TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

------\*\*\*-------



**BÁO CÁO THỰC NGHIỆM**

Học phần: An toàn & Bảo mật thông tin

**Chủ đề**: **Tìm hiểu về hệ chữ ký DSA và viết ứng dụng minh họa (Sử dụng ngôn ngữ C++, C#)**

.

Giáo viên hướng dẫn : TS. Phạm Văn Hiệp

Nhóm sinh viên thực hiện :

1. Nguyễn Lương Chiến Mã SV: 2021601975

2. Dương Văn Chung Mã SV: 2022600137

3. Nguyễn Xuân Sơn Mã SV:

4. Nguyễn Minh Quang Mã SV: 2021603794

5. Lê Trung Hiếu Mã SV:

Mã Lớp học phần:  20241IT6001002

Nhóm: 13

Hà Nội - Năm 2024

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **TỪ VIẾT TẮT** | **NỘI DUNG** |
| 1 | ECC | Elliptic Curve Cryptography |
| 2 | DSA | Digital Signature Algorithm |
| 3 | P | Plaintext |
| 4 | C | Ciphertext |
| 5 | K | Key |
| 6 | E | Encryption |
| 7 | D | Decryption |

# LỜI CẢM ƠN

Để hoàn thành bài tập lớn kết thúc môn học, chúng em xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn TS. Phạm Văn Hiệp, Khoa Công nghệ Thông tin, Trường Đại học Công nghiệp Hà Nội. Nhờ sự tận tình chỉ dẫn, những ý kiến góp ý sâu sắc và những kiến thức quý báu thầy đã chia sẻ, chúng em đã có cơ hội hoàn thành tiểu luận một cách tốt nhất.

Trong suốt quá trình nghiên cứu đề tài và tham gia khóa học, chúng em không chỉ tích lũy được nhiều kiến thức chuyên môn và kinh nghiệm thực tiễn, mà còn nhận ra những khía cạnh cần tiếp tục rèn luyện và hoàn thiện bản thân.

Đây thực sự là cơ hội quý giá để chúng em chuẩn bị sẵn sàng cho những chặng đường dài phía trước. Tuy nhiên, do hạn chế về kiến thức và kinh nghiệm thực tế, bài nghiên cứu khó tránh khỏi những thiếu sót.

Vì vậy, chúng em rất mong nhận được sự đóng góp ý kiến từ thầy giáo để có thể không ngừng cải thiện và hoàn thiện đề tài này. Một lần nữa, chúng em xin trân trọng cảm ơn TS. Phạm Văn Hiệp vì sự hướng dẫn tận tâm và đồng hành cùng chúng em trong suốt hành trình thực hiện bài tập này.

Chúng em xin chân thành cảm ơn!

Nhóm sinh viên thực hiện

Nhóm 13

# LỜI NÓI ĐẦU

Trước đây khi công nghệ máy tính chưa phát triển, khi nói đến vấn đề an toàn bảo mật thông tin, chúng ta thường hay nghĩ đến các biện pháp nhằm đảm bảo cho thông tin được trao đổi hay cất giữ một cách an toàn và bí mật, chẳng hạn là các biện pháp như: Đóng dấu và ký niêm phong một bức thư để biết rằng lá thư có được chuyển nguyên vẹn đến người nhận hay không, dùng mật mã mã hóa thông điệp để chỉ có người gửi và người nhận hiểu được thông điệp, lưu giữ tài liệu trong các két sắt có khóa tại nơi được bảo vệ nghiêm ngặt.

Ngày nay với sự phát triển của khoa học công nghệ, đặt biệt là sự phát triển của Internet, việc sử dụng máy tính và điện thoại cá nhân càng trở lên rộng rãi, dẫn đến càng nhiều thông tin được lưu trữ trên máy tính và gửi đi trên mạng Internet. Do đó nhu cầu về an toàn và bảo mật thông tin trên máy tính càng nhiều và việc sử dụng mật mã mã hóa càng được phổ biến. Trong báo cáo này, nhóm em tìm hiểu về hệ chữ ký DSA và viết ứng dụng minh họa. Báo cáo này gồm 3 chương:

Chương 1 : Tổng quan.

Chương 2 : Kết quả nghiên cứu.

Chương 3 : Kết luận và bài học kinh nghiệm.

Trong quá trình thực hiện đề tài nhóm chúng em xin gửi lời cám ơn tới thầy Phạm Văn Hiệp giảng viên hướng dẫn nhóm chúng em thực hiện đề tài này. Trong quá trình nghiên cứu và thực hiện đề tài được sự chỉ bảo tận tình của các thầy cô, nhóm chúng em đã cố gắn hết sức để hoàn thiện đề tài. Tuy nhiên chúng em rất mong nhận được sự góp ý của thầy cô và các bạn.

# CHƯƠNG 1: TỔNG QUAN

## Tổng quan về An toàn bảo mật thông tin

An toàn thông tin là bảo vệ các đặc tính riêng tư (confidentiality), toàn vẹn (integrity) và khả dụng (availability) của thông tin.

* C: (Confidentiality) bảo vệ tính riêng tư của dữ liệu thông qua các cơ chế chứng thực và mã hóa, ngăn ngừa những người không hợp lệ sẽ không được đọc những thông tin. Giống như các bì thư khi phát lương thưởng được dán chữ Confidential, chúng ta có thể hình dung trong môi trường công nghệ thông tin là một người chưa đăng nhập vào Domain sẽ không được truy cập những dữ liệu chỉ chia sẻ cho các Domain User.
* I: (Integrity) bảo vệ tính toàn vẹn của dữ liệu thông qua các thuật toán RSA, SHA, MD5 ... ngăn ngừa attacker thay đổi các thông tin nhạy cảm trong quá trình truyền.
* A: (Available) bảo đảm dữ liệu luôn ở trong trạng thái sẵn sàng đáp ứng nhu cầu của người dùng.
* Non-Repudiation: Tính không thể chối bỏ, nghĩa là dữ liệu người nào gửi đi thì họ phải có trách nhiệm với các thông tin của mình thông qua các xác nhận nguồn gốc như chữ ký điện tử.

Để đảm bảo việc truyền tin an toàn và kiểm tra tính toàn vẹn của thông tin, người ta thường mã hóa thông tin trước khi truyền đi bằng các một số các hệ mật như DSA, DES, Triple, DES (3DES), RC4, AES, RSA, Rabin, Diffie-Hellman,

Elgamal,...

Qua dự án lần này nhóm sinh viên chúng em đã đi sâu tìm hiểu về xây dựng và phát triển mã hóa giải mã DSA , nắm được những kiến thức cơ bản

Trong phương pháp mã hóa và giải mã cũng như là các thuật toán cần thiết bằng các ngôn ngữ như: C#, C++..

## 1.2 Các kiến thức cơ sở

### 1.2.1 Kiến thức cơ sở về Toán học

* **Số học modulo**: Hiểu về phép toán modulo, đặc biệt là trong ngữ cảnh số nguyên lớn.
* **Lý thuyết nhóm**: Nắm vững các khái niệm cơ bản như nhóm, vành, trường, nhóm cyclic, và nhóm con.
* **Số nguyên tố và phân tích số nguyên tố**: Hiểu về các thuật toán kiểm tra tính nguyên tố và phân tích số nguyên lớn.
* **Hàm số hash**: Kiến thức về hàm băm mật mã như SHA-1, SHA-3, SHA-256, và cách chúng hoạt động.

### 1.2.2 Kiến thức về Thuật toán

* **Thuật toán số học**: Các thuật toán tính toán với số nguyên lớn như phép nhân, phép chia, và phép lũy thừa modulo.
* **Thuật toán Euclid mở rộng**: Dùng để tìm nghịch đảo modulo.
* **Thuật toán tạo số nguyên tố**: Hiểu cách tạo ra các số nguyên tố lớn cần thiết cho DSA.

### 1.2.3. Ngôn ngữ lập trình

* **C#**
* **Cấu trúc dữ liệu**: Hiểu các cấu trúc dữ liệu cơ bản trong C# (Array, List, Map).
* **Quản lý ngoại lệ**: Biết cách xử lý ngoại lệ khi làm việc với các thao tác mã hóa.
* **C++**
* **Thư viện mã hóa C++**: Sử dụng thư viện như cmath để thực hiện các chức năng tính toán số học.
* **Con trỏ và quản lý bộ nhớ**: Hiểu cách quản lý bộ nhớ động trong C++.
* **Xử lý lỗi**: Biết cách xử lý lỗi và ngoại lệ, đặc biệt trong ngữ cảnh mã hóa.
* **Kiểm tra tình hợp lệ dữ liệu**: Biết cách kiểm tra tính hợp lệ của dữ liệu đầu vào.

## 1.3 Nội dung nghiên cứu

### 1.3.1 Lý do nghiên cứu

* Phải che giấu được nội dung của văn bản rõ (plaintext)
* Tạo các yếu tố xác thực thông tin, đảm bảo thông tin lưu hành trong hệ thống đến người nhận hợp pháp là xác thực
* Tổ chức các sơ đồ chữ ký điện tử, đảm bảo k có hiện tượng giả mạo, mạo danh để gửi thông tin lên mạng

### 1.3.2 Khái niệm cơ bản

* Bản rõ X: được gọi là bản tin gốc, bản rõ có thể được chia nhỏ có kích thước phù hợp.
* Bản mã Y: là bản tin gốc đã được mã hoá.
* Mã: là thuật toán mã hoá chuyển bản rõ thành bản mã, thông thường chúng ta cần thuật toán mã hoá mạnh, cho dù kẻ thù biết được thuật toán, nhưng không biết thông tin về khoá thì cũng không tìm được bản rỏ
* Khoá K: là thông tin tham số dùng để mã hoá, chỉ có người gửi và người nhận biết. Khoá là độc lập với bản rõ và có độ dài phù hợp với yêu cầu bảo mật.
* Mã hoá: là quá trình chuyển bản rõ thành bản mã.
* Giải mã: là quá trình chuyển bản mã thành bản rõ, đây là quá trình ngược lại của mã hoá.

### 1.3.3 Các thành phần của hệ mật mã

Một hệ mật mã là một hệ bao gồm 5 thành phần (P, C, K, E, D) thoả mãn:

* P (Plaintext): không gian bản rõ, là tập hợp hữu hạn các bản rõ có thể.
* C (Ciphertext): không gian bản mã, là tập hợp những bản mã có thể.
* K (Key): không gian khoá, là tập hợp các khoá có thể.
* E (Encryption): không gian hàm mã hoá, là tập hợp các quy tắc mã hoá có thể.
* D (Decryption): không gian hàm giải mã, là tập hợp các quy tắc giải mã có thể.

Đối với mỗi k thuộc K có một quy tắc mã hoá ek: P -> C thuộc E và một quy tắc giải mã tương ứng dk: C -> P thuộc D

#### 1.3.4 Giới thiệu về hệ mã DSA.

#### 1.3.4.1 Giới thiệu chung về hệ mã DSA.

Hệ chữ ký số DSA (Digital Signature Algorithm) là một phương pháp chữ ký số được thiết kế để xác minh tính toàn vẹn và nguồn gốc của thông điệp điện tử. DSA được phát triển bởi Cơ quan An toàn Quốc gia của Mỹ (NSA) và được đưa vào sử dụng trong các ứng dụng an ninh và mật mã.

#### 1.3.4.2 Tạo chữ ký và xác minh chữ ký trong DSA.

* **Tạo chữ ký (Signing):** Để tạo chữ ký cho một thông điệp, người ký sử dụng khóa bí mật của mình để tạo ra một chữ ký dựa trên nội dung của thông điệp. Quá trình này bao gồm việc áp dụng một hàm băm để tạo ra một giá trị băm, sau đó sử dụng một phép toán số học để ký kết giá trị băm đó.
* **Xác minh chữ ký (Verification):** Người nhận thông điệp sử dụng khóa công khai của người ký để xác minh tính toàn vẹn của thông điệp và tính hợp pháp của chữ ký. Quá trình này bao gồm việc sử dụng cùng một hàm băm để tạo ra một giá trị băm từ nội dung của thông điệp, sau đó so sánh giá trị băm này với giá trị băm được ký kết trong chữ ký.

#### 1.3.4.3 Ưu điểm và nhược điểm của hệ mã DSA.

* **Ưu điểm**: DSA cung cấp một cơ chế hiệu quả để xác minh tính toàn vẹn và nguồn gốc của thông điệp mà không cần phải chia sẻ khóa bí mật.
* **Nhược điểm**: DSA có thể mất nhiều thời gian để tạo và xác minh chữ ký so với các phương pháp khác như RSA. Đồng thời, việc triển khai DSA cần sự chú ý đến việc chọn các thông số an toàn và đảm bảo tính ngẫu nhiên trong quá trình tạo chữ ký.

# CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU

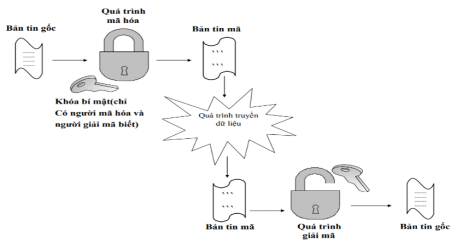
## 2.1 Nghiên cứu, tìm hiểu hệ mã hóa

### 2.1.1. Mã hóa bằng khóa bí mật

Các hệ thống mã hóa với khóa bí mật còn được gọi là mã hóa bằng khóa riêng, mã hóa đối xứng sử dụng duy nhất một khóa cho cả quá trình mã hóa lẫn quá trình giải mã.

* Stream Algorithms/Stream Ciphers: các thuật toán hoạt động trên văn bản bình thường theo từng bit một.
* Block Algorithms/Block Ciphers : các thuật toán hoạt động trên văn bản theo các khối (32 bit, 64 bit, 128 bit, ...).
* Một số thuật toán đang được sử dụng rộng rãi hiện nay : DES, TripleDES, RC5, RC6, Rijndael ...

Quá trình mã hóa và giải mã bằng cách sử dụng khóa bí mật được minh họa như hình dưới đây:



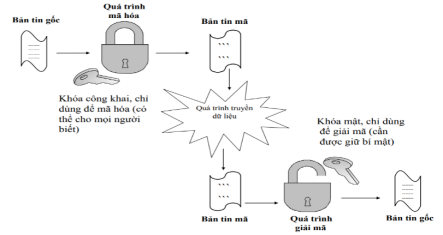
Hình 2.1 Sơ đồ mã hóa và giải mã bằng khóa riêng

### 2.1.2 Mã hóa bằng khóa công khai

Mã hóa bằng khóa công khai còn gọi là mã hóa bất đối xứng hay mã hóa bằng khóa chung. Sự khác biệt cơ bản giữa một hệ thống mã hóa bằng khóa bí mật với hệ thống mã hóa bằng khóa công khai là hệ thống mã hóa khóa công khai dùng hai khóa khác nhau để mã hóa và giải mã. Do đó, một bộ mã công khai sẽ bao gồm hai khóa: một khóa dành cho người mã hóa thường được công khai, và khóa còn lại dùng cho người giải mã thường được giữ bí mật. Như vậy, hệ thống mã hóa với khóa công khai cần có một quá trình sinh ra hai khóa để mã hóa và giải mã thông điệp. Các khóa này được xem như là một đôi:

* Public-key (khóa công khai): được phép công khai mà không phải chịu rủi ro về an toàn. Khóa này được dùng để mã hóa thông điệp.
* Private-key (khóa bí mật): không được để lộ. Mỗi thông điệp được mã hóa bằng public-key chỉ có thể giải mã bằng một khóa mật thích hợp.
* Một số thuật toán mã hóa công khai phổ biến : RSA, Diffie-Hellman KeyExchange Algorithm (dùng cho việc phân phối và trao đổi khóa).

Như vậy, với sự bùng nổ của mạng toàn cầu mọi hệ thống thông tin đều phải đương đầu với bài toán an toàn và bảo mật. Như đã trình bày, có nhiều chiến lược cũng như phương pháp bảo đảm bảo an toàn thông tin. Trong đó, an toàn thông tin bằng mật mã có vai trò pháp quan trọng và được ứng dụng rộng khắp không chỉ trong ngành công nghệ thông tin mà còn dùng để bảo mật những thông tin và tài liệu quan trọng ngoài đời. (Ví dụ như bảo mật đề thi trong tuyển sinh được đề cập đến trong chương sau)



Hình 2.2 Sơ đồ mã hóa và giải mã bằng khóa công khai

## Nghiên cứu, tìm hiểu về hệ mã hóa DSA

### 2.2.1. Các thành phần của DSA

DSA bao gồm các thành phần sau:

* **Số nguyên tố p:** Một số nguyên tố lớn, thường có độ dài từ 1024 đến 3072 bit.
* **Số nguyên tố q:** Một số nguyên tố nhỏ hơn, là ước số của p-1.
* **Số nguyên g:** Một số nguyên thỏa mãn g^q ≡ 1 (mod p).
* **Khóa bí mật x:** Một số nguyên ngẫu nhiên được chọn trong khoảng [0, q-1].
* **Khóa công khai y:** Được tính bằng công thức y = g^x mod p.

### 2.2.2 Quá trình ký

Để ký một thông điệp, người gửi thực hiện các bước sau:

1. **Tạo giá trị ngẫu nhiên k:** Chọn một số nguyên ngẫu nhiên k trong khoảng [1, q].
2. **Tính r:** r = (g^k mod p) mod q.
3. **Tính s:** s = (k^-1 (H(m) + x\*r)) mod q, trong đó H(m) là giá trị băm của thông điệp m.
4. **Chữ ký:** Chữ ký của thông điệp là cặp số (r, s).

### 2.2.3 Quá trình xác minh

Để xác minh chữ ký, người nhận thực hiện các bước sau:

1. **Tính w:** w = s^-1 mod q.
2. **Tính u1:** u1 = (H(m) \* w) mod q.
3. **Tính u2:** u2 = (r \* w) mod q.
4. **Tính v:** v = ((g^u1 \* y^u2) mod p) mod q.
5. **Xác minh:** Chữ ký hợp lệ nếu v = r.

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

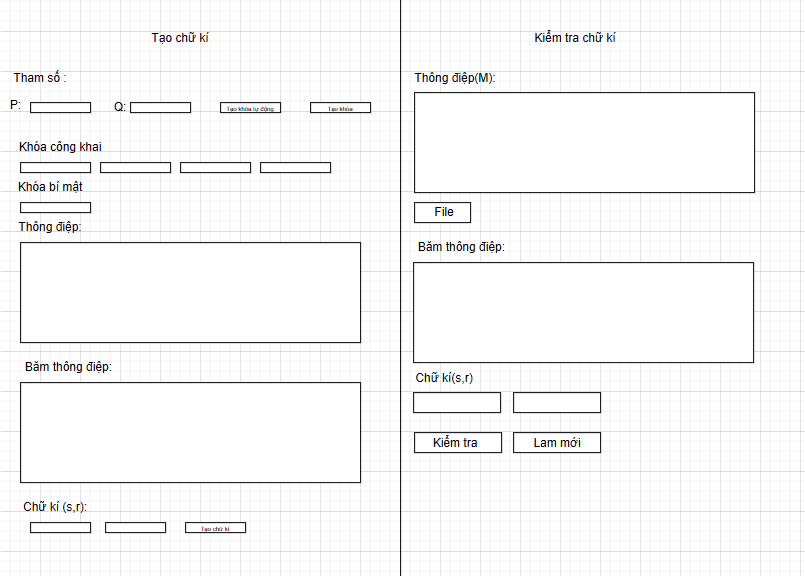
### Thiết kế kịch bản chương trình

#### 2.3.1.1. Phát sinh chữ ký:

* **Đầu vào:**
  + **File văn bản:** Người dùng chọn file văn bản cần ký.
  + **Chìa khóa:** Người dùng nhập hoặc chương trình tự động lấy chìa khóa bí mật tương ứng.
* **Xử lý:**
  + Chương trình sử dụng thuật toán mã hóa bất đối xứng (ví dụ: RSA, DSA) để tạo chữ ký số từ file văn bản và chìa khóa bí mật.
  + **Chữ ký:** Chữ ký số được tạo ra (thường là một chuỗi ký tự đặc biệt).
* **Đầu ra:**
  + **Chữ ký:** Chữ ký số được hiển thị hoặc lưu vào một file riêng biệt.
  + **Lưu:** Chữ ký có thể được lưu trữ để sử dụng sau này.

#### 2.3.1.2. Kiểm tra chữ ký:

* **Đầu vào:**
  + **File văn bản:** Người dùng chọn file văn bản đã được ký.
  + **File chữ ký:** Người dùng chọn file chứa chữ ký số tương ứng.
  + **Chìa khóa:** Người dùng nhập hoặc chương trình tự động lấy chìa khóa công khai của người ký.
* **Xử lý:**
  + Chương trình sử dụng thuật toán giải mã bất đối xứng để kiểm tra tính hợp lệ của chữ ký dựa trên file văn bản và chìa khóa công khai.
* **Đầu ra:**
  + **Thông báo:**
    - **Chữ ký đúng:** Nếu chữ ký hợp lệ, thông báo cho người dùng biết văn bản không bị chỉnh sửa từ khi được ký.
    - **Chữ ký không đúng:** Nếu chữ ký không hợp lệ, thông báo cho người dùng biết văn bản có thể đã bị chỉnh sửa hoặc chữ ký không phải của người sở hữu chìa khóa bí mật tương ứng.



Hình 2.3 Hình ảnh minh họa chữ ký số

### Giới thiệu ngôn ngữ lập trình sử dụng để cài đặt thuật toán

#### Ngôn ngữ lập trình C++

* Lịch sử hình thành C++

Trước C++, ngôn ngữ lập trình C được phát triển trong năm 1972 bởi  
Dennis Ritchie tại phòng thí nghiệm Bell Telephone, C chủ yếu là một ngôn ngữ  
lập trình hệ thống, một ngôn ngữ để viết ra hệ điều hành. Vào năm 1999, ủy ban ANSI đã phát hành một phiên bản mới của ngôn ngữ  
lập trình C, gọi là C99. C++ được tạo ra bởi Bjarne Stroustrup - một nhà khoa máy tính người Đan Mạch tại phòng thí nghiệm AT&T Bell vào năm 1979, được ISO  
công nhận vào năm 1998, lần phê chuẩn tiếp theo vào năm 2003 (người ta gọi là C++ 03). Hai lần cập nhật gần đây nhất là C++ 11 và C++ 14 (được phê chuẩn vào năm 2011 và 2014). Phiên bản C++ 17 đã được công bố, dự đoán sẽ hoàn thành trong năm 2017. C++ là một phiên bản mở rộng của ngôn ngữ lập trình C.



Hình 2.4 Ngôn ngữ lập trình C++

* Một số đặc trưng của ngôn ngữ C++
* C++ là một ngôn ngữ lập trình bậc trung. Nó có nghĩa là bạn có thể sử dụng C++ để phát triển những ứng dụng bậc cao, và cả những chương trình bậc thấp hoạt động tốt trên phần cứng.
* C++ là một ngôn ngữ lập trình hướng đối tượng.

Khác với ngôn ngữ lập trình C - một ngôn ngữ lập trình hướng thủ tục, chương trình được tổ chức theo thuật ngữ "chức năng", một chức năng gồm có những hành động mà bạn muốn làm. C++ được thiết kế với một cách tiếp cận hoàn toàn mới được gọi là lập trình hướng đối tượng, nơi mà chúng ta sử dụng những đối tượng, các lớp và sử dụng các khái niệm như: thừa kế, đa hình, tính đóng gói, tính trừu tượng...

Những khái niệm này khá phức tạp, nên nếu bạn chưa hiểu về chúng, đừng lo lắng, chúng ta sẽ lần lượt làm rõ từng khái niệm trong mỗi bài học khác nhau.

* C++ là một ngôn ngữ lập trình hướng cấu trúc giống ngôn ngữ C, Nó có nghĩa là chúng ta có thể tổ chức chương trình trên khái niệm function.
* C++ có thể chạy trên nhiều nền tảng khác nhau như Windows, Mac OS, một số biến thể của UNIX...
* Lý do chọn ngôn ngữ lập trình C++

Ngôn ngữ lập trình C++ được lựa chọn vì hiệu năng cao và khả năng kiểm soát bộ nhớ tốt, phù hợp cho các ứng dụng mật mã (DSA).

Dưới đây là một số điểm quan trọng về việc sử dụng C++ để triển khai DSA:

1. Cấu trúc Dữ liệu (Data Structures):
   * C++ cung cấp nhiều kiểu dữ liệu và cấu trúc dữ liệu như mảng, danh sách liên kết, cây, đồ thị, hàng đợi, ngăn xếp và nhiều loại khác.
   * Sử dụng các cấu trúc này để tổ chức và lưu trữ dữ liệu một cách hiệu quả.
2. Giải thuật (Algorithms):
   * C++ hỗ trợ việc triển khai các thuật toán quan trọng như tìm kiếm, sắp xếp, đệ quy, duyệt đồ thị và nhiều thuật toán khác.
   * Các thư viện chuẩn của C++ như STL (Standard Template Library) cung cấp các hàm và lớp tiện ích cho việc triển khai thuật toán.
3. Hiệu suất và tối ưu hóa:
   * C++ cho phép tối ưu hóa mã nguồn, giúp bạn viết các thuật toán nhanh chóng và hiệu quả.
   * Sử dụng con trỏ và thao tác trực tiếp với bộ nhớ để tối ưu hóa hiệu suất.
4. Thư viện và tài liệu phong phú:
   * C++ có nhiều thư viện hỗ trợ cho DSA, bao gồm STL, Boost và nhiều thư viện khác.
   * Tài liệu và ví dụ phong phú giúp bạn học và triển khai DSA một cách hiệu quả.

#### Ngôn ngữ lập trình C#

C# được phát hành lần đầu vào năm 2000 bởi Microsoft, là một phần trong nền tảng .NET Framework. Đây là ngôn ngữ lập trình hướng đối tượng, được thiết kế để đơn giản hóa việc phát triển ứng dụng, phù hợp với cả những người mới học lập trình và các lập trình viên chuyên nghiệp.

**Đặc điểm nổi bật của C#:**

* **Tính trừu tượng:** Tập trung vào việc xác định và nhóm các thuộc tính, hành động của một thực thể cụ thể, hỗ trợ mô hình hóa đối tượng thực tế.
* **Tính đa hình:** Cho phép sử dụng các phương thức cùng tên trên nhiều đối tượng khác nhau, hỗ trợ dễ dàng mở rộng và tái sử dụng mã nguồn.
* **Tính kế thừa:** Được thiết kế để cho phép chia sẻ và mở rộng các thuộc tính, hành vi từ lớp cơ sở.
* **Tính đóng gói:** Hỗ trợ bảo vệ dữ liệu, che giấu chi tiết thực thi của các đối tượng, cung cấp các phương thức truy cập được kiểm soát.



Hình 2.5 Ngôn ngữ lập trình C#

**Ứng dụng của C#:**

**Phát triển ứng dụng web:**

* C# được sử dụng rộng rãi trong các ứng dụng web nhờ tích hợp tốt với ASP.NET, hỗ trợ xây dựng các hệ thống bảo mật cao và có hiệu suất ổn định.
* Các ứng dụng như trang web thương mại điện tử, hệ thống quản lý, và API đều thường được phát triển bằng C#.

**Phát triển ứng dụng desktop:**

* C# hỗ trợ phát triển các ứng dụng chạy trên Windows thông qua Windows Forms, WPF (Windows Presentation Foundation).
* Các ứng dụng được viết một lần, dễ dàng triển khai trên nhiều hệ thống mà không cần viết lại mã.

**Phát triển ứng dụng di động:**

* Với Xamarin, C# có thể sử dụng để viết các ứng dụng di động chạy trên iOS, Android và Windows Phone. Điều này giúp tiết kiệm thời gian và công sức nhờ tính năng "viết một lần, chạy mọi nơi."

**Phát triển trò chơi:**

* C# được sử dụng trong Unity Engine – một nền tảng phát triển game phổ biến. Điều này giúp C# trở thành lựa chọn hàng đầu cho các nhà phát triển game.

**Các công cụ hỗ trợ lập trình C#:**

**Visual Studio IDE**

Visual Studio là công cụ hỗ trợ lập trình hàng đầu cho C#, được Microsoft phát triển. Đây là một IDE mạnh mẽ với nhiều tính năng hỗ trợ người lập trình, từ viết mã, gỡ lỗi, đến triển khai ứng dụng.

**Ưu điểm:**

* **Hỗ trợ đa nền tảng:** Visual Studio hỗ trợ Windows, macOS và có phiên bản nhẹ hơn là Visual Studio Code chạy trên cả Linux.
* **Tích hợp đầy đủ:** Cung cấp môi trường tích hợp với khả năng thiết kế giao diện, quản lý cơ sở dữ liệu, kiểm tra hiệu năng ứng dụng.
* **Hỗ trợ gỡ lỗi:** Visual Studio cung cấp các công cụ gỡ lỗi mạnh mẽ, cho phép kiểm tra từng bước mã nguồn.
* **Cộng đồng lớn:** Với cộng đồng đông đảo, tài liệu và các plugin phong phú, việc học và phát triển ứng dụng bằng C# trở nên dễ dàng hơn.

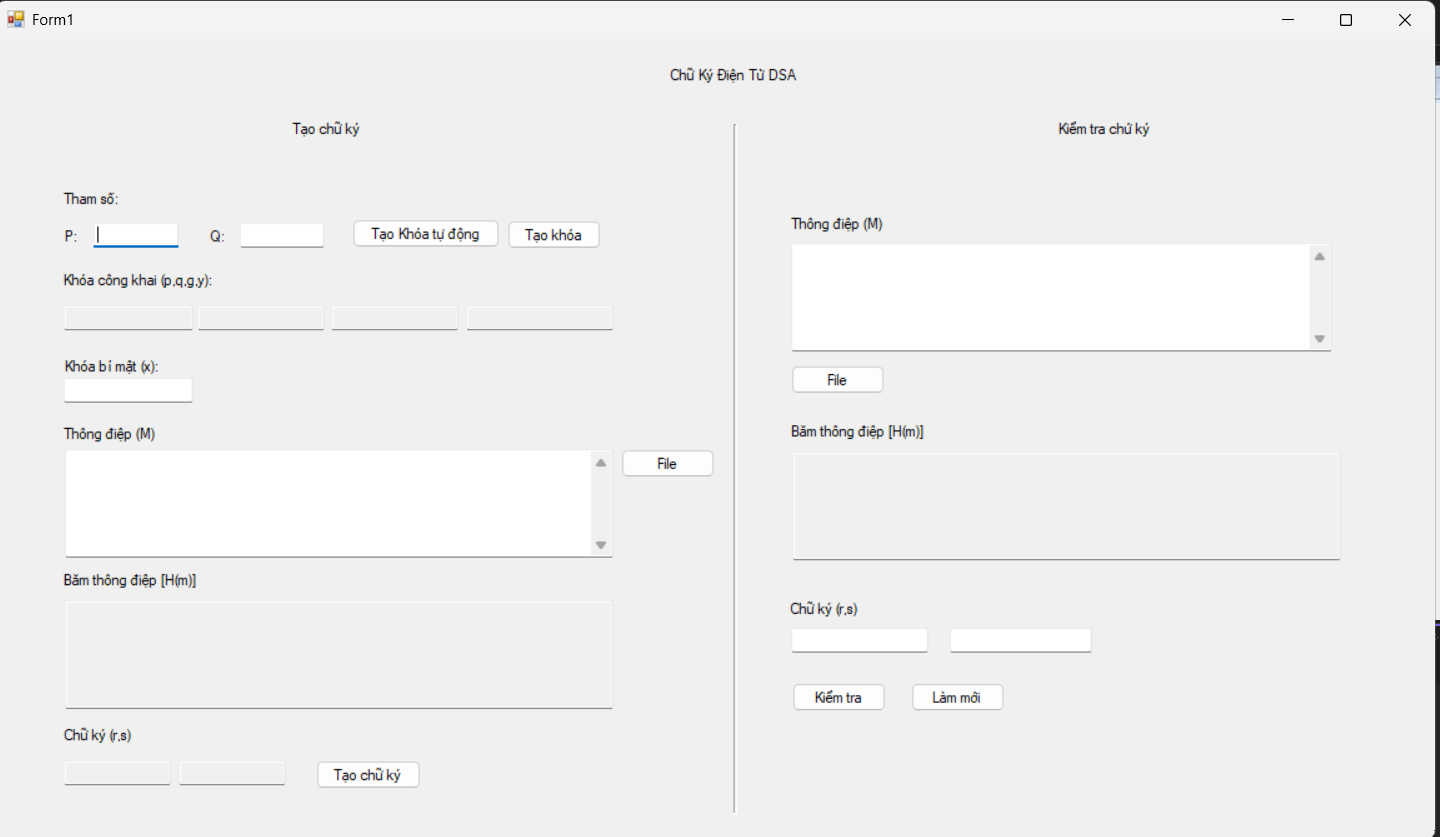
**Nhược điểm:**

* **Tốn tài nguyên:** Visual Studio cần cấu hình máy tính cao để hoạt động mượt mà, đặc biệt khi làm việc với dự án lớn.
* **Khởi động chậm:** So với một số IDE nhẹ hơn, Visual Studio có thời gian khởi động khá lâu.

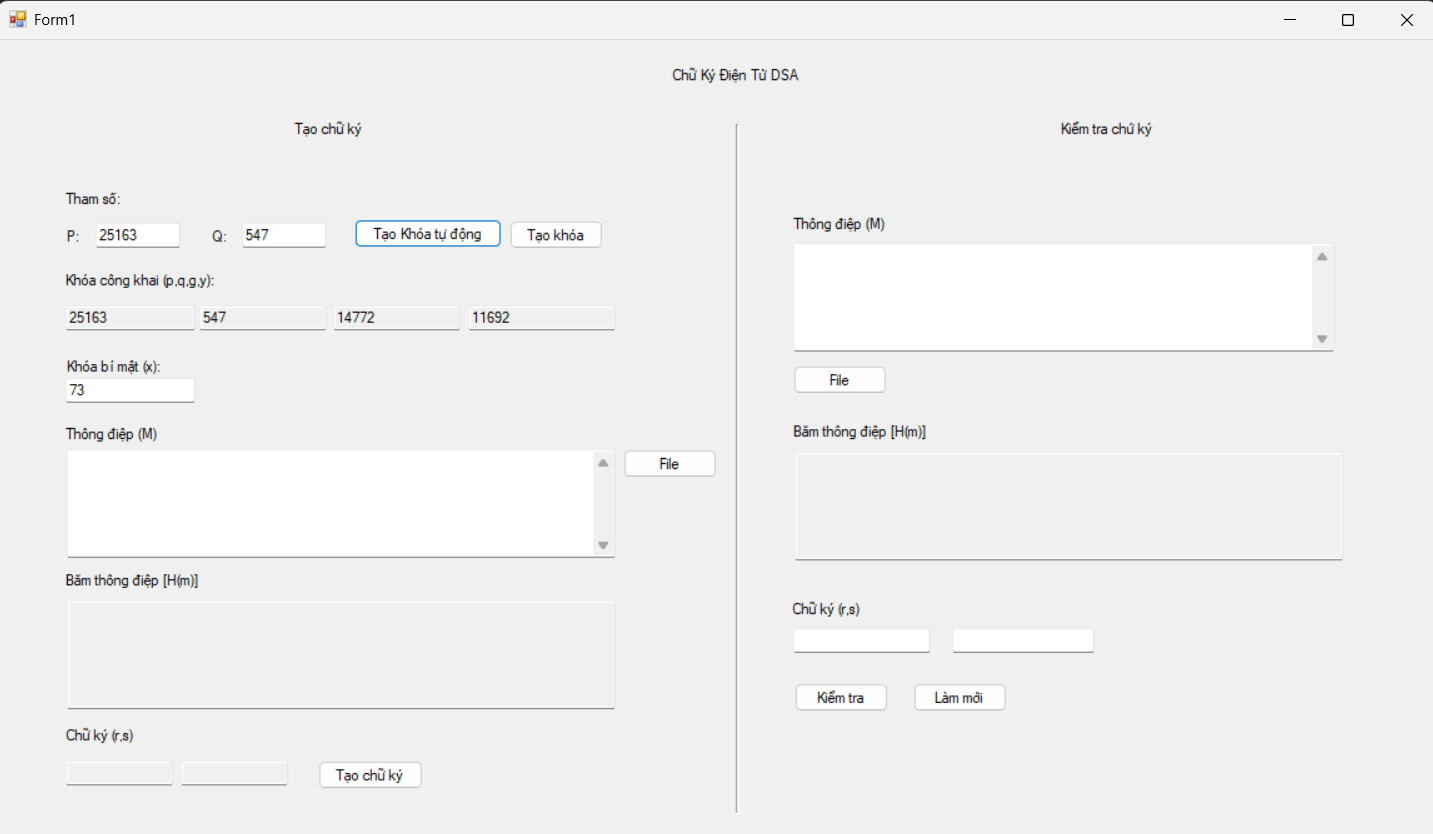
### Cài đặt thuật toán, giao diện chương trình

#### Ngôn ngữ C++

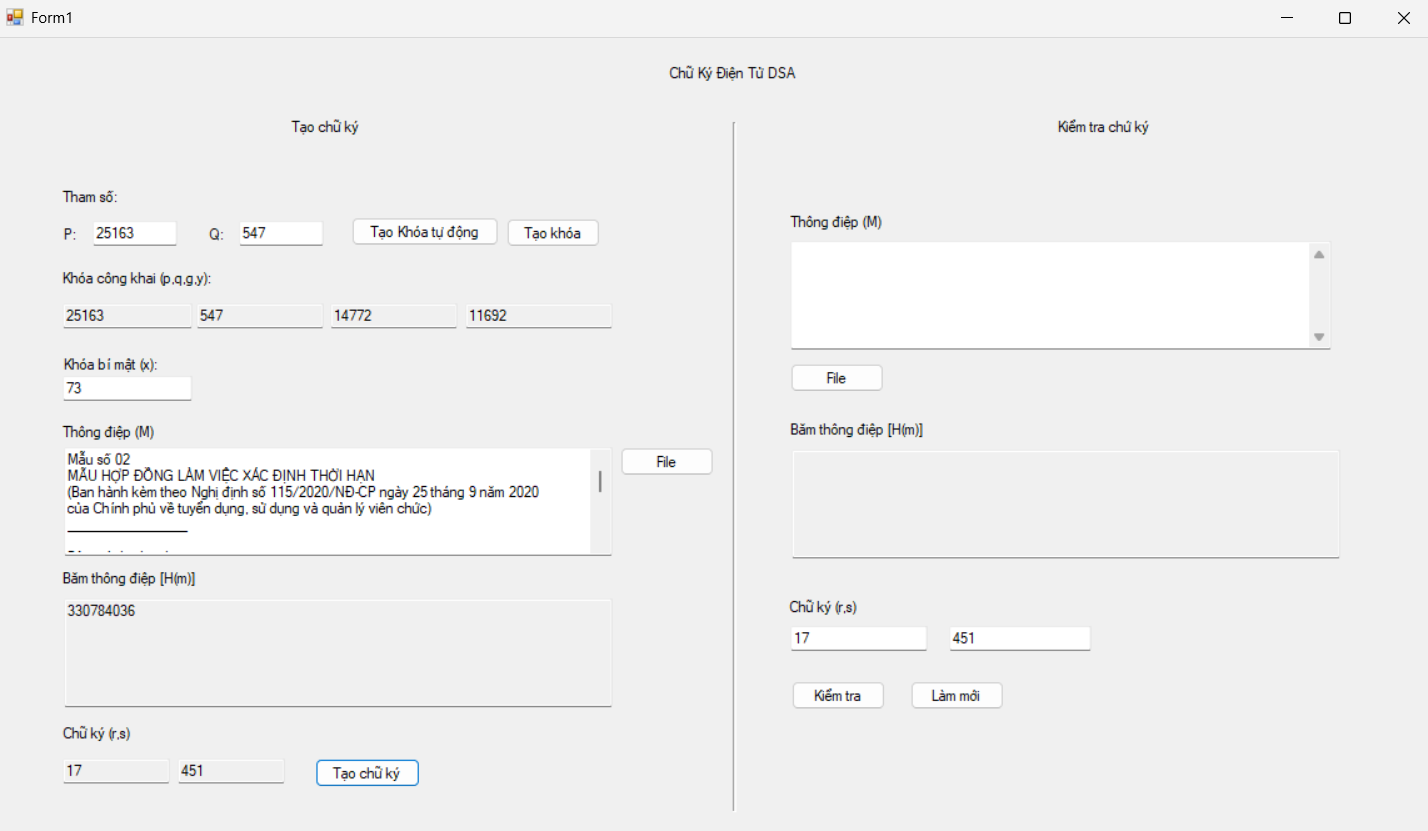
* **Các màn hình giao diện:**



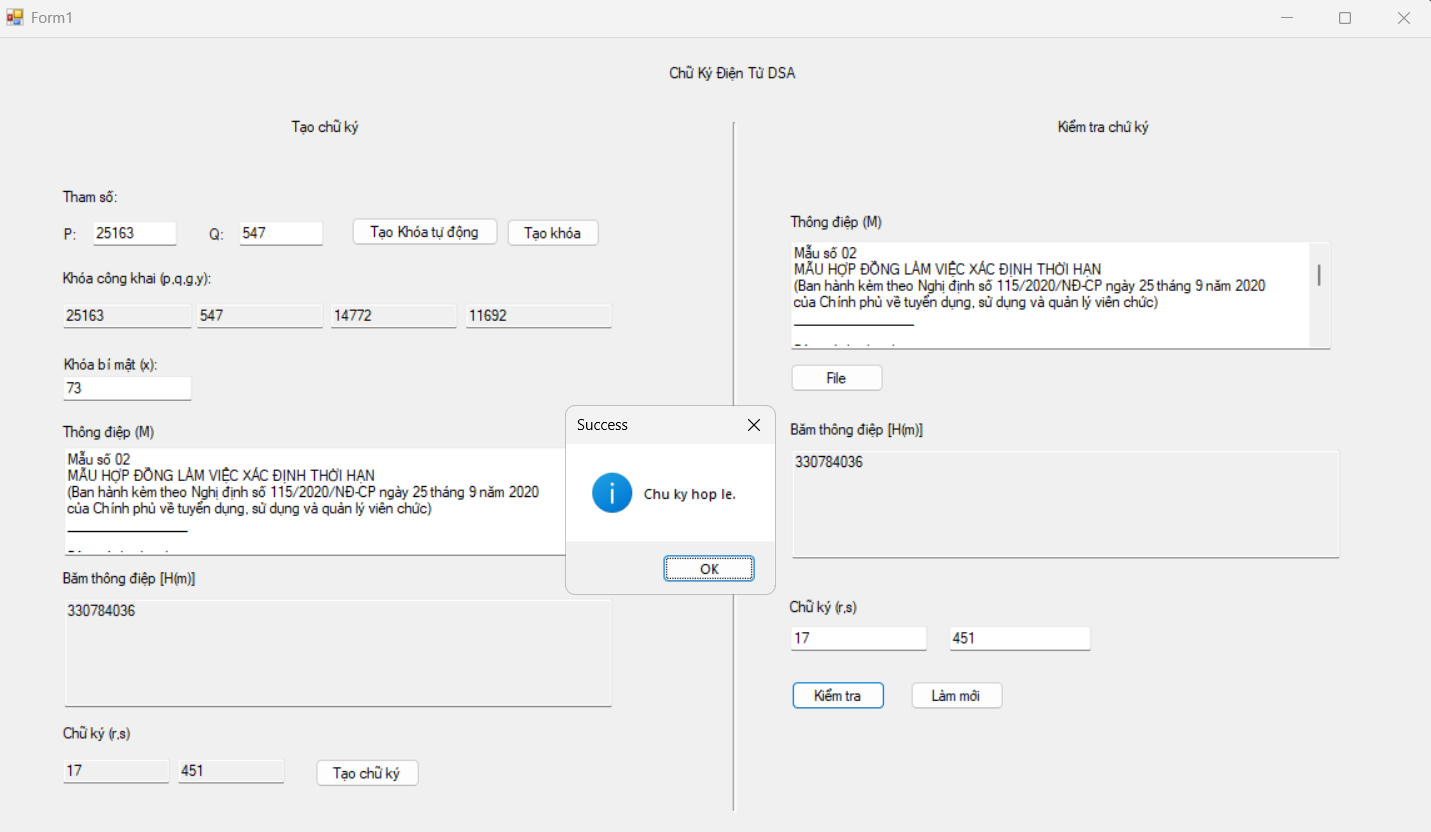
Hình 2.6 Giao diện bắt đầu



Hình 2.7 Giao diện khi tạo khóa

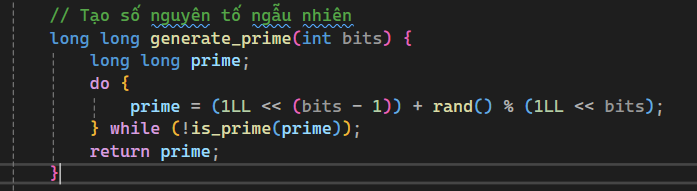


Hình 2.8 Màn hình khi thêm văn bản kí

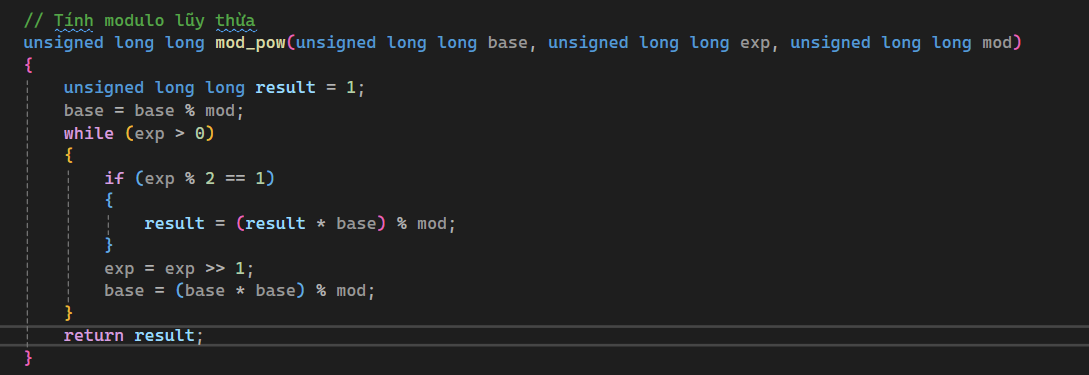


Hình 2.9 Màn hình khi xác nhận chữ kí

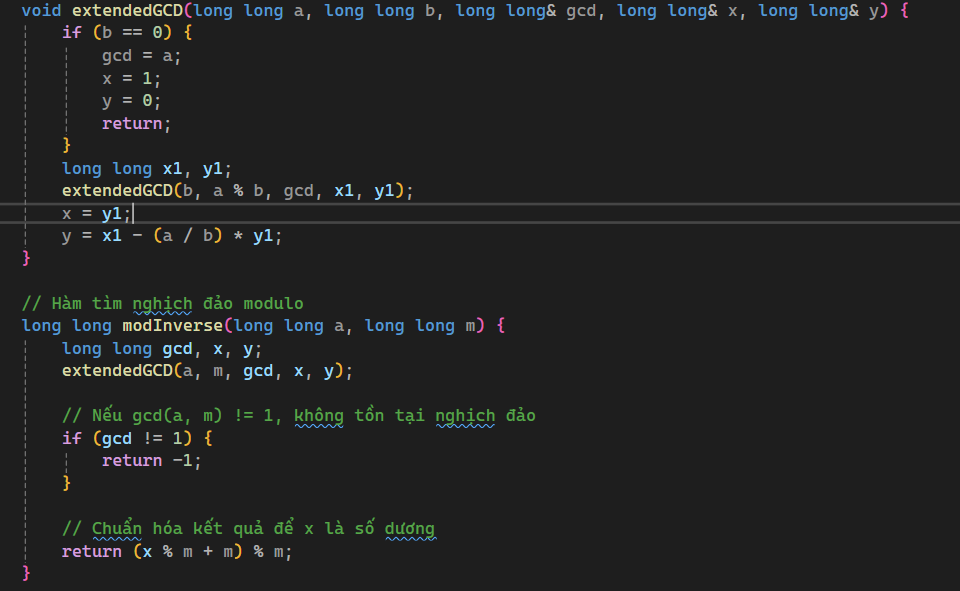
* **Cài đặt thuật toán:**
* Hàm sinh ngẫu nhiên



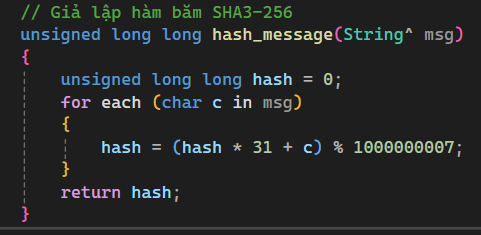
* Hàm a mũ b mod n



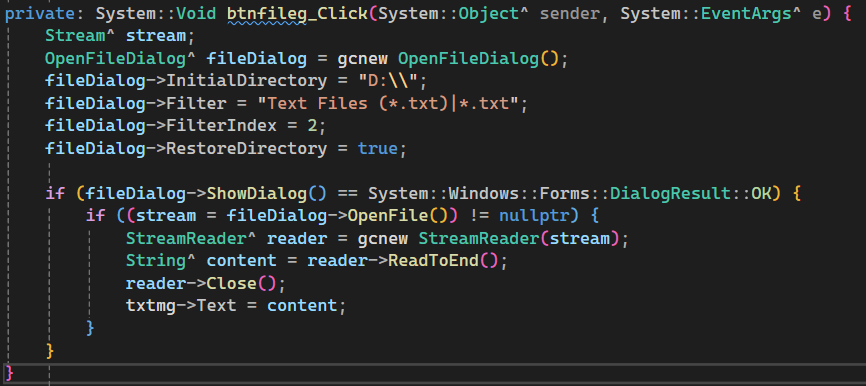
* Hàm nghịch đảo



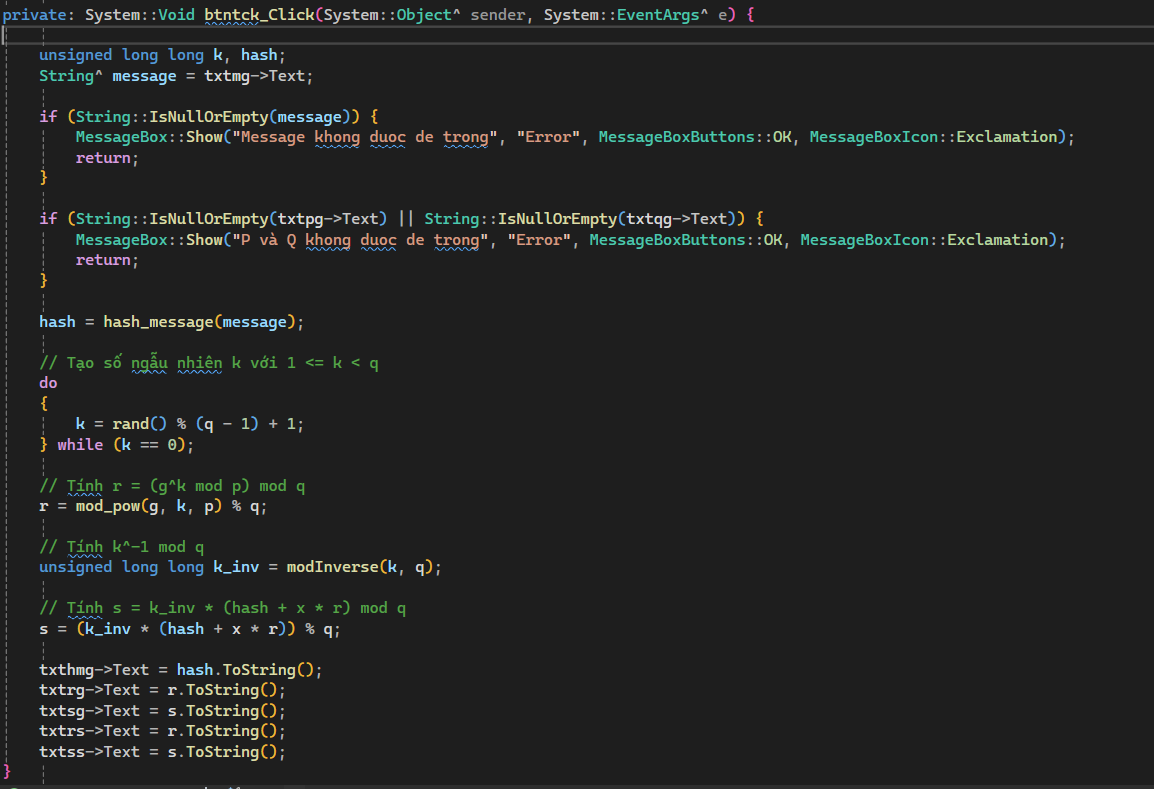
* Tính hàm băm



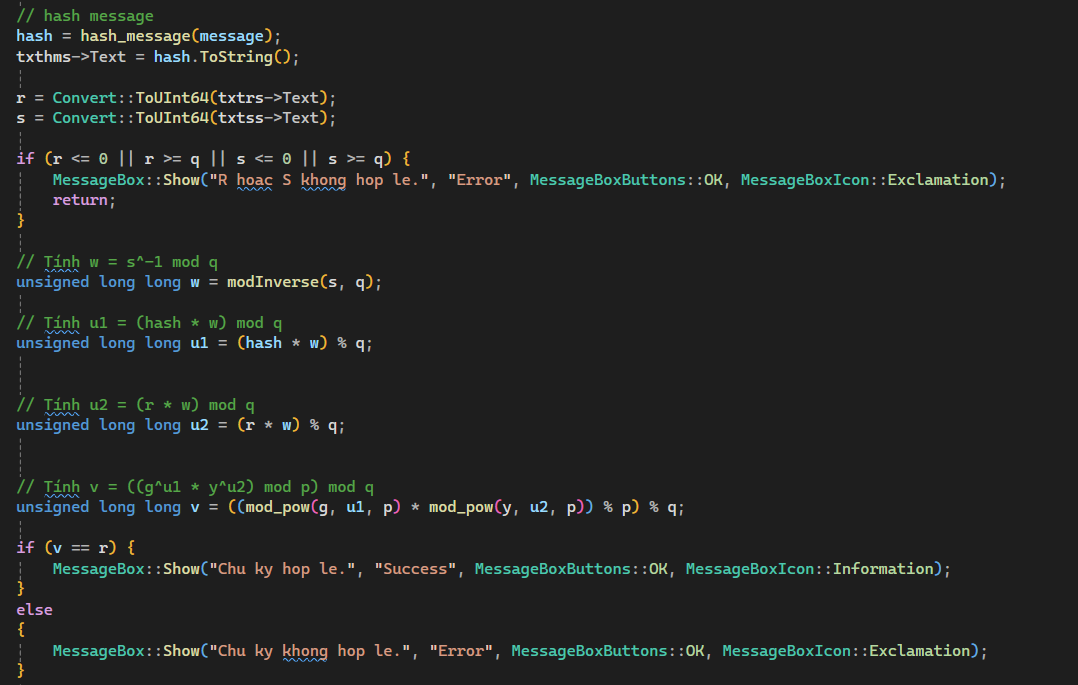
* Hàm đọc file đã chọn



* Hàm tạo chữ ký

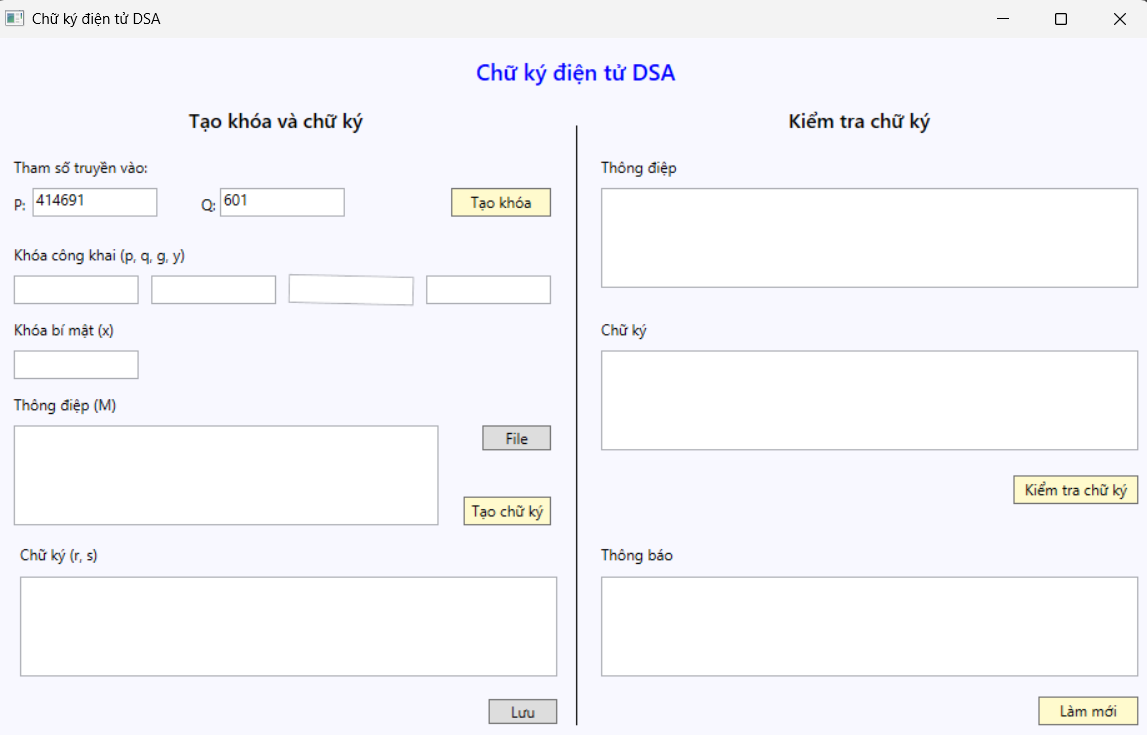


* Hàm kiểm tra khóa



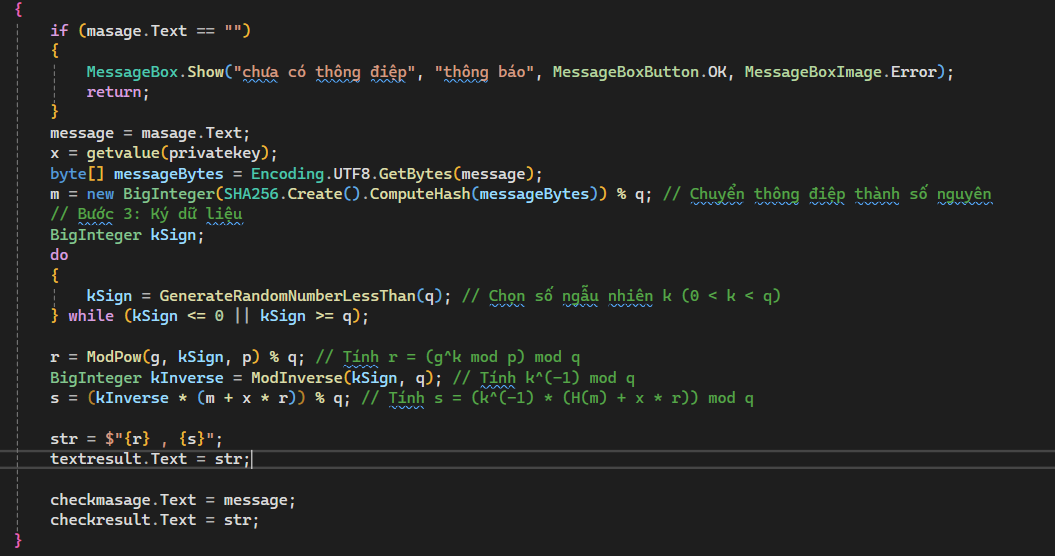
#### Ngôn ngữ C#

* **Giao diện chương trình**

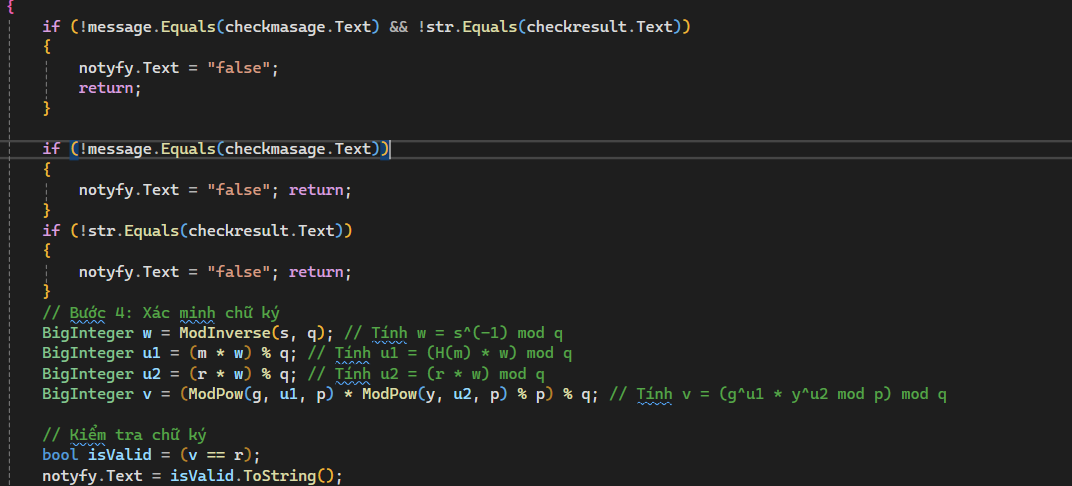
****

Hình 2.10 Giao diện chương trình C#

* **Thuật toán:**
* Phát sinh chữ ký

****

* Kiểm tra chữ ký



# CHƯƠNG 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM

## Kiến thức kỹ năng đã học được trong quá trình thực hiện đề tài.

Trong quá trình thực hiện đề tài về hệ chữ ký số DSA và phát triển ứng dụng minh họa bằng ngôn ngữ C# và C++, chúng tôi đã học được nhiều kiến thức và kỹ năng quan trọng:

* **Kiến thức về chữ ký số DSA**: Hiểu rõ nguyên lý hoạt động, cấu trúc và các thành phần của hệ chữ ký số DSA, bao gồm quá trình tạo khóa, ký và xác thực chữ ký.
* **Lập trình C# và C++**: Cải thiện kỹ năng lập trình trong cả hai ngôn ngữ này, đặc biệt là trong việc quản lý bộ nhớ, xử lý chuỗi và sử dụng các thư viện mật mã.
* **Sử dụng thư viện mật mã**: Biết cách sử dụng các thư viện mật mã phổ biến như Cryptography trong C# để thực hiện các thao tác liên quan đến DSA.
* **Phân tích và thiết kế hệ thống**: Nắm vững quy trình phân tích và thiết kế hệ thống, từ việc xác định yêu cầu, thiết kế kiến trúc đến triển khai và kiểm thử.
* **Bảo mật thông tin**: Hiểu rõ hơn về các nguyên tắc và phương pháp bảo mật thông tin, cách thức bảo vệ dữ liệu trong quá trình truyền và lưu trữ.

## Bài học kinh nghiệm

Qua quá trình thực hiện đề tài, chúng tôi đã rút ra được một số bài học kinh nghiệm quý báu:

* **Lập kế hoạch chi tiết**: Việc lập kế hoạch chi tiết và phân chia công việc cụ thể ngay từ đầu giúp đảm bảo tiến độ và chất lượng của dự án.
* **Tìm hiểu sâu về lý thuyết**: Nắm vững lý thuyết trước khi bắt tay vào lập trình là rất quan trọng để tránh những sai sót cơ bản và hiểu rõ mục đích của từng bước trong quá trình thực hiện.
* **Thực hành nhiều hơn**: Lập trình là một kỹ năng yêu cầu thực hành liên tục. Việc thường xuyên viết mã và giải quyết các vấn đề phát sinh giúp cải thiện kỹ năng lập trình và tư duy logic.
* **Kiểm thử kỹ lưỡng**: Kiểm thử kỹ lưỡng và đa dạng các trường hợp (test cases) giúp phát hiện sớm các lỗi và đảm bảo tính chính xác của chương trình.
* **Hợp tác và giao tiếp hiệu quả**: Làm việc nhóm yêu cầu sự hợp tác và giao tiếp hiệu quả giữa các thành viên, điều này giúp dự án diễn ra suôn sẻ và giải quyết nhanh chóng các vấn đề.

## Đề xuất về tính khả thi của chủ đề nghiên cứu , những thuận lợi và khó khăn

* **Tính khả thi của chủ đề nghiên cứu**:

Chủ đề "Tìm hiểu về hệ chữ ký DSA và viết ứng dụng minh họa" là một chủ đề khả thi với nhiều tiềm năng:

* **Ứng dụng thực tiễn**: Chữ ký số DSA được sử dụng rộng rãi trong các hệ thống bảo mật thông tin, chữ ký điện tử, và các giao dịch trực tuyến, nên việc hiểu và áp dụng nó mang lại giá trị thực tiễn cao.
* **Tài liệu và công cụ hỗ trợ phong phú**: Có nhiều tài liệu và thư viện mã nguồn mở hỗ trợ cho việc nghiên cứu và triển khai DSA, giúp giảm bớt khó khăn trong quá trình thực hiện.
* **Thuận lợi**:
* **Nhu cầu thị trường**: Bảo mật thông tin là một lĩnh vực đang phát triển mạnh mẽ, do đó kiến thức và kỹ năng về DSA sẽ có nhiều cơ hội ứng dụng và phát triển trong tương lai.
* **Sự hỗ trợ từ các công cụ và thư viện**: Có nhiều thư viện và công cụ hỗ trợ cho việc triển khai DSA, giúp tiết kiệm thời gian và công sức.
* **Khó khăn**:
* **Độ phức tạp của thuật toán**: DSA là một thuật toán phức tạp đòi hỏi kiến thức sâu rộng về toán học và lập trình, điều này có thể là một thách thức đối với những người mới bắt đầu.
* **Vấn đề bảo mật**: Việc triển khai các hệ thống bảo mật yêu cầu độ chính xác cao và tuân thủ các tiêu chuẩn an toàn nghiêm ngặt, đòi hỏi phải kiểm tra và kiểm thử kỹ lưỡng để tránh các lỗ hổng bảo mật.

Tóm lại, chủ đề nghiên cứu về hệ chữ ký DSA không chỉ khả thi mà còn mang lại nhiều lợi ích thiết thực. Tuy nhiên, để thành công, cần phải có kế hoạch chi tiết, sự kiên nhẫn và nỗ lực trong quá trình thực hiện.

# TÀI LIỆU THAM KHẢO

[1]. Bùi Doãn Khanh, Nguyễn Đình Thúc, *Mã hóa thông tin – Lý thuyết và ứng dụng*, NXB Lao động xã hội, 2011.

[2]. Nguyễn Xuân Dũng, *Bảo mật thông tin – Mô Hình và ứng dụng*, NXB thống kê, 2009.