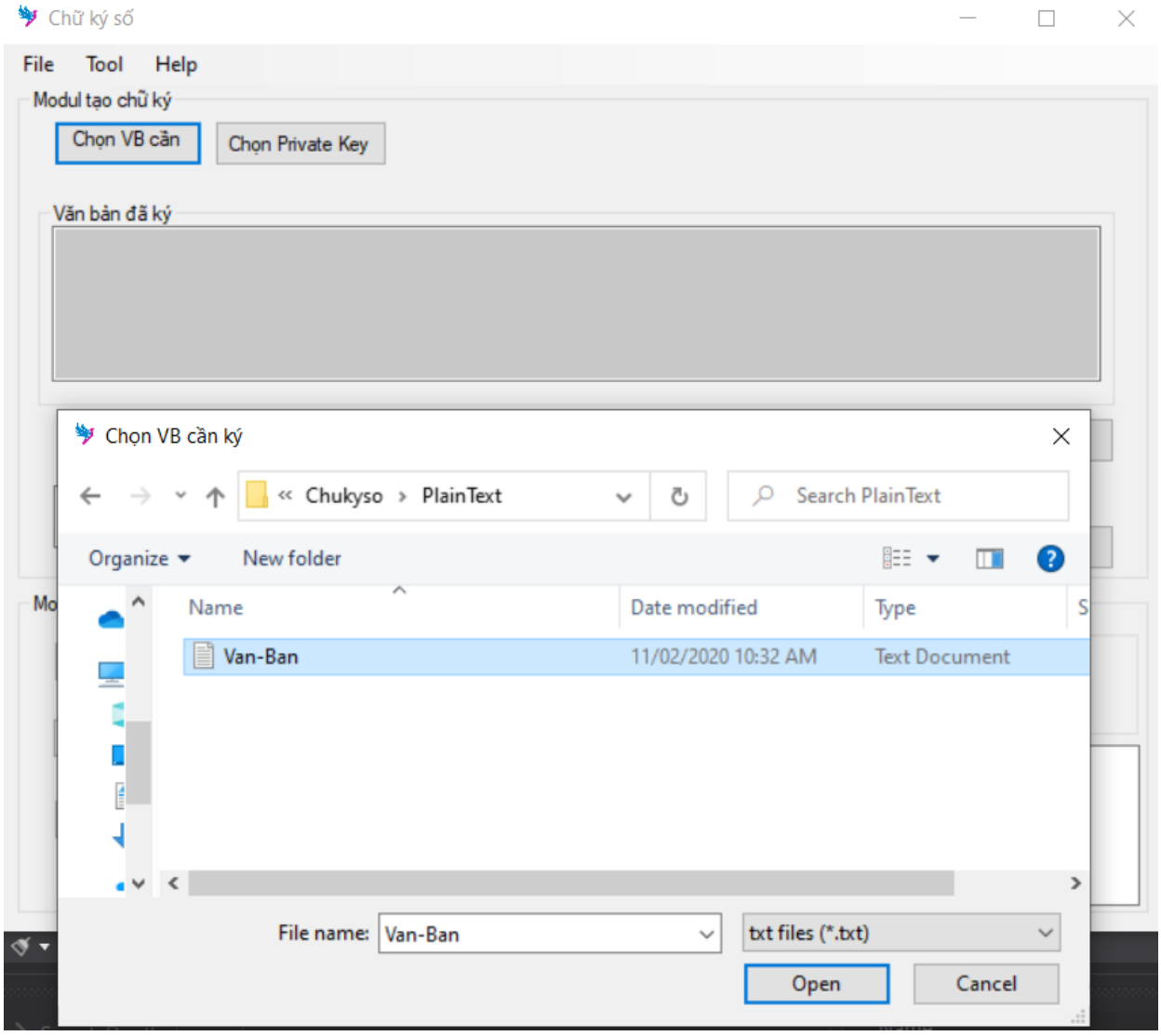
- Nhiệm vụ của modul này chính là tạo cặp khóa công khai và khóa bí mật cho người dùng. Mặc định chương trình tạo cặp khóa có độ dài 1024 bit.
- Sử dụng lớp ***RSACryptoServiceProvider*** để khởi tạo cặp khóa và chuyển về định dạng XML.



Hình 3.2 Giao diện modul tạo cặp khóa

2.2.2. Modul tạo chữ ký cho file văn bản.

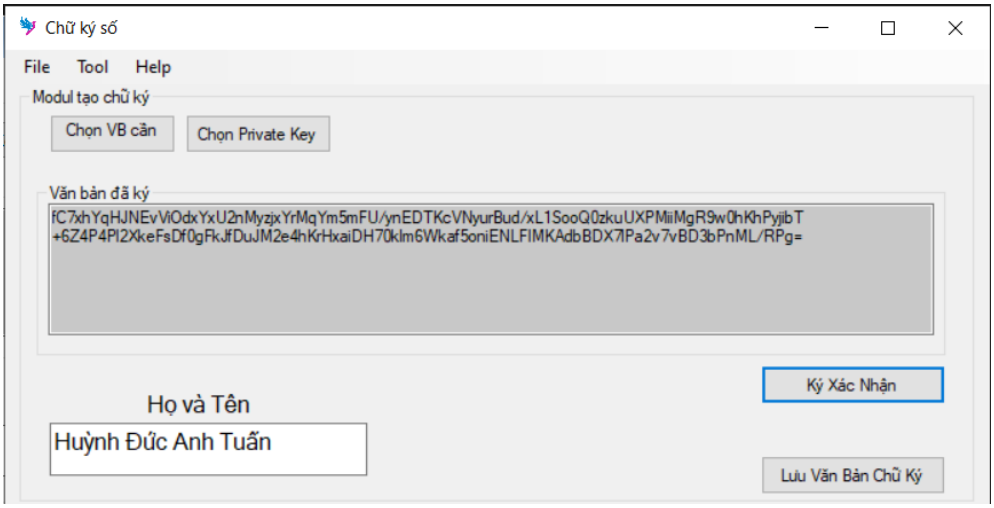
- Nhiệm vụ cơ bản của modul này là tạo ra file chữ ký (*sig) có tên trùng với tên file được chọn để ký. Người dùng muốn sử dụng chứng năng này phải thông qua việc tạo khóa.
- Đầu tiên ta chọn file Văn bản cần ký (Plain text)



Hình 3.2 Giao diện chọn file plain text

- Kế tiếp ta chọn file private key của người ký văn bản.
- Sau đó điền họ và tên của người ký vào ô trống.

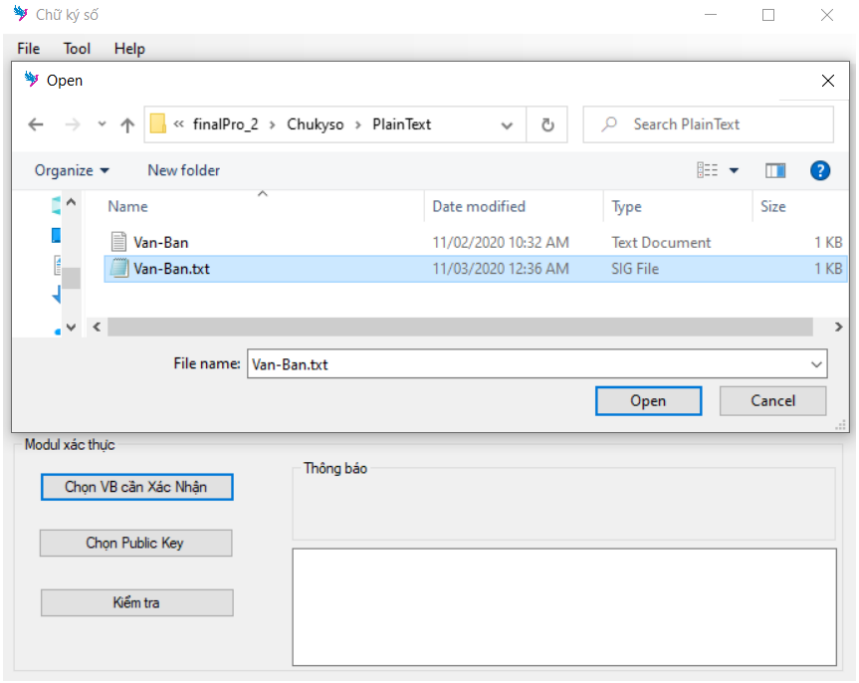
- Kế tiếp chọn ký xác nhận, và lưu văn bản đã được ký (chữ ký số) lại thành một thông điệp.



Hình 3.3 Giao diện modul tạo chữ ký số

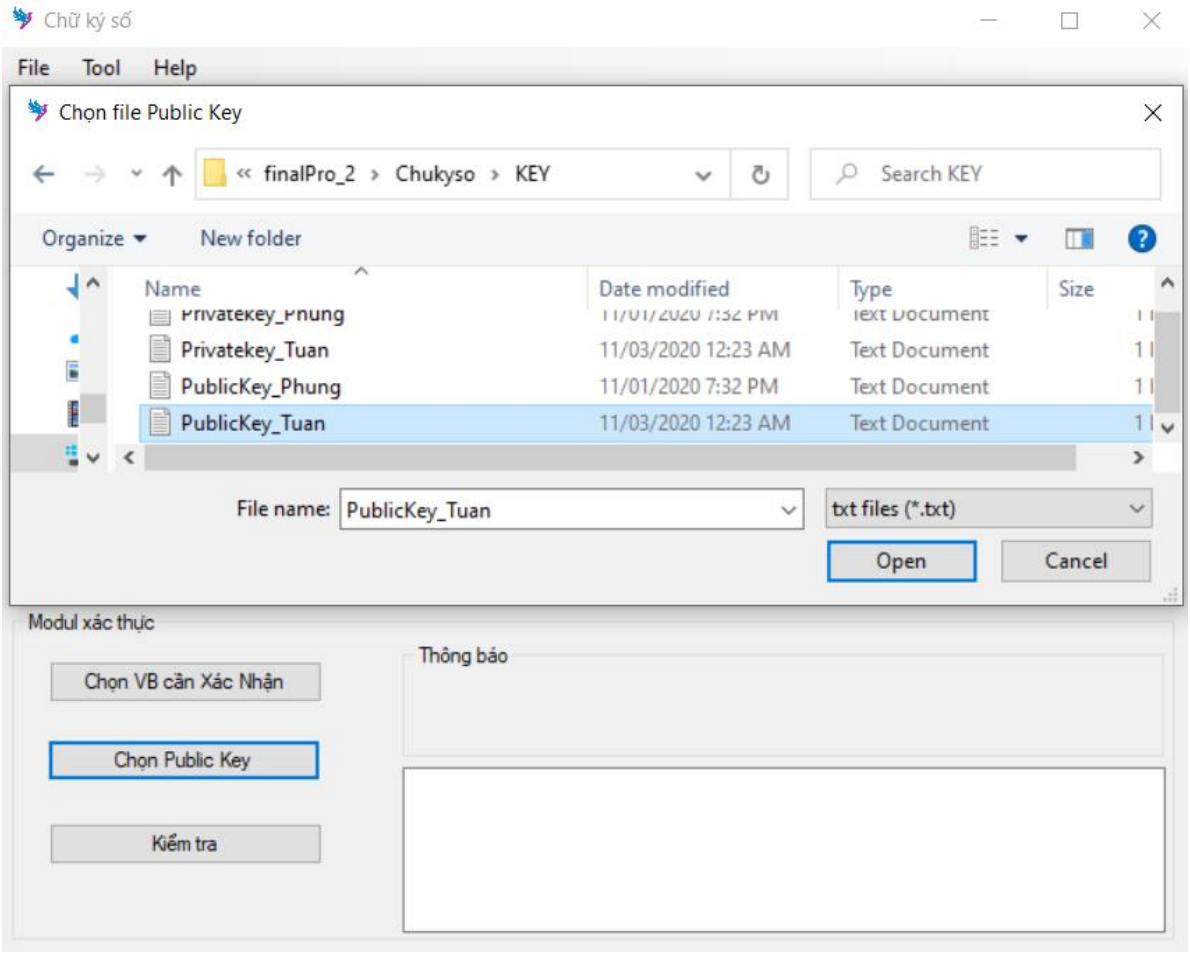
2.2.3. Modul kiểm tra xác thực.

- Đầu vào của modul này là file cần xác thực + với chữ ký của nó (thông điệp). Nhiệm vụ của modul này chính là kiểm tra tính đúng đắn của chữ ký số.



Hình 3.4 Giao diện chọn file thông điệp cần xác nhận

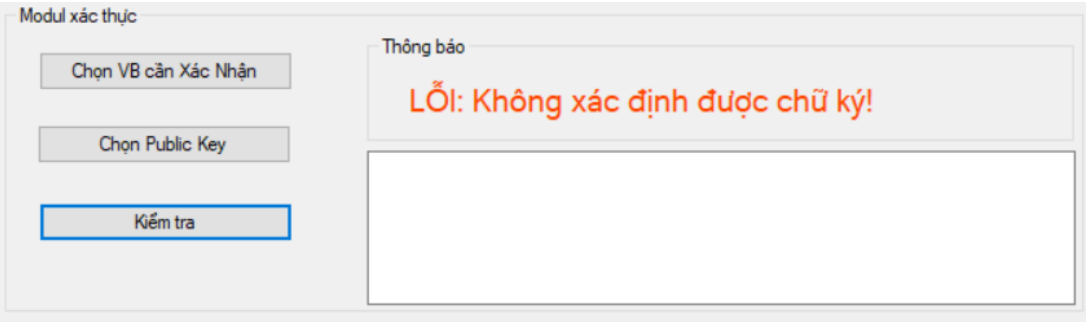
- Đầu tiên ta chọn file thông điệp cần được xác thực hình 3.4.
- Kế tiếp ta chọn public key của người gửi thông điệp, và bấm kiểm tra



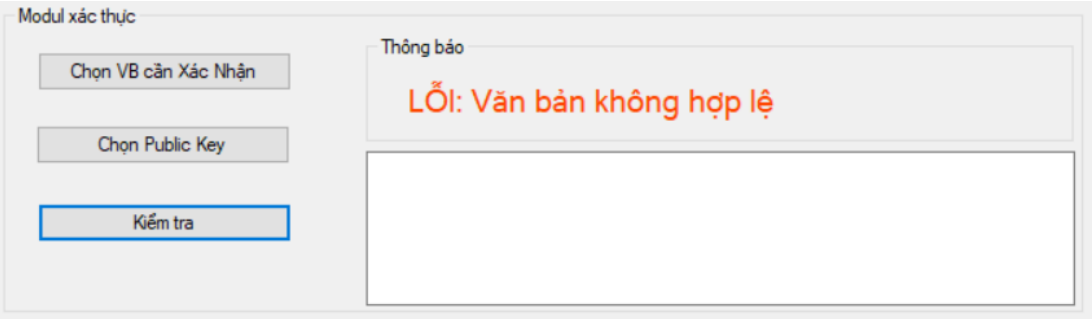
Hình 3.5 Chọn public key của người gửi thông điệp

Hình 3.6 Kết quả xác thực chữ ký và định danh người gửi

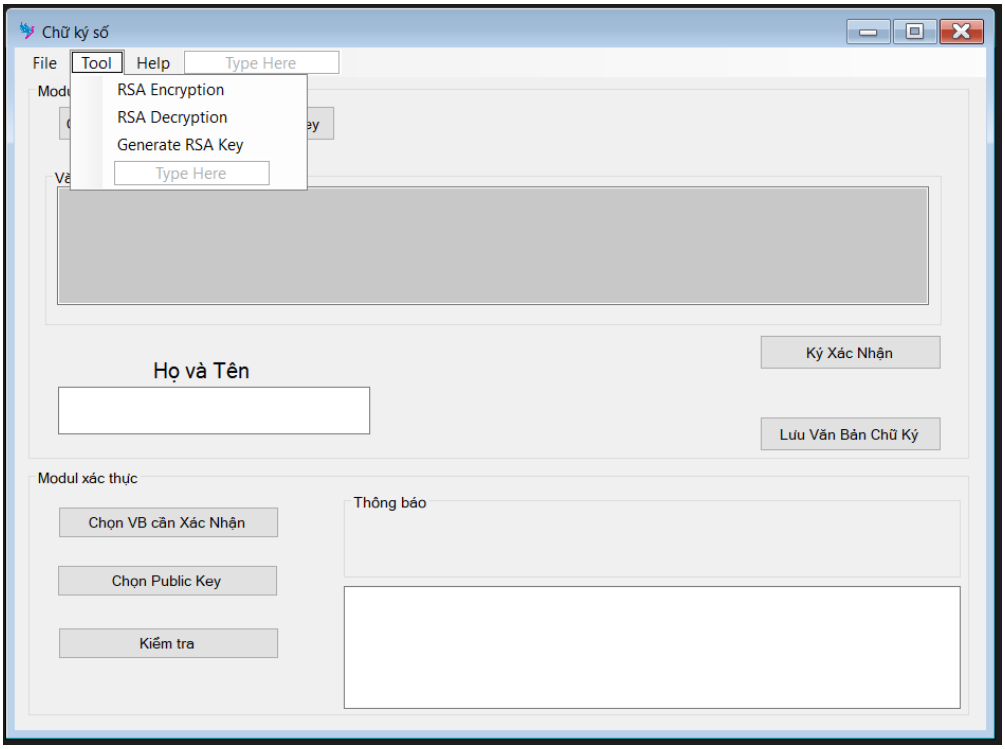
- Trường hợp nếu không giải mã được public key thì sẽ thông báo lỗi.



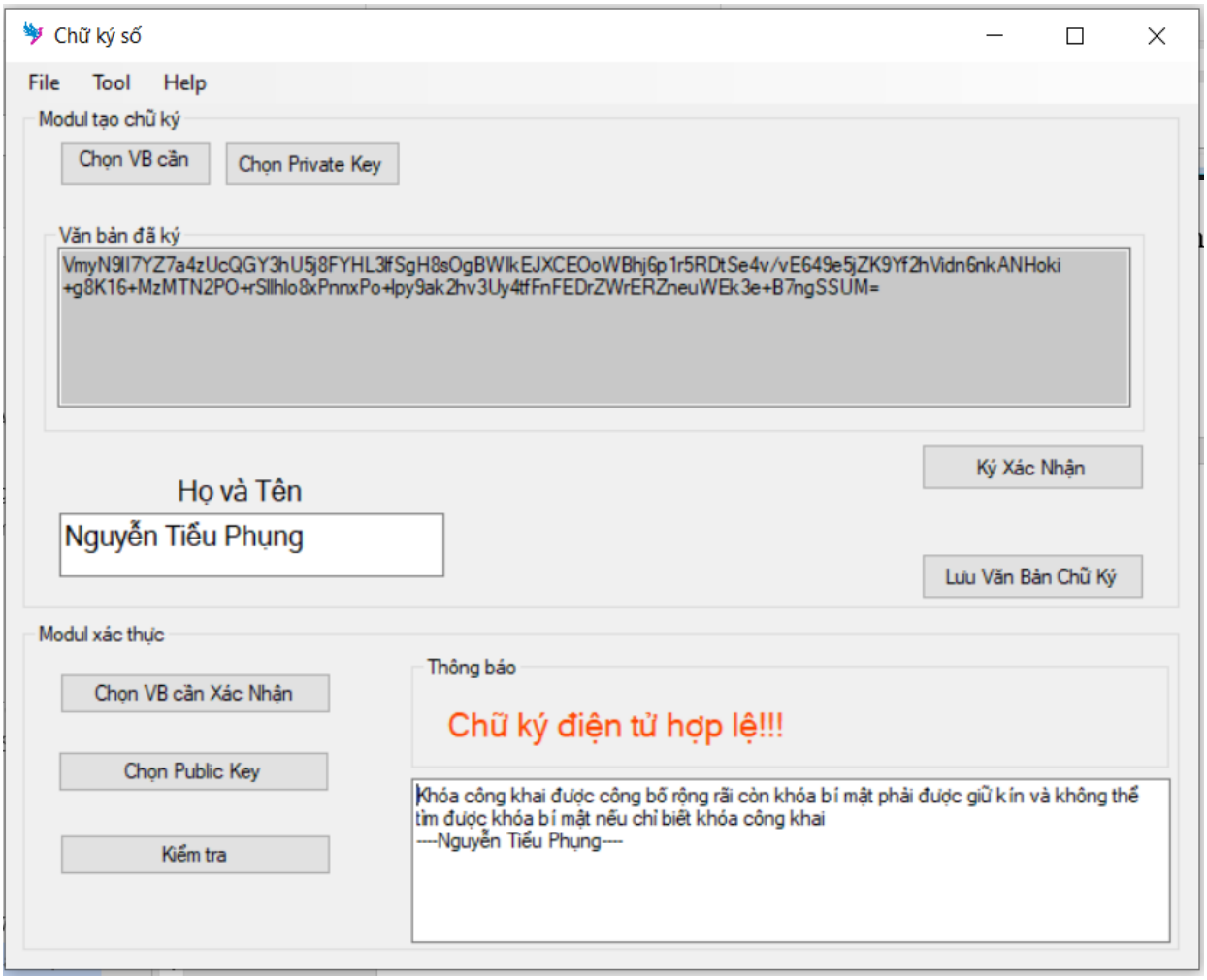
Hình 3.7 Thông báo lỗi khi không thể giải mã được public key



Hình 3.8 Thông báo lỗi khi không thể chứng thực được văn bản



Hình 3.9 Giao diện chính của ứng dụng



Hình 3.10 Giao diện của ứng dụng

TÀI LIỆU THAM KHẢO

1. An toàn và Bảo mật thông tin – Trần Minh Văn
2. Bảo mật hệ thống thông tin – Lê Phúc
3. Cryptography and Network Security Principles and Practices - William Stallings