

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC**

---



**BÁO CÁO THỰC TẬP KỸ THUẬT**

**HỆ MÃ RSA VÀ CHỮ KÝ SỐ**

**Cơ sở thực tập: Tổng công ty Công nghệ và Giải pháp CMC**

**Người hướng dẫn: TS. Vũ Thành Nam**

**Sinh viên: Nguyễn Minh Quân**

**Lớp: KSTN Toán Tin - K61**

**HÀ NỘI, 7/2020**

---

## NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN

1. Mục đích và nội dung của báo cáo

2. Kết quả đạt được

3. Ý thức làm việc của sinh viên

Hà Nội, ngày      tháng 07 năm 2020

Người hướng dẫn  
(Ký và ghi rõ họ tên)

# Mục lục

<b>1</b>	<b>Tổng quan về chữ ký số</b>	<b>1</b>
1.1	Giới thiệu về hệ mật RSA . . . . .	1
1.2	Cơ sở lý thuyết về chữ ký số . . . . .	3
1.3	Lý thuyết hàm băm mật mã . . . . .	5
<b>2</b>	<b>Xây dựng chương trình</b>	<b>7</b>
2.1	Chữ ký điện tử và thiết bị HSM . . . . .	7
2.2	Phân tích thiết kế hệ thống . . . . .	9
2.3	Xây dựng chức năng . . . . .	12
<b>3</b>	<b>Cài đặt và kết quả thực nghiệm</b>	<b>14</b>
3.1	Cài đặt core và API . . . . .	14
3.2	Xây dựng giao diện . . . . .	18
3.3	Đánh giá và kết luận . . . . .	21

# Lời mở đầu

Mật mã học là một trong những vấn đề quan trọng trong lĩnh vực bảo mật và an toàn thông tin. Trên thế giới, mật mã học đã được ra đời từ thời La Mã cổ đại và ngày càng được nghiên cứu, phát triển đạt được những thành tựu to lớn. Trong mật mã học, vấn đề bảo mật luôn đi đôi với vấn đề xác thực thông tin, đặc biệt trong hệ thống mã hóa khóa công khai vấn đề xác thực là vô cùng quan trọng. Để giải quyết vấn đề trên người ta đưa ra một cách giải quyết hiệu quả, đó là chữ ký số.

Với sự bùng nổ của mạng Internet hiện nay, mạng máy tính đang ngày càng đóng vai trò thiết yếu trong mọi lĩnh vực hoạt động an toàn của 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 ngành công nghệ thông tin, mật mã học còn được áp dụng nhiều trong lĩnh vực khác như ngân hàng, viễn thông,...

Mật mã học khóa công khai tạo ra chữ ký số và ứng dụng trong các tài liệu. Hệ mã hóa RSA- hệ mã hóa điển hình của mật mã công khai cùng với hàm băm mật mã học một chiều chính là những công cụ chính trong việc tạo ra chữ ký số điện tử.

Bài báo cáo này tập trung chủ yếu vào sơ đồ chữ ký số RSA và ứng dụng của nó.

1. **Chương 1** : Tổng quan về chữ ký số, chữ ký số RSA.
2. **Chương 2** : Phân tích thiết kế và xây dựng chức năng cho chương trình.
3. **Chương 3** : Cài đặt và kết quả thực nghiệm.

Để hoàn thành bài báo cáo thực tập kỹ thuật này, em xin gửi lời cảm ơn tới thầy giáo TS. Vũ Thành Nam đã có những nhận xét, góp ý trong phần trình bày của em để bài báo cáo của em được hoàn thiện hơn và em cũng xin gửi lời cảm ơn đến cơ sở thực tập: tổng công ty Công Nghệ và Giải pháp CMC đã tạo những điều kiện thuận lợi trong thời gian em thực tập. Trong quá trình thực hiện do kiến thức còn nhiều hạn chế và thời gian có giới hạn, nên bài báo cáo chắc chắn vẫn còn những thiếu sót, em rất mong những nhận xét và góp ý của thầy cô và các bạn sinh viên để bài báo cáo được hoàn thiện hơn.

Hà Nội, ngày 10 tháng 07 năm 2020

# Danh sách hình vẽ

2.1	Sơ đồ chữ ký số điện tử. . . . .	8
3.1	Cấu trúc file của chương trình. . . . .	14
3.2	Cài đặt thư viện và kết nối với cơ sở dữ liệu. . . . .	15
3.3	Các chức năng cơ bản của người dùng. . . . .	16
3.4	Giao diện đăng nhập người dùng của ứng dụng. . . . .	18
3.5	Giao diện đăng ký người dùng của ứng dụng. . . . .	19
3.6	Giao diện chính của chương trình. . . . .	19
3.7	Sinh cặp khóa bí mật, khóa công khai với độ dài tùy chọn. . . . .	20
3.8	Tạo chữ ký số với văn bản tương ứng. . . . .	20
3.9	Xác thực chữ ký số với văn bản tương ứng nhận được. . . . .	20

# Chương 1

## Tổng quan về chữ ký số

### 1.1 Giới thiệu về hệ mật RSA

Thuật toán RSA được Ron Rivest, Adi Shamir và Len Adleman mô tả lần đầu tiên vào năm 1977 tại Học viện Công nghệ Massachusetts (MIT). Tên của thuật toán lấy từ 3 chữ cái đầu của tên 3 tác giả. Thuật toán RSA được MIT đăng ký bằng sáng chế tại Hoa Kỳ vào năm 1983 (Số đăng ký 4,405,829).

Thuật toán RSA có hai khóa: khóa công khai (hay khóa công cộng) và khóa bí mật (hay khóa cá nhân). Mỗi khóa là những số cố định sử dụng trong quá trình mã hóa và giải mã. Khóa công khai được công bố rộng rãi cho mọi người và được dùng để mã hóa. Những thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa bí mật tương ứng. Nói cách khác, mọi người đều có thể mã hóa nhưng chỉ có người biết khóa cá nhân (bí mật) mới có thể giải mã được. Ta có thể mô phỏng trực quan một hệ mật mã khóa công khai như sau : Bob muốn gửi cho Alice một thông tin mật mà Bob muốn duy nhất Alice có thể đọc được. Để làm được điều này, Alice gửi cho Bob một chiếc hộp có khóa đã mở sẵn và giữ lại chìa khóa. Bob nhận chiếc hộp, cho vào đó một tờ giấy viết thư bình thường và khóa lại (như loại khóa thông thường chỉ cần sập chốt lại, sau khi sập chốt khóa ngay cả Bob cũng không thể mở lại được-không đọc lại hay sửa thông tin trong thư được nữa). Sau đó Bob gửi chiếc hộp lại cho Alice. Alice mở hộp với chìa khóa của mình và đọc thông tin trong thư. Trong ví dụ này, chiếc hộp với khóa mở đóng vai trò khóa công khai, chiếc chìa khóa chính là khóa bí mật.

**Định nghĩa 1.1.1.** (*Hệ mật mã*)

Hệ mật mã là bộ 5  $S = (P, C, K, E, D)$  thỏa mãn các điều kiện sau:

- $P$  là tập hữu hạn các từ hiện.
- $C$  là tập hữu hạn các từ mã.
- $K$  là không gian mã, tập hợp các khóa mã.
- Với mỗi khóa  $k \in K$ , có một quy tắc mã  $e_k \in E$  và  $d_k \in D$ . Các hàm  $e_k : P \rightarrow C, d_k : C \rightarrow P$  thỏa mãn  $d_k(e_k(x)) = x$  với mỗi từ hiện  $x \in P$ .

**Định nghĩa 1.1.2.** (Hệ mã RSA)

$$S = (P, C, K, E, D), P = C = Z_n$$

- Đặt  $n = p \cdot q$  trong đó  $p$  và  $q$  là hai số nguyên tố lớn.
- Hàm số Euler  $\phi(n) = (p - 1)(q - 1)$ .
- $K = \{n, p, q, a, b\}, n = pq, ab \equiv 1 \pmod{\phi(n)}$
- Với mỗi  $k \in K, e_k(x) = x^b \pmod{n}, d_k(y) = y^a \pmod{n}, x, y \in Z_n$
- Khóa công khai  $(n, b)$  và khóa bí mật  $(p, q, a)$ .

Độ an toàn của hệ thống RSA dựa trên hai vấn đề của toán học: bài toán phân tích ra thừa số nguyên tố các số nguyên lớn và bài toán RSA. Nếu hai bài toán trên là khó (không tìm được thuật toán hiệu quả để giải chúng) thì không thể thực hiện được việc phá mã toàn bộ đối với RSA.

Tại thời điểm năm 2005, số lớn nhất có thể được phân tích ra thừa số nguyên tố có độ dài 663 bit với phương pháp phân tán trong khi khóa của RSA hiện nay thường có độ dài từ 1024 tới 2048 bit. Một số chuyên gia cho rằng khóa 1024 bit có thể sớm bị phá vỡ (cũng có nhiều người phản đối việc này). Với khóa 4096 bit thì hầu như không có khả năng bị phá vỡ trong tương lai gần. Do đó, người ta thường cho rằng RSA đảm bảo an toàn với điều kiện  $n$  được chọn đủ lớn. Nếu  $n$  có độ dài 256 bit hoặc ngắn hơn, nó có thể bị phân tích trong vài giờ với máy tính cá nhân dùng các phần mềm có sẵn. Nếu  $n$  có độ dài 512 bit, nó có thể bị phân tích bởi vài trăm máy tính tại thời điểm năm 1999. Một thiết bị lý thuyết có tên là TWIRL do Shamir và Tromer mô tả năm 2003 đã đặt ra câu hỏi về độ an toàn của khóa 1024 bit. Vì vậy hiện nay người ta khuyến cáo sử dụng khóa có độ dài tối thiểu 2048 bit. Năm 1993, Peter Shor công bố thuật toán Shor chỉ ra rằng: máy tính lượng tử (trên lý thuyết) có thể giải bài toán phân tích ra thừa số trong thời gian đa thức. Tuy nhiên, máy tính lượng tử vẫn chưa thể phát triển được tới mức độ này trong nhiều năm nữa.

## 1.2 Cơ sở lý thuyết về chữ ký số

Trong một văn bản, chữ ký viết tay là một minh chứng về bản quyền, nhằm xác nhận các nội dung trong văn bản. Chữ ký viết tay được chính tay người ký nên không thể sao chụp được. Trong một văn bản, chữ ký viết tay là một minh chứng về bản quyền, nhằm xác nhận các nội dung trong văn bản. Chữ ký viết tay được chính tay người ký nên không thể sao chụp được. Một số đặc điểm của chữ ký viết tay:

- Chữ ký là bằng chứng thể hiện người ký có chủ định khi ký văn bản.
- Chữ ký thể hiện chủ quyền, nó cho người nhận thông tin về người ký văn bản.
- Văn bản đã ký thì không thể thay đổi được, người đã ký văn bản không thể phủ định việc mình đã ký văn bản, vậy chữ ký là thứ không thể chối bỏ.
- Chữ ký có thể giả mạo, tuy nhiên với khả năng kiểm định, việc giả mạo chữ ký là không hề đơn giản.

Với xu hướng toàn cầu hóa, việc trao đổi thông tin cần nhanh gọn, chính xác và đặc biệt phải an toàn. Việc trao đổi thông tin, chứng thực thông tin theo phong cách truyền thống mang phong cách thủ công gây ra sự chậm chễ và thiếu chính xác trong trao đổi thông tin. Trong thế giới máy tính và việc ký trong các văn bản điện tử, vấn đề ký như trong thực tế gặp phải nhiều khó khăn: các dòng thông tin trên máy tính có thể thay đổi dễ dàng, hình ảnh chữ ký tay của một người cũng có thể thay đổi từ văn bản này sang văn bản khác, việc thay đổi nội dung văn bản điện tử sau khi ký không để lại dấu vết,...Để có những đặc tính như trên, giao thức ký trong thế giới điện tử cần phải có sự hỗ trợ của công nghệ mã hóa. Giao thức cơ bản của chữ ký số dựa trên ý tưởng của Diffie và Hellman:

- Người gửi (chủ nhân của văn bản) ký văn bản bằng cách mã hóa nó với khóa bí mật của mình.
- Người gửi chuyển văn bản đã ký cho người nhận.
- Người nhận kiểm tra chữ ký bằng việc sử dụng chìa khóa công khai của người gửi để giải mã văn bản.

### Khái niệm 1.2.1. (Chữ ký số)

*Chữ ký số (khóa công khai) là mô hình sử dụng kỹ thuật mật mã để gắn với mỗi người sử dụng một cặp khóa công khai- bí mật và qua đó có thể ký văn bản điện tử cũng như trao đổi các thông tin mật. Quá trình sử dụng chữ ký số gồm hai quá trình: tạo chữ ký và kiểm tra chữ ký.*



Các đặc điểm chính của chữ ký số:

- Xác thực danh tính.
- Bảo mật dữ liệu.
- Chống chối bỏ nguồn gốc.
- Toàn vẹn dữ liệu.

Chữ ký số và chữ ký thông thường có nhiều điểm khác nhau:

1. Về tài liệu được ký: Với tài liệu thông thường, nó là một phần vật lý của tài liệu. Ngược lại, chữ ký số không phải theo kiểu vật lý gắn vào thông báo nên không thể nhìn thấy trên bức điện.
2. Về vấn đề kiểm tra chữ ký: Chữ ký thông thường được kiểm tra bằng cách so sánh nó với các chữ ký xác thực khác (chữ ký mẫu). Điểm yếu của chữ ký thông thường là không an toàn và dễ có thể giả mạo. Ngược lại, chữ ký số lại được kiểm tra nhờ dùng thuật toán kiểm tra công khai, bất kì ai có thể kiểm tra được.

**Định nghĩa 1.2.2.** (*Hệ thống chữ ký số điện tử*)

*Một sơ đồ chữ ký số gồm hai thành phần chủ chốt là thuật toán ký và thuật toán xác minh. Một sơ đồ chữ ký là một bộ năm  $(P, C, K, E, D)$  thỏa mãn các điều kiện:*

- $P$  là tập hữu hạn các bản rõ.
- $C$  là tập hữu hạn các chữ ký.
- $K$  là tập hữu hạn các khóa.
- $E$  là tập các thuật toán ký.
- $D$  là tập các thuật toán xác minh.

*Với mỗi  $k \in K$ , tồn tại một thuật toán ký  $sig_k$  thuộc  $E$  và một thuật toán xác minh  $ver_k$  thuộc  $D$ , trong đó  $sig_k : P \rightarrow C$  và  $ver_k : C \rightarrow \{True, False\}$  thỏa mãn với mọi  $x \in P$  và  $y \in C$ ,  $ver(x, y) = True$  nếu  $y = sig(x)$  và  $ver(x, y) = False$  nếu  $y \neq sig(x)$ .*

Khi một người dùng muốn kí lên thông báo  $x$  thì người đó dùng thuật toán an toàn để tạo chữ ký  $y = sig(x)$  nhận được và gửi cho người nhận. Người nhận sử dụng chữ ký  $sig(x)$  thì dùng thuật toán xác minh  $ver(x, y)$  để xác định tính đúng đắn của chữ kí (trả về  $True$  hoặc  $False$ ).

**Định nghĩa 1.2.3.** (Sơ đồ chữ ký số RSA)

Đặt  $n = pq$  trong đó  $p$  và  $q$  là hai số nguyên tố. Đặt  $P = C = Z_n$  và  $K = \{(n, p, q, a, b) : n = pq; p, q \text{ nguyên tố và } ab \equiv 1 \pmod{\phi(n)}\}$ . Với  $k \in K$ ,  $\text{sig}_k(x) = x^a \pmod n$ ,  $\text{ver}_k(x, y) = \text{True} \Leftrightarrow x \equiv y^b \pmod n$  với  $x, y \in Z_n$ , trong đó  $n, b$  là khóa công khai và  $p, q, a$  là khóa bí mật. Thông điệp  $x$  được ký theo đồng dư với khóa riêng với khóa riêng của người gửi và quá trình xác thực chữ ký dựa vào phép tính đồng dư nhưng với khóa công khai của người gửi.

Chữ ký RSA là một trong những loại chữ ký sử dụng phổ biến nhất hiện nay nó sử dụng những công cụ chính: số học, hàm băm mật mã học, mật mã khóa công khai. Những giao thức mã hóa đặc biệt là chữ ký số điện tử đều dựa trên lý thuyết số học để tạo khóa, mã hóa và giải mã. An toàn của những giao thức này đều liên quan tới vấn đề trong số học: giải thuật công khai và phân tích thừa số nguyên tố.

## 1.3 Lý thuyết hàm băm mật mã

Trong ngành mật mã học, một hàm băm mật mã học là một hàm băm với một số tính chất bảo mật nhất định để phù hợp việc sử dụng trong nhiều bảo mật thông tin đa dạng 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 hoặc chữ ký số. Các hàm băm nhận một chuỗi bit có độ dài tùy ý (hữu hạn) làm dữ liệu đầu vào và tạo ra một chuỗi bit có độ dài  $n$  bit gọi là mã băm. Ký hiệu  $D$  là miền xác định,  $R$  là miền giá trị của hàm băm  $h(x)$ .

$$h(x) : D \rightarrow R$$

**Định nghĩa 1.3.1.** (Hàm băm)

Hàm băm là một giải thuật nhằm sinh ra các giá trị băm tương ứng với mỗi khối dữ liệu. Giá trị băm đóng vai trò gần như một khóa để phân biệt các khối dữ liệu.

Hàm băm một chiều: với mọi mã băm biết trước, không thể tính toán để tìm được chuỗi bit ban đầu có mã băm bằng với mã băm đã cho.

Hàm băm kháng xung đột: không thể tính toán để tìm ra hai chuỗi bit có cùng giá trị băm.

Một số tính chất cơ bản của hàm băm:

- i) Có thể áp dụng đầu vào thông báo đầu vào với độ dài bất kỳ.
- ii) Tạo ra giá trị băm  $y = h(x)$  có độ dài cố định.
- iii)  $h(x)$  có thể tính được với bất kỳ  $x$  nào.
- iv) Tính một chiều: với mọi đầu ra  $y$  cho trước không tìm  $x'$  sao cho  $h(x') = y$  cho trước.

v) Tính chống xung đột yếu, với mọi đầu vào là  $x_1$  không thể tìm được giá trị  $x_2$  nào ( $x_2 \neq x_1$ ), sao cho  $h(x_2) = h(x_1)$ .

vi) Tính chống xung đột mạnh, không thể tính toán để tìm được hai dữ liệu đầu vào  $x_1, x_2$  phân biệt sao cho chúng có cùng giá trị băm  $h(x_1) = h(x_2)$ .

Như vậy, hàm băm một chiều thỏa mãn tính chất iv) và tính chất v), còn hàm băm kháng xung đột thỏa mãn tính chất iv) và tính chất vi).

Khối đầu vào  $x$  có chiều dài hữu hạn tùy ý sẽ được phân thành các khối con liên tiếp có chiều dài cố định  $r$ , giả sử là  $x_1, x_2, \dots, x_m$ . Tuy nhiên do chiều dài của khối dữ liệu ban đầu  $x$  là tùy ý, do đó cần phải thêm vào một số bit phụ sao cho tổng số bit của khối của khối dữ liệu  $x'$  sau khi thêm vào là bội số của  $r$ .

Tiếp theo, lần lượt cắt các khối con  $r$  bit từ khối mở rộng  $x'$ . Mỗi khối con  $r$  bit  $x_i$  lần lượt qua một hàm nén  $f$  của hàm băm  $h(x)$ . Tại bước thứ  $i$ , hàm nén  $f$  nhận dữ liệu đầu vào  $x_i$  và kết quả trung gian của bước trước đó (bước  $i - 1$ ) để tạo đầu ra là kết quả bước trung gian thứ  $i$ , kí hiệu là  $H_i$ . Kết quả trung gian của mỗi bước  $H_i$  là một chuỗi bit có độ dài cố định  $n > 0$ .

$$H_0 = a$$

$$H_i = f(H_{i-1}, x_i) (i = 1, 2, \dots, m)$$

$$h(x) = g(H_m).$$

Các tính chất của hàm băm là cần thiết cho chữ ký điện tử, bởi vì:

- Tính chất i) và tính chất ii) cần cho việc sinh một chữ ký hiệu quả.
- Tính chất iv), v) và vi) dùng để chống giả mạo chữ ký.

Một số hàm mật mã thông dụng: MD4, MD5 và SHA-1.

## Chương 2

# Xây dựng chương trình

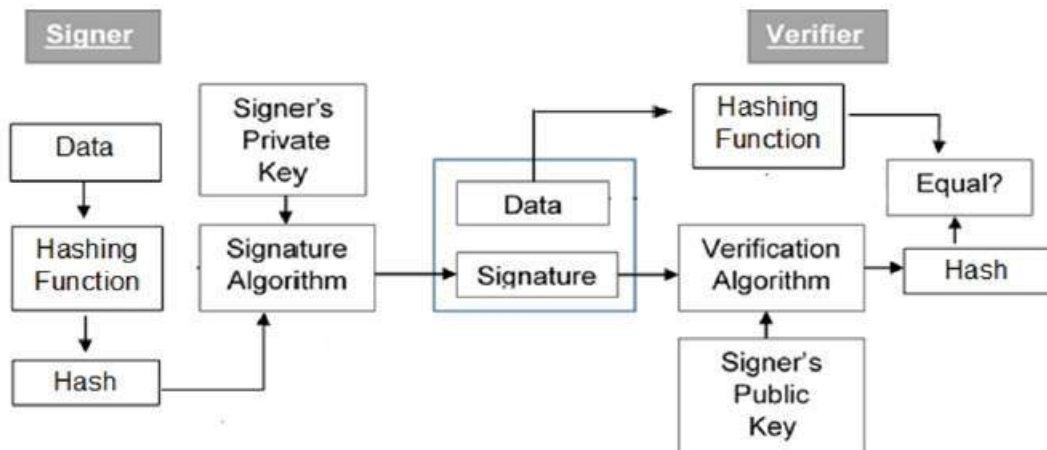
### 2.1 Chữ ký điện tử và thiết bị HSM

#### Các bước tạo ra chữ ký điện tử

- Bên gửi A thực hiện băm thông điệp cần gửi M bằng hàm băm H, rồi mã hóa giá trị vừa được băm này bằng mật mã khóa công khai với khóa bí mật của bên A, phần thông tin này chính là chữ ký xác thực của người dùng A với thông điệp M gửi đi.
- A gửi cho B văn bản và chữ ký. Thông điệp có thể được mã hoặc không mã hóa theo yêu cầu. Chữ ký có thể gắn liền với thông điệp rồi mã hóa bằng mật mã khóa đối xứng với khóa dùng chung K giữa A và B.
- B nhận được thông điệp (hoặc bản mã rồi giải mã với khóa chung K để lấy thông điệp) thì tiến hành 2 việc: băm giá trị thông điệp bằng hàm băm H, giải mã chữ ký bằng khóa công khai của bên A gửi rồi so sánh 2 giá trị vừa tính toán được. Nếu 2 giá trị này trùng khớp thì chứng tỏ thông điệp nhận được có nội dung không thay đổi so với phần ký.

#### Các thuật toán trong lược đồ chữ ký số

- Thuật toán sinh khóa. Thuật toán sinh ra cặp khóa bí mật/ công khai.
- Thuật toán ký. Đầu vào là một thông điệp, thông qua hàm băm mật mã, sinh ra một chữ ký số nhờ khóa bí mật.
- Thuật toán xác thực chữ ký số. Thuật toán tiến hành bởi bên thứ ba muốn kiểm tra tính đúng đắn của một chữ ký số. Thuật toán nhận đầu vào là thông điệp, chữ ký số của thông điệp đó và khóa công khai được cung cấp. Đầu ra của thuật toán là câu trả lời "đúng" hoặc "sai".



Hình 2.1: Sơ đồ chữ ký số điện tử.

### Thiết bị chữ ký số HSM

HSM (viết tắt của từ tiếng Anh: Hardware Security Module) là một thiết bị điện toán vật lý có chức năng quản trị và bảo vệ các cặp khóa chứng thư số cho các ứng dụng xác thực mạnh và xử lý mật mã. HSM thường được sản xuất dưới dạng một card PCI cắm vào máy tính hoặc một thiết bị độc lập có kết nối mạng.

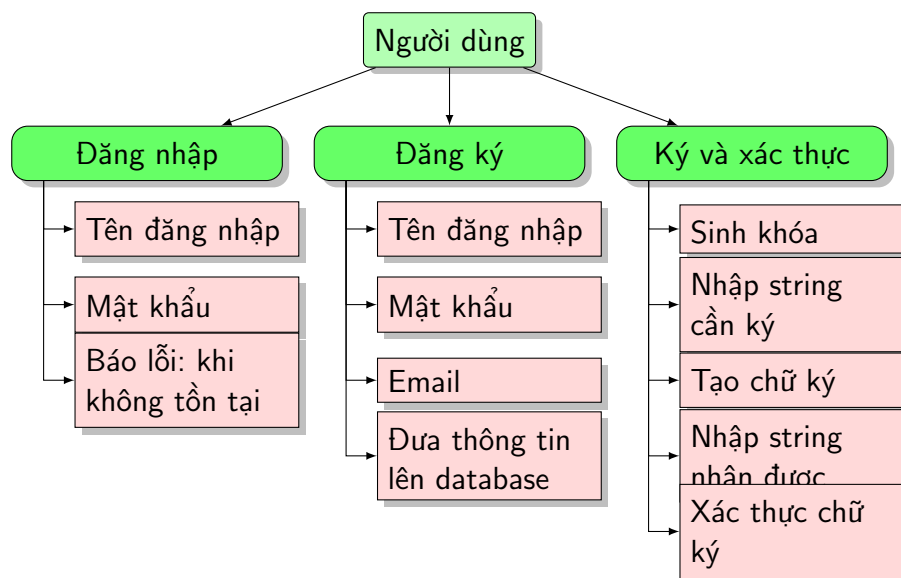
HSM có các tính năng chống can thiệp để lại các dấu hiệu có thể phát hiện được, các cảnh báo khi bị can thiệp, hoặc việc xâm nhập là khó khăn đến mức không thể không làm cho HSM ngừng hoạt động hoặc xóa cặp khóa khi bị phát hiện có can thiệp.

HSM hỗ trợ cả mật mã đối xứng và bất đối xứng (khóa công khai). Với một số ứng dụng như chứng thực số hay ký số, các cặp khóa bất đối xứng được dùng trong mật mã khóa công khai. Với các ứng dụng khác như mã hóa dữ liệu hay hệ thống thanh toán tài chính thì thường dùng cặp khóa đối xứng. Các chức năng của HSM:

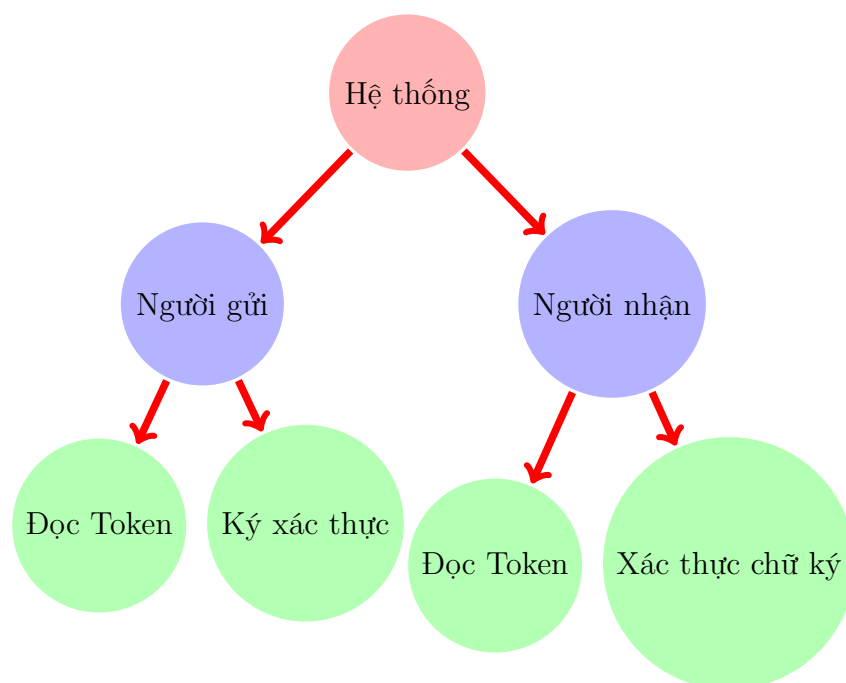
- Sinh khóa mật mã an toàn trên thiết bị.
- Lưu chứa khóa mật mã an toàn trên thiết bị.
- Quản lý khóa.
- Ký số và mã hóa.

## 2.2 Phân tích thiết kế hệ thống

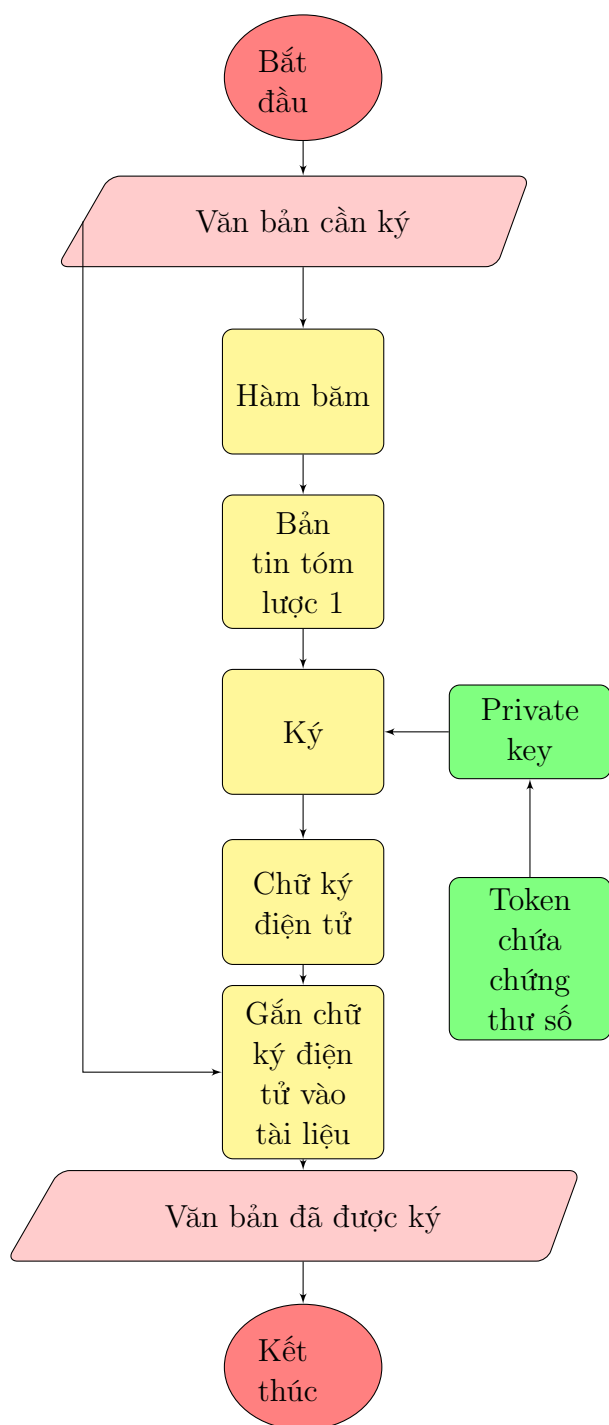
- Các chức năng của người dùng



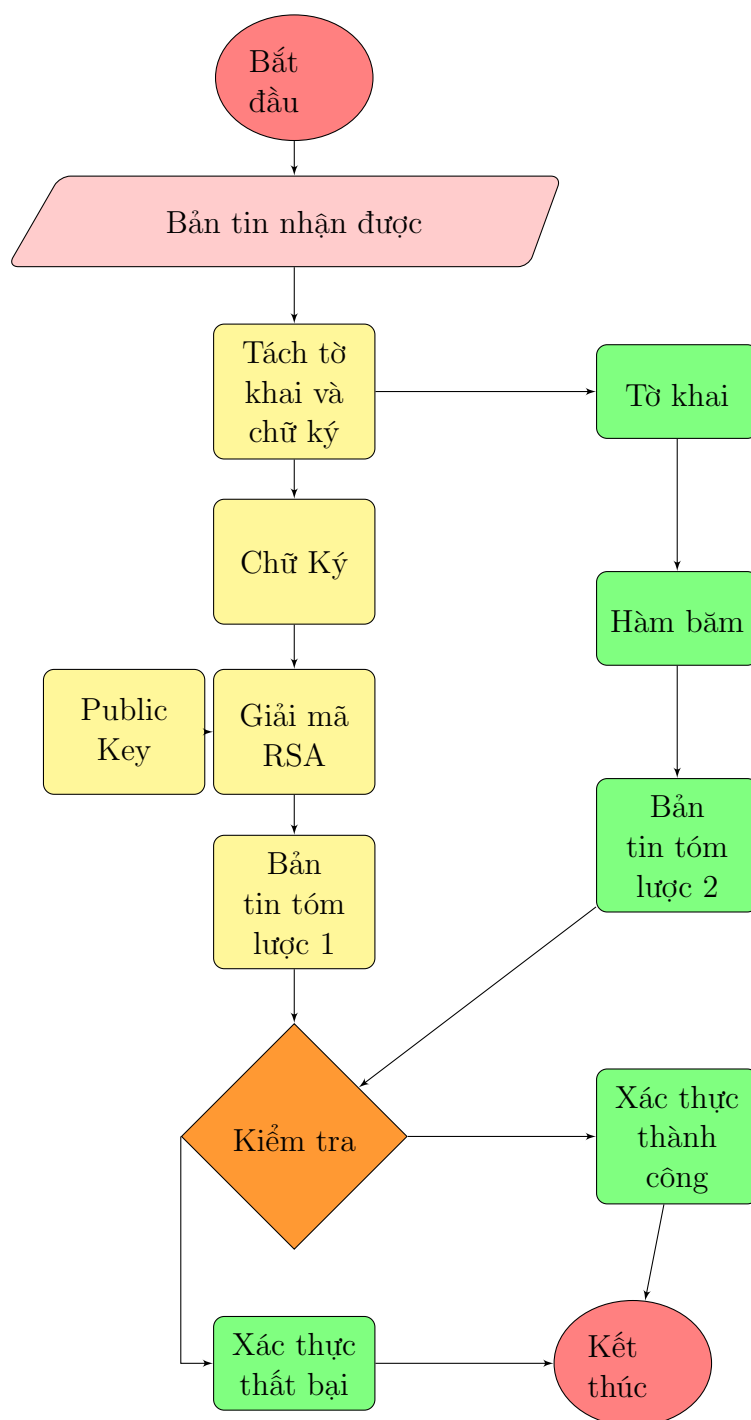
- Sơ đồ phân cấp chức năng cho hệ thống



- Ký xác thực



- Xác thực chữ ký số





## 2.3 Xây dựng chức năng

- Ngôn ngữ lập trình

```
1 Python 3.7
2 IDE PyCharm
```

- Cài đặt các thư viện

```
1 import rsa
2 import binascii
```

- Modul tạo khóa:

```
1 def generate_signature(key_size):
2     # key_size = {512, 1024, 2048, 4096}
3     # create the public key and private key
4     (public_key, private_key) = rsa.newkeys(key_size)
5     # write the public key to a file
6     with open('public_key.key', 'wb') as key_file:
7         key_file.write(public_key.save_pkcs1('PEM'))
8
9     # write the private key to a file
10    with open('private_key.key', 'wb') as key_file:
11        key_file.write(private_key.save_pkcs1('PEM'))
```

- Modul băm thông điệp và tạo chữ ký:

```
1 def hash_message(hash_type='SHA-512'):
2     # hash_type = {SHA-224, SHA-256, SHA-384, SHA-512}
3     # open private key file and load in key
4     private_Key = \
5     rsa.PrivateKey.load_pkcs1(file_open('private_key.key'))
6     #open the secret message file and return data variable
7     message = file_open('message')
8     hash_value = rsa.compute_hash(message, hash_type)
9     # sign the message with owners private key
10    signature = rsa.sign(message, private_Key, hash_type)
11    # save signature
12    s = open('signature_file', 'wb')
13    s.write(signature)
```

- Modul xác thực chữ ký số

```
1 def Verify_Signature():
2     # open public key file and load in key
3     public_Key =\
4     rsa.PublicKey.load_pkcs1(file_open('public_key.key'))
5     message = file_open('message')
6     signature = file_open('signature_file')
7     # verify the signature to show if successful or failed
8     try:
9         rsa.verify(message, signature, public_Key)
10        print("Signature successfully verified!")
11    except:
12        print("Warning!! signature could not be verified")
```

Các chức năng trên ngoài cách sử dụng thư viện có sẵn trong ngôn ngữ lập trình python là **rsa**, chúng ta cũng có thể tự triển khai các thuật toán để xây dựng chương trình. Độ phức tạp tính toán của thuật toán tùy thuộc vào cách chúng ta xây dựng chương trình: sinh cặp số nguyên tố lớn, các phép lũy thừa và modulo.

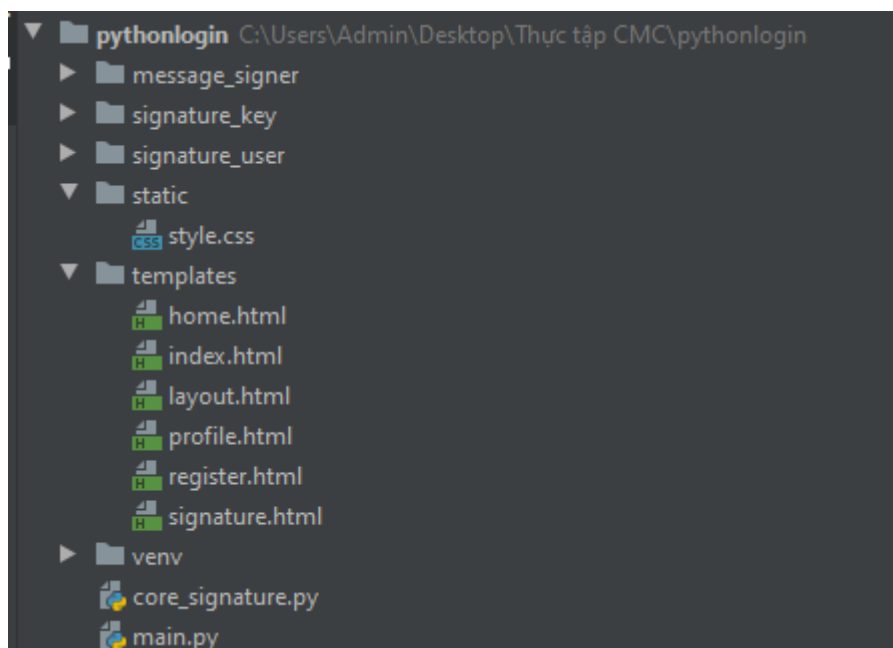
## Chương 3

# Cài đặt và kết quả thực nghiệm

### 3.1 Cài đặt core và API

Dựa trên phân tích về mặt chức năng, tôi đề xuất các công cụ sau để xây dựng chương trình

Công cụ	Chú thích	Pha sử dụng
Python	Ngôn ngữ lập trình	
Flask	Thư viện của python	API
RSA	Thư viện của python	CORE
HTML, CSS	Thiết kế giao diện	Giao diện của sản phẩm
MySQL	Hệ cơ sở dữ liệu	Lưu trữ thông tin người dùng



Hình 3.1: Cấu trúc file của chương trình.

### Thiết lập API

API đóng vai trò chạy ở phía sever, khi người dùng gửi lên một câu lệnh web, web gửi dữ liệu lên sever và sever gửi lại kết quả cho người dùng. Ở trong bài báo cáo này, tôi chọn Flask làm back-end cho sản phẩm.

- Định tuyến, kết nối với cơ sở dữ liệu MySQL

```
from flask import Flask, render_template, request, redirect, url_for, session
from flask_mysql import MySQL
import MySQLdb.cursors
import re
import core_signature
app = Flask(__name__)
# Change this to your secret key (can be anything, it's for extra protection)
app.secret_key = 'your secret key'
# Enter your database connection details below
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = 'minhquan123987'
app.config['MYSQL_DB'] = 'pythonlogin'
# Intialize MySQL
mysql = MySQL(app)
```

Hình 3.2: Cài đặt thư viện và kết nối với cơ sở dữ liệu.

- Các chức năng của người dùng: đăng nhập, đăng ký, đăng xuất, hồ sơ người dùng và trang chủ người dùng.

```
# http://localhost:5000/pythonlogin/ - this will be the login page, we need to use both GET and POST requests
@app.route('/pythonlogin/', methods=['GET', 'POST'])
def login():...
# http://localhost:5000/python/logout - this will be the logout page
@app.route('/pythonlogin/logout')
def logout():...
# http://localhost:5000/pythonlogin/register - this will be the registration page, we need to use both GET and POST
# requests
@app.route('/pythonlogin/register', methods=['GET', 'POST'])
def register():...

# http://localhost:5000/pythonlogin/home - this will be the home page, only accessible for loggedin users
@app.route('/pythonlogin/home')
def home():...

# http://localhost:5000/pythonlogin/profile - this will be the profile page, only accessible for loggedin users
@app.route('/pythonlogin/profile')
def profile():...
```

Hình 3.3: Các chức năng cơ bản của người dùng.

- Xây dựng chức năng tạo và xác thực chữ ký.

```
1 @app.route('/pythonlogin/signature')
2 def signature():
3     return render_template("signature.html",\
4         public_key="signature public key",\
5         private_key="signature private key",\
6         show_form1=True, show_form2=True, \
7         show_form3=True)
8
9
10 @app.route('/pythonlogin/signature/generate-key',\
11     methods=['POST'])
12 def generate_key_signature():
13     option = request.form['options']
14     if option == "512":
15         a, b = \
16             core_signature.signature_generate_key(512)
17     elif option == "1024":
18         a, b = \
19             core_signature.signature_generate_key(1024)
20     elif option == "2048":
21         a, b = \
22             core_signature.signature_generate_key(2048)
23     elif option == "4096":
24         a, b = \
```

```

25         core_signature.signature_generate_key(4096)
26     return render_template("signature.html", \
27         public_key=a, private_key=b, \
28         show_form1=True, show_form2=True, \
29         show_form3=True)
30
31
32 @app.route('/pythonlogin/signature/signature-message', \
33 methods=['POST'])
34 def message_signature():
35     message_send = request.form['message_send']
36     option = request.form['options']
37     if option == "SHA-256":
38         signature_message = \
39 core_signature.signature_message_hash(message_send, 'SHA-256')
40     elif option == "SHA-384":
41         signature_message = \
42 core_signature.signature_message_hash(message_send, 'SHA-384')
43     elif option == "SHA-512":
44         signature_message = \
45 core_signature.signature_message_hash(message_send, 'SHA-512')
46     return render_template("signature.html", \
47         signature_message=signature_message, \
48         show_form1=False, \
49         show_form2=True, \
50         show_form3=True)
51
52
53 @app.route('/pythonlogin/signature/signature-verify', \
54 methods=['POST'])
55 def verify_signature():
56     message_receive = request.form['message_receive']
57     signature_verify = \
58 core_signature.signature_verify(message_receive)
59     return render_template("signature.html", \
60         signature_verify=signature_verify, \
61         show_form1=False, show_form2=False, \
62         show_form3=False)
63
64
65 if __name__ == '__main__':
66     app.run(host='localhost', port=5000, debug=True)

```

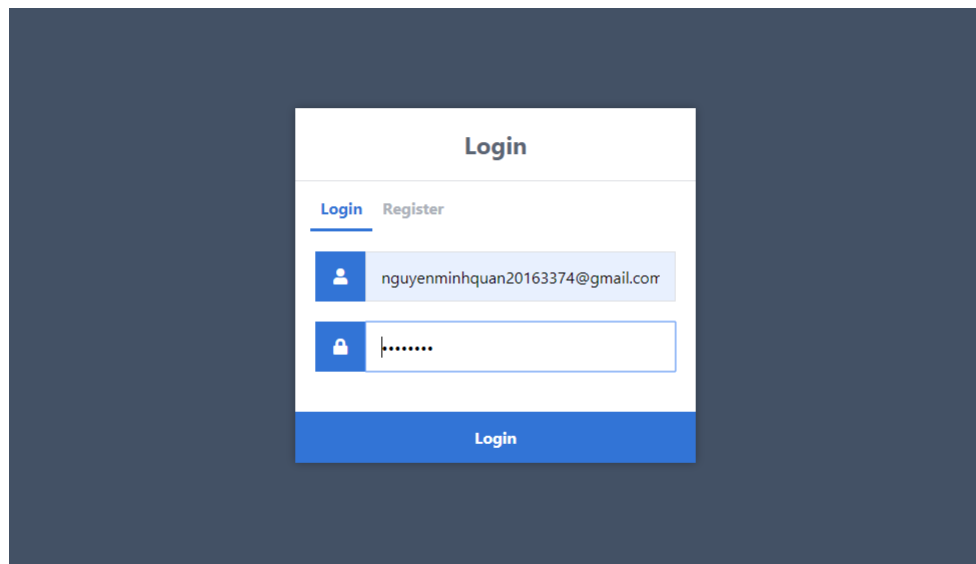
## 3.2 Xây dựng giao diện

Giao diện là nơi tương tác trực tiếp với người sử dụng, đóng vai trò trung gian giữa người dùng và sever. Giao diện được thiết kế đơn giản bằng HTML5 và CSS3.

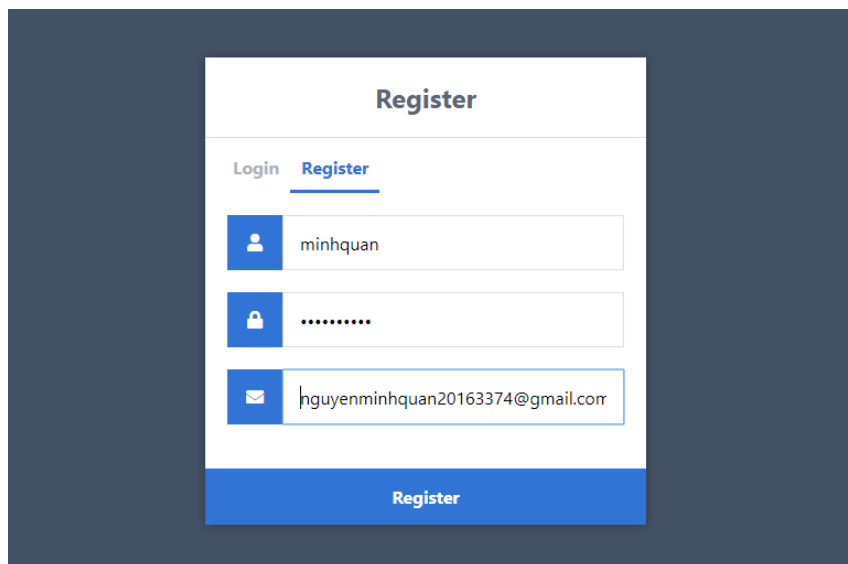
Tên file	Chức năng
index.html	Trang đăng nhập vào ứng dụng
register.html	Trang đăng ký tài khoản
home.html	Trang chủ của người dùng
profile.html	Trang hồ sơ người dùng
layout.html	Trang tiêu đề
signature.html	Trang thực hiện việc ký số và xác thực ký số

Các button trong giao diện và chức năng được mô tả trong bảng dưới đây.

Tên các button	Chức năng
Home	Trở về trang chủ
Profile	Đến trang hồ sơ người dùng
Signature	Đến trang thực hiện chữ ký số
Logout	Đăng xuất trở về trang login
Generate Key	Sinh khóa bí mật và khóa công khai
Create Signature	Tạo chữ ký số với văn bản gửi đi tương ứng
Verify Signature	Xác thực chữ ký số kèm theo văn bản nhận được

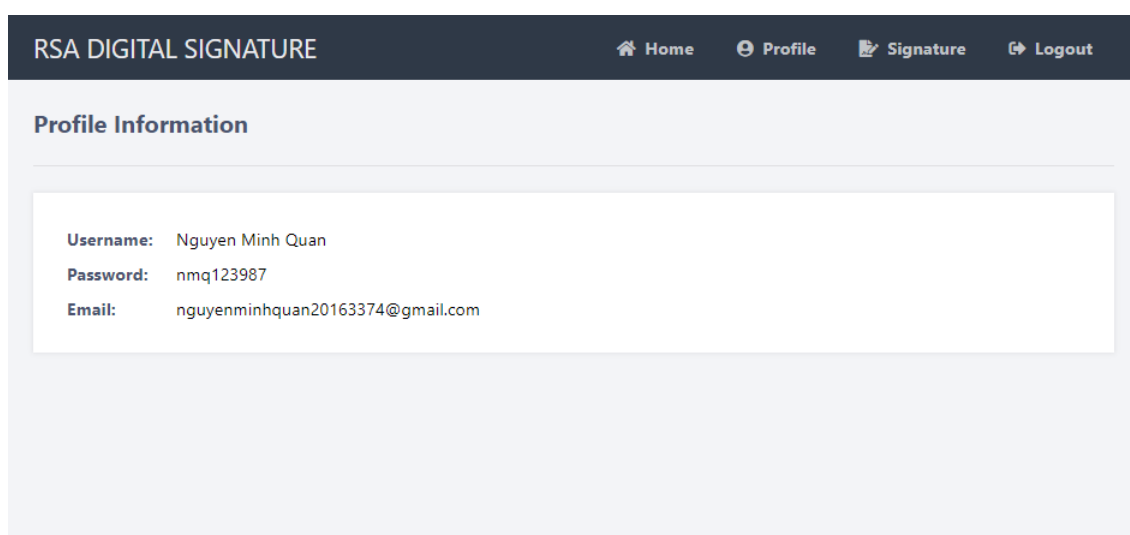


Hình 3.4: Giao diện đăng nhập người dùng của ứng dụng.



The image shows a 'Register' form within a dark blue container. The form has a white background and a blue header with the title 'Register'. Below the header, there are two tabs: 'Login' and 'Register', with 'Register' being the active tab. The form contains three input fields: a username field with the value 'minhquan', a password field with masked characters '.....', and an email field with the value 'nguyenminhquan20163374@gmail.com'. Each input field has a blue icon on the left (person, lock, and envelope respectively). At the bottom of the form is a blue button labeled 'Register'.

Hình 3.5: Giao diện đăng ký người dùng của ứng dụng.



The image shows the 'Profile Information' page of the 'RSA DIGITAL SIGNATURE' application. The page has a dark blue header with the application name and navigation links: 'Home', 'Profile', 'Signature', and 'Logout'. The 'Profile Information' section is highlighted in light blue. Below this section, there is a white box containing the user's profile details: 'Username: Nguyen Minh Quan', 'Password: nmq123987', and 'Email: nguyenminhquan20163374@gmail.com'.

Hình 3.6: Giao diện chính của chương trình.



**Generate RSA Key Size**

☐ 512 bit ☐ 1024 bit ☐ 2048 bit ☐ 4096 bit

**Public Key:**

```
b'-----BEGIN RSA PUBLIC KEY-----
\nMIGAoGBAlt4AnhWgijVKITksBjYju9NBXmDVOX3/9Qt4aozy5pm
kSqLU9z3ZIMU\ynsr5x2yLyVfufqbWPHbjpgpr1mwUAB5xXJF9F5WV3
ndqvNhThDtl+oeaoBPHOBevR\4QzhkAVMv7FG7tqST+LG98CEQ
MKVqJJeNL9UiwDt5+AweAUjotpAgMBAAE=\n-----END RSA
```

**Private Key:**

```
b'-----BEGIN RSA PRIVATE KEY-----
\nMIIICyAIBAAKBgQCLeAJ4Vol1Sok5LAY2I7VTQV5g1TI9/5fULeGqM
8uaZpEqVPc\n92SDFLK+cdsi8IX7n6m1jx246YKa9ZsFAAecVYRfReV
ld53arzYU4Q7S/qHmqATx\onzgXr0eECc4ZAFtL+xRu7akk/ixvfAhEDC
laqXjS/VIsA7efgMHgFi6LaQIDAQAB\nAoGAe6yQ7CsyTZVp9+O9Ilc
```

**Generate Key**

Hình 3.7: Sinh cặp khóa bí mật, khóa công khai với độ dài tùy chọn.

**Hash Function**

☐ SHA-256 ☒ SHA-384 ☐ SHA-512

**Message Send:**

Viện Toán Ứng dụng và Tin học Đại Học Bách Khoa Hà Nội

**Provide Signature Value:**

```
b'=\x19\xfc\x15\xa7\xd0N\n\xc5G\xa0\xa1\x8f\x15\xe4\xa2\xf2\x80\x
3\x93\xc4\xb1\xc6\x12\x13\xdf\x90\x90\x85,P\xad\xec3\xdc\x9d\xee\x
1a\xa0\xd3't\xa3\x0e\xc7\x08\xdc2\xdb\x96\xe3\xed\x8a<\xb1\x5ak.\
xac\x9e\x85=\xde\xfb\xbc\xbb\x7\xd7=\\to/\x1a\x95\xa9\x8fm\xfb\x
6f\xca)\f\x90\x8em\x92^xe1\x9fU\x80\xc5\x9b\x9f\x92\xbe\xa6\xaf\x
```

**Create Signature**

Hình 3.8: Tạo chữ ký số với văn bản tương ứng.

**Message Receive:**

Viện Toán Ứng dụng và Tin học Đại Học Bách Khoa Hà Nội

**Signature Verify:**

**Verify Signature**

Activate Windows  
Go to Settings to activate Windows

Hình 3.9: Xác thực chữ ký số với văn bản tương ứng nhận được.

### 3.3 Đánh giá và kết luận

#### Kết quả đạt được

- Tìm hiểu các thuật toán trong mật mã khóa công khai RSA.
- Tìm hiểu chữ ký số, thiết bị HSM.
- Xây dựng được chương trình thực hiện được các chức năng cơ bản.

#### Hạn chế

- Hệ thống còn đơn giản, tính bảo mật chưa cao.
- Các chức năng còn hạn chế, chưa có các chức năng tùy chọn ký như là ký PDF hay ký WORD.
- Giao diện hệ thống đơn giản, chưa được tối ưu.

#### Định hướng phát triển

- Bổ sung nhiều tính năng tùy chọn.
- Giao diện dễ sử dụng, tùy biến cao.
- Phát triển trên nền tảng di động, ký mobile.
- Tương tác giữa phần mềm với phần cứng USB token, giả lập HSM,...

#### Tài liệu và source code của bài báo cáo

- github: <https://github.com/minhquan27/RSAsignaturesimple>.
- gmail: [nguyenminhquan20163374@gmail.com](mailto:nguyenminhquan20163374@gmail.com).

# Kết luận

Như vậy, bài báo cáo đã giới thiệu một số các khái niệm về hệ mã RSA, chữ ký số, hàm băm và quá trình hình thành nên chữ ký điện tử. Bài báo cáo đã trình bày về lý thuyết về chữ ký số và xây dựng một ứng dụng đơn giản để tạo và xác thực chữ ký số. Có rất nhiều cách cải tiến ứng dụng trên như: xây dựng ký số cho file PDF, xây dựng ký số trên nền tảng các thiết bị di động,... Trong quá trình thực hiện, vì thời gian và kiến thức có hạn, nên bài báo cáo chỉ xây dựng được một chương trình có thể kiểm tra và hiểu được lý thuyết, song rất khó ứng dụng vào thực tiễn. Trong các bài báo cáo tiếp theo, em hi vọng sẽ hoàn thiện, xây dựng và bổ sung các chức năng hơn cho chương trình, cũng như các đọc tài liệu, các thư viện trong bảo mật để xây dựng một chương trình có tính hoàn thiện và ứng dụng cao hơn trong thực tế. Một lần nữa, em xin cảm ơn thầy giáo đã có những nhận xét sâu sắc để bài báo cáo của em được hoàn thiện hơn.

# Tài liệu tham khảo

- [1] *"Giáo trình An toàn và Bảo mật thông tin"*, Đại học Bách Khoa Hà Nội.
- [2] Bài giảng môn học "Mật mã và độ phức tạp tính toán", Viện Toán Ứng dụng và Tin học, Đại học Bách Khoa Hà Nội.
- [3] Các nguồn tham khảo khác trên internet.