

TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT  
KHOA CÔNG NGHỆ THÔNG TIN  
BỘ MÔN HỆ THỐNG THÔNG TIN VÀ TRI THỨC

BÁO CÁO SINH HOẠT HỌC THUẬT

## TÌM HIỂU THUẬT TOÁN RSA VÀ ỨNG DỤNG

Người thực hiện : Dương Chí Thiện

Bộ môn Hệ thống thông tin và Tri thức

HÀ NỘI, 5/2024

## MỤC LỤC

<b>MỤC LỤC .....</b>	<b>i</b>
<b>DANH MỤC HÌNH ẢNH .....</b>	<b>iii</b>
<b>MỞ ĐẦU .....</b>	<b>1</b>
<b>CHƯƠNG 1: TỔNG QUAN.....</b>	<b>2</b>
<b>1.1. Tổng quan về thông tin .....</b>	<b>2</b>
<b>1.1.1. Khái niệm về thông tin .....</b>	<b>2</b>
1.1.2. Sự khác biệt giữa thông tin và dữ liệu .....	2
1.2. Hệ thống thông tin và vấn đề an ninh.....	2
<b>1.3. Khái quát về an toàn thông tin.....</b>	<b>3</b>
<b>1.3.1. An toàn thông tin là gì? .....</b>	<b>3</b>
<b>1.3.2. Các thành phần của an toàn thông tin.....</b>	<b>3</b>
<b>1.3.3. Tầm quan trọng của an toàn thông tin .....</b>	<b>4</b>
<b>1.3.4. Mục tiêu của an toàn thông tin.....</b>	<b>4</b>
<b>1.3.5. Các yêu cầu của việc trao đổi thông tin và nguyên tắc cơ bản của an toàn thông tin .....</b>	<b>5</b>
<b>1.3.6. Các nội dung chính của an toàn thông tin.....</b>	<b>5</b>
<b>1.4. Các công cụ bảo đảm an toàn thông tin .....</b>	<b>5</b>
<b>1.4.1. Mật mã .....</b>	<b>5</b>
<b>1.4.2. Giấu tin .....</b>	<b>6</b>
<b>1.4.3 Các bài toán thông dụng trong an toàn thông tin .....</b>	<b>6</b>
<b>CHƯƠNG 2: THUẬT TOÁN RSA .....</b>	<b>7</b>
<b>2.1 Lý do, mục đích chọn đề tài.....</b>	<b>7</b>
<b>2.2 Tổng quan về thuật toán RSA .....</b>	<b>8</b>
<b>2.3 Cài đặt thuật toán RSA .....</b>	<b>10</b>

<b>2.4 Cải tiến.....</b>	<b>13</b>
<b>CHƯƠNG 3: KẾT LUẬN VÀ ĐỊNH HƯỚNG.....</b>	<b>23</b>
<b>3.1 Kết luận .....</b>	<b>23</b>
<b>3.2 Định hướng.....</b>	<b>23</b>
<b>Tài Liệu Tham Khảo.....</b>	<b>25</b>

## DANH MỤC HÌNH ẢNH

Hình 2.1. Giao diện quản lý .....	177
Hình 2.2. Giao diện để xóa.....	188
Hình 2.3. Giao diện thêm .....	188
Hình 2.4. Giao diện sửa.....	1919
Hình 2.5. Chức năng tìm kiếm .....	19
Hình 2.6. Giao diện tùy chỉnh hiển thị cột .....	200
Hình 2.7. Giao diện lọc theo giới tính và nơi cấp .....	20
Hình 2.8. Giao diện đăng nhập H2sign .....	21
Hình 2.9. Giao diện để tạo file gửi.....	21
Hình 2.10. Giao diện để giải mã và xác nhận chữ ký .....	22

## MỞ 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, 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, đặc 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.

Theo thống kê của Cục An ninh mạng và Phòng chống tội phạm sử dụng công nghệ cao, chỉ tính riêng trong năm 2021 đã có hơn 76.977 vụ tấn công tại một số hệ thống mạng trọng yếu được thực hiện theo hình thức khai thác lỗ hổng. Có khoảng 14 nghìn vụ tấn công dò quét mạng, hơn 12 nghìn vụ tấn công có chủ đích (APT), hơn 7.300 vụ tấn công xác thực, gần 7 nghìn vụ tấn công mã độc và khoảng 650 vụ tấn công từ chối dịch vụ,...

Do đó nhu cầu về an toàn và bảo mật thông tin ngày càng quan trọng và cấp thiết hơn bao giờ hết. Vì vậy tôi đã lựa chọn đề tài nghiên cứu khoa học về lĩnh vực này với tên đề tài “Tìm hiểu Thuật toán RSA và ứng dụng”. Bằng cách nghiên cứu và ứng dụng hai công nghệ nổi tiếng là RSA (Rivest-Shamir-Adleman) và chữ ký RSA vào việc bảo vệ thông tin.

Đề tài nghiên cứu gồm 3 chương:

Chương 1: Tổng quan

Chương 2: Thuật toán RSA và chữ ký điện tử

Chương 3: Kết luận và định hướng

## CHƯƠNG 1: TỔNG QUAN

### 1.1. Tổng quan về thông tin

#### 1.1.1. Khái niệm về thông tin

Thông tin là một khái niệm trừu tượng mô tả những gì đem lại hiểu biết, nhận thức cho con người cũng như các sinh vật khác.

Thông tin có những đặc điểm sau:

- Tồn tại khách quan
- Có thể tạo ra, truyền ra, lưu trữ, chọn lọc
- Có thể bị sai lệch do nhiều tác động

#### 1.1.2. Sự khác biệt giữa thông tin và dữ liệu

Hai thuật ngữ dữ liệu và thông tin liên quan chặt chẽ với nhau, trên thực tế chúng thường được sử dụng thay thế cho nhau. Tuy nhiên, giữa chúng cũng có sự khác biệt và chúng ta nên phân biệt giữa dữ liệu và thông tin.

Chúng ta quan niệm thông tin là những gì được tạo nên nhằm giảm sự không xác định. Nói khác, chúng ta có thể hiểu thông tin như là dữ liệu đã được xử lý theo cách mà chúng ta có thể làm tăng hàm lượng tri thức cho người sử dụng nó, hay nói cách khác là thông tin dữ liệu được đặt trong ngữ cảnh với một hình thức thích hợp và có lợi cho người sử dụng

### 1.2. Hệ thống thông tin và vấn đề an ninh

Có ba hình thức chủ yếu đe dọa đối với hệ thống:

1. Phá hoại: Kẻ thù phá hỏng thiết bị phần cứng hoặc phần mềm hoạt động trên hệ thống.
2. Sửa đổi: Tài sản của hệ thống bị sửa đổi trái phép. Điều này thường làm cho hệ thống không làm đúng chức năng của nó. Chẳng hạn như thay đổi mật khẩu, quyền người dùng trong hệ thống, làm họ không thể truy cập vào hệ thống việc làm
3. Can thiệp: Tài sản bị truy cập bởi những người không có thẩm quyền. Các truyền thông thực hiện trên hệ thống bị ngăn chặn, sửa đổi.

Có ba biện pháp ngăn chặn phổ biến:

1. Điều khiển thông tin qua phần mềm: dựa vào các cơ chế an toàn bảo mật của hệ thống nền (hệ điều hành), các thuật toán mật mã học.
2. Điều khiển qua phần cứng: các cơ chế bảo mật, các thuật toán mật mã học được cứng hóa để sử dụng
3. Điều khiển thông qua các chính sách của tổ chức: ban hành các quy định của tổ chức nhằm đảm bảo tính an toàn bảo mật của hệ thống.

An toàn hệ thống thông tin có liên quan mật thiết với các lĩnh vực an toàn máy tính (computer security), an toàn mạng (network security), đảm bảo tính chính xác của thông tin (information assurance). Tất cả lĩnh vực này đều có chung một mục tiêu là phải đảm bảo được 3 đặc tính chính, đó là tính riêng tư (confidentiality), tính toàn vẹn (integrity), và tính sẵn sàng (availability). Ba đặc tính đó được coi là các đặc tính mang bản chất an toàn thông tin.

### **1.3. Khái quát về an toàn thông tin**

#### **1.3.1. An toàn thông tin là gì?**

An toàn thông tin là một khái niệm liên quan đến việc bảo vệ tài sản thông tin. Đó là một lĩnh vực rộng lớn, bao gồm cả sản phẩm và những quy trình nhằm ngăn chặn truy cập trái phép, hiệu chỉnh, xóa thông tin,...

Hiểu một cách đơn giản, an toàn thông tin là thuật ngữ dùng để chỉ việc bảo vệ thông tin số và các hệ thống tin để chống lại các nguy cơ can thiệp có mục đích như các hành động truy cập, sử dụng, phát tán, phá hoại, sửa đổi và phá hủy bất hợp pháp nhằm đảm bảo cho thông tin và hệ thống thông tin thực hiện đúng chức năng, phục vụ đúng đối tượng.

#### **1.3.2. Các thành phần của an toàn thông tin**

An toàn thông tin có thể được chia thành ba thành phần chính: an toàn máy tính và dữ liệu (Computer & data security), an ninh mạng (Network security) và quản lý an toàn thông tin (Management of information security). Ba thành phần của an toàn thông tin có quan hệ mật thiết và giao thoa với nhau, trong đó phần chung của cả ba thành phần trên là chính sách an toàn thông tin (Policy)

### 1.3.3. Tầm quan trọng của an toàn thông tin

Như chúng ta đều biết, đối với nhiều tổ chức, doanh nghiệp, cá nhân thì thông tin và dữ liệu đóng một vai trò hết sức quan trọng trong đời sống và có khi ảnh hưởng tới sự tồn vong của họ. Vì vậy, việc bảo mật những thông tin và dữ liệu đó là điều vô cùng cần thiết, nhất là trong bối cảnh hiện nay các hệ thống thông tin ngày càng được mở rộng và trở nên phức tạp dẫn đến tiềm ẩn nhiều nguy cơ không lường trước được.

Mặt khác, tính chất, mức độ, và phạm vi của các cuộc tấn công vào hệ thống máy tính và mạng ngày càng gia tăng bởi chưa bao giờ việc tiếp cận với các kỹ thuật và sử dụng các công cụ tấn công lại trở nên dễ dàng và đơn giản hơn thế. Và cuối cùng, xuất phát từ động cơ kiếm lợi hoặc chính trị mà các tổ chức tài chính và cơ quan chính phủ đang đang trở thành mục tiêu chính của các hacker.

Tất cả những điều này cho thấy vai trò cốt yếu của các chuyên gia an toàn thông tin trong cuộc chiến bảo mật đầy khốc liệt và không có hồi kết này. Nhưng các bạn khoan vội nghĩ ngay tới các công nghệ hay cách thức để đảm bảo an toàn cho hệ thống thông tin.

### 1.3.4. Mục tiêu của an toàn thông tin

Vấn đề an toàn thông tin được đặt ra nhằm 4 mục tiêu chính:

1. Bảo đảm tính bí mật (bảo mật): Thông tin không được phép cung cấp hay tiết lộ cho bất kỳ cá nhân, tổ chức, hay quy trình trái phép.
2. Bảo đảm tính toàn vẹn (bảo toàn): Ngăn chặn hay hạn chế việc bổ sung, loại bỏ và sửa dữ liệu không được phép, tức là bảo đảm tính chính xác và đầy đủ của tài sản thông tin.
3. Bảo đảm tính xác thực (chứng thực): Xác thực đúng thực thể cần kết nối giao dịch, xác thực đúng thực thể có trách nhiệm về nội dung thông tin (xác thực nguồn gốc thông tin).
4. Bảo đảm tính sẵn sàng: Thông tin được tiếp cận và sử dụng một cách dễ dàng theo nhu cầu của người/tổ chức được phép.



### **1.3.5. Các yêu cầu của việc trao đổi thông tin và nguyên tắc cơ bản của an toàn thông tin**

Do giao dịch mở trong môi trường Internet, giao dịch xuyên quốc gia nên việc trao đổi thông tin đã đặt ra những yêu cầu (còn được xem là những nguyên lý hoạt động cơ bản của vấn đề bảo mật thông tin) sau:

1. Tính bí mật/riêng tư (confidentiality/privacy)
2. Tính toàn vẹn (integrity)
3. Tính xác thực (authenticity)
4. Tính không thể chối bỏ (non repudiation)
5. Tính nhận dạng (identification)

### **1.3.6. Các nội dung chính của an toàn thông tin**

Để đảm bảo ATTT bên trong máy tính hay đường truyền tin, chúng ta quan tâm cả hai phương diện về An toàn máy tính và An toàn truyền tin:

1. An toàn máy tính (computer security): Là sự bảo vệ các thông tin cố định bên trong máy tính (thông tin tĩnh). Đó là khoa học về đảm bảo an toàn thông tin trong máy tính
2. An toàn truyền tin (communication security): Là sự bảo vệ thông tin lên đường truyền tin (thông tin động). Đó là khoa học về đảm bảo an toàn thông tin trên đường truyền tin

## **1.4. Các công cụ bảo đảm an toàn thông tin**

### **1.4.1. Mật mã**

Mật mã là một ngành khoa học nghiên cứu cách viết và giữ thông tin một cách bí mật. Nói một cách khái quát, đó là phương thức đảm bảo sự bí mật cho thông tin tại nơi lưu trữ cũng như khi truyền thông tin trên mạng, cho dù sự trao đổi này diễn ra trên môi trường truyền thông không an toàn.

### 1.4.2. Giấu tin

Trong lĩnh vực an toàn và bảo mật hệ thống thông tin, khái niệm “giấu tin” đề cập đến việc bảo vệ thông tin bằng cách ẩn nó trong một đối tượng dữ liệu khác mà không làm thay đổi cảm nhận thông thường của đối tượng đó. Điều này giúp ngăn chặn việc truy cập hoặc phát hiện thông tin một cách không mong muốn. Kỹ thuật giấu tin bao gồm hai phương pháp chính:

1. **Ẩn mã (Steganography):** Nhúng thông tin vào trong một đối tượng như hình ảnh, âm thanh hoặc video mà không làm thay đổi cảm nhận về đối tượng đó. Điều này giúp thông tin được bảo mật và khó bị phát hiện bởi người không có quyền truy cập.
2. **Thủy văn số (Digital Watermarking):** Đánh dấu thông tin vào trong một đối tượng để xác thực hoặc bảo vệ bản quyền, thường được sử dụng trong các tệp đa phương tiện.

### 1.4.3 Các bài toán thông dụng trong an toàn thông tin

#### 1.4.3.1. Bài toán bảo mật thông tin

Các bài toán bảo mật thông tin cơ bản gồm:

1. Nén thông tin
2. Mã hóa thông tin
3. Giấu thông tin

#### 1.4.3.2. Bài toán bảo toàn thông tin

Bảo toàn thông tin hay đảm bảo tính toàn vẹn của thông tin: Người ngoài cuộc khó thể thay đổi được thông tin. Toàn vẹn thông tin là đảm bảo thông tin là đảm bảo thông tin không bị thay đổi xuyên tạc bởi những kẻ không có thẩm quyền.

Có một số cách bảo toàn thông tin sau:

1. Không cho phép sửa đổi thông tin. Bảo toàn thông tin bằng các kỹ thuật: nén thông tin, mã hóa thông tin, giấu tin.
2. Dùng hàm băm tạo đại diện bản tin
3. Dùng chữ kí số
4. Dùng thủy văn kí

## CHƯƠNG 2: THUẬT TOÁN RSA

### 2.1 Lý do, mục đích chọn đề tài

Khi nghiên cứu về bảo mật thông tin, việc chọn một thuật toán mã hóa phù hợp là một quyết định quan trọng. Trong đề tài nghiên cứu khoa học này, nhóm em sẽ trình bày lý do chọn thuật toán RSA, một trong những thuật toán mã hóa và giải mã khóa công khai phổ biến nhất.

1. **Bảo mật cao:** RSA dựa trên việc giải bài toán phân tích số nguyên tố thành các thừa số nguyên tố, một vấn đề tính toán rất phức tạp. Hiện nay, không có thuật toán hiệu quả nào để giải quyết bài toán này trong thời gian đa thức, giúp cho RSA có một mức độ bảo mật cao.
2. **Sự phổ biến :** RSA là một trong những thuật toán mã hóa và giải mã khóa công khai được sử dụng rộng rãi nhất và được hỗ trợ trong nhiều thư viện và nền tảng phát triển phần mềm khác nhau. Điều này đảm bảo rằng việc triển khai và tích hợp RSA vào các hệ thống và ứng dụng là dễ dàng và thuận tiện.
3. **Hiệu suất tốt:** Mặc dù RSA không phải là thuật toán nhanh nhất, nhưng hiệu suất của nó vẫn khá tốt đối với hầu hết các ứng dụng. Đặc biệt là với các kích thước khóa lớn, RSA vẫn duy trì được mức độ bảo mật cao mà không ảnh hưởng quá nhiều đến hiệu suất.
4. **Độc lập về số nguyên tố:** Trong RSA, không cần phải chia sẻ thông tin về số nguyên tố cụ thể nào đã được chọn. Điều này tạo ra một mô hình độc lập và linh hoạt, giúp cho việc triển khai và quản lý hệ thống RSA trở nên đơn giản và tiện lợi.
5. **Tính một chiều:** RSA là một thuật toán một chiều, có nghĩa là quá trình mã hóa dễ dàng nhưng việc giải mã ngược lại là khó khăn. Điều này giúp bảo vệ thông tin của người gửi và chỉ cho phép người nhận được khóa bí mật mới có thể giải mã được dữ liệu.

Với các ưu điểm trên, việc chọn thuật toán RSA mang lại sự bảo mật cao, tính phổ biến, hiệu suất tốt.

## 2.2 Tổng quan về thuật toán RSA

Thuật toán RSA là một thuật toán mã hóa và giải mã khóa công khai được phát triển bởi ba nhà toán học là Ronald Rivest, Adi Shamir và Leonard Adleman vào năm 1977. Tên của thuật toán được đặt theo các chữ cái đầu tiên của ba người sáng tạo.

### Nguyên lý hoạt động:

1. Khóa công khai và khóa bí mật: RSA sử dụng cặp khóa bao gồm khóa công khai và khóa bí mật. Khóa công khai được chia sẻ công khai và dùng để mã hóa dữ liệu, trong khi khóa bí mật được giữ bí mật và dùng để giải mã dữ liệu.
2. Mã hóa và giải mã: Quá trình mã hóa dữ liệu bằng RSA là quá trình nhanh chóng và dễ dàng, trong khi quá trình giải mã dữ liệu đã được mã hóa trở lại dựa trên khóa bí mật là một vấn đề rất khó khăn đối với kẻ tấn công nếu họ không biết khóa bí mật.
3. Sử dụng số nguyên tố: RSA dựa trên tính khó của việc phân tích một số thành các thừa số nguyên tố. Khóa công khai và khóa bí mật của RSA dựa trên việc chọn hai số nguyên tố lớn và tính toán một số khác từ chúng.

Để minh họa nguyên lý mã hóa RSA, hãy xem xét một ví dụ đơn giản với các bước sau:

#### 1. Khởi tạo các tham số:

- Chọn hai số nguyên tố lớn  $p$  và  $q$ .
- Tính  $n = p * q$ .
- Tính hàm Euler của  $n$ :  $\phi(n) = (p-1) * (q-1)$ .
- Chọn một số nguyên  $e$  sao cho  $1 < e < \phi(n)$  và  $e$  là số nguyên tố cùng nhau với  $\phi(n)$ .  $e$  sẽ là khóa công khai.

#### 2. Tính khóa bí mật:

Tìm  $d$  sao cho  $(d * e) \% \phi(n) = 1$ .  $d$  sẽ là khóa bí mật.

**3. Mã hóa:** Mã hóa một thông điệp  $m$  thành một số nguyên  $c$  bằng cách sử dụng khóa công khai  $(n, e)$ :  $c = m^e \bmod n$ .

**4. Giải mã:** Giải mã số nguyên  $c$  thành thông điệp  $m$  bằng cách sử dụng khóa bí mật  $(n, d)$ :  $m = c^d \bmod n$ .

#### **Tính bảo mật :**

Độ an toàn của hệ thống RSA dựa trên 2 vấn đề: bài toán phân tích ra thừa số nguyên tố các số nguyên lớn và bài toán RSA. Vì vậy muốn xây dựng hệ RSA an toàn thì  $n=pq$  phải là một số đủ lớn, để không có khả năng phân tích nó về mặt tính toán. Vì vậy hiện nay người ta khuyến cáo sử dụng khóa có độ dài tối thiểu 2048 bit

#### **Các vấn đề đặt ra trong thực tế:**

Trong thực tế, mã hóa RSA đối mặt với một số vấn đề và thách thức, bao gồm:

1. **Lựa chọn khóa:** Việc chọn các số nguyên tố lớn và duy nhất là rất quan trọng để đảm bảo an toàn. Nếu các số nguyên tố không được chọn cẩn thận, hệ thống có thể dễ bị tấn công.
2. **Tính toán hiệu suất:** Mã hóa và giải mã RSA yêu cầu các phép tính mũ và modulo với các số rất lớn, điều này có thể tốn nhiều thời gian và tài nguyên máy tính.
3. **Quản lý khóa:** Việc phân phối và quản lý khóa công khai/riêng tư cần được thực hiện một cách an toàn để tránh bị lộ hoặc bị thay đổi.
4. **Bảo mật khóa riêng tư:** Khóa riêng tư phải được bảo mật tuyệt đối, vì nếu bị lộ, toàn bộ hệ thống mã hóa sẽ bị phá vỡ.
5. **Tấn công lượng tử:** Với sự phát triển của máy tính lượng tử, mã hóa RSA có thể bị đe dọa vì khả năng giải mã nhanh chóng các khóa RSA bằng thuật toán Shor.
6. **Cập nhật và bảo trì:** Hệ thống mã hóa cần được cập nhật thường xuyên để đối phó với các mối đe dọa mới và tăng cường bảo mật.

Để giải quyết những vấn đề này, các tổ chức thường sử dụng các biện pháp bảo mật bổ sung như hệ thống quản lý khóa, sử dụng các thuật toán mã hóa khác nhau cho các mục đích khác nhau, và thực hiện các biện pháp an ninh mạng nâng cao.

### **Ứng dụng của hệ mã RSA:**

1. Bảo mật dữ liệu truyền tải trên Internet: RSA được sử dụng để mã hóa dữ liệu, đảm bảo rằng chỉ người có khóa riêng mới có thể giải mã và đọc thông tin<sup>1</sup>.
2. Xác thực và chữ ký số: Trong giao dịch ngân hàng và các giao thức như SSL/TLS và SSH, RSA giúp xác minh danh tính và tính toàn vẹn của dữ liệu<sup>2</sup>.
3. Bảo mật thông tin cá nhân và tài chính: Trên các nền tảng trực tuyến, RSA giúp bảo vệ thông tin cá nhân và tài chính khỏi các nguy cơ tấn công mạng<sup>2</sup>.
4. Thương mại điện tử: RSA được sử dụng rộng rãi trong các hoạt động thương mại điện tử để bảo mật thông tin khách hàng và giao dịch<sup>1</sup>.
5. Chứng thực dữ liệu: RSA còn được sử dụng trong việc chứng thực dữ liệu, đảm bảo rằng dữ liệu không bị thay đổi trong quá trình truyền tải<sup>3</sup>.
6. Giao thức mạng: RSA là một phần quan trọng của các giao thức mạng như TLS và OpenVPN, giúp bảo mật thông tin truyền tải giữa các máy tính<sup>3</sup>.
7. Hệ thống ngân hàng và ứng dụng công nghệ thông tin: RSA được sử dụng trong các hệ thống ngân hàng để bảo mật giao dịch và trong các ứng dụng công nghệ thông tin khác để bảo vệ dữ liệu.

Những ứng dụng này cho thấy tầm quan trọng của RSA trong việc bảo vệ thông tin và dữ liệu trong thời đại số hiện nay.

### **2.3 Cài đặt thuật toán RSA**

Trong báo cáo này, chúng ta sẽ thảo luận về việc xây dựng một hệ mã hóa văn bản tiếng Việt sử dụng thuật toán RSA:

- **Ngôn ngữ Lập trình:** Chúng ta sẽ sử dụng C#, ngôn ngữ lập trình đa năng và phổ biến, để thực hiện đồ án này.
- **Tài nguyên Tính toán:** Đồ án sẽ tận dụng khả năng xử lý số lớn mạnh mẽ đã có sẵn trong C#

- **Nội dung Trọng tâm:** Chúng ta sẽ tập trung vào việc mã hóa các loại file như .pdf, .word ... sử dụng thuật toán RSA để mã hóa và thuật toán ký điện tử được tích hợp sẵn trong thư viện. Khóa được tạo ra sẽ có độ dài 1024 bits, đảm bảo an ninh và bảo mật cao.

## Cấu trúc thuật toán

### Class User:

```
public int id { get; set; }
public string name { get; set; }

public RSACryptoServiceProvider RSA { get; set; }
public string diaChiLuuKhoaXML { get; set; } = "";
public string diaChiLuuKhoaCongKhaiDoiPhuong { get; set; } = "";
public bool isEncryptFile { get; set; } = true;
public string N { get; set; } = "";
public string D { get; set; } = "";
public string E { get; set; } = "";
public string giaTriBam { get; set; } = "";

public string N_DoiPhuong { get; set; } = "";
public string D_DoiPhuong { get; set; } = "";
public string E_DoiPhuong { get; set; } = "";
public string diaChiFileMaHoa { get; set; } = "";
public string diaChiThuMucLuuFileDaMaHoa { get; set; } = "";
public string diaChiFileCanGiaiMa { get; set; } = "";
public string diaChiThuMucLuuFileSauGiaiMa { get; set; } = "";
public string diaChiFileMaHoaBamCanGiaiMa { get; set; } = "";
public byte[] giaTriChuKyByte { get; set; }

public string giaTriChuKyString { get; set; }
public string diachireturn { get; set; } = "";

public string diaChiLuuChuKy { get; set; } = "";

public string diaChiFileCanXacMinhTinhToanVen { get; set; } = "";
```

- `TaoKey(int doDaiBitKhoa)` : dùng để tạo khóa riêng tư có độ dài (512,1024,2048,4096 ...) sau khi phương thức chạy xong thì sẽ lưu thành 2 file 1 file chứa thông tin liên quan đến khóa riêng tư 1 và 1 file lưu thông tin liên quan đến khóa công khai được lấy từ thông tin của khóa riêng tư, các thuộc tính liên quan đến khóa của đối tượng đều được cập nhật bằng thông tin khóa vừa tạo.
- `LoadKeyCuaChinhMinh()` : dùng để sử dụng khóa đã được tạo từ trước đó . Sau khi phương thức chạy xong hàm thì đối tượng sẽ dùng các thông tin khóa trong file lưu khóa mà người dùng chọn.
- `void DungKeyCuaDoiPhuongCoSan()` : dùng để chọn file lưu khóa công khai của đối phương . Sau khi phương thức chạy xong thì các thông tin khóa của người nhận sẽ được cập nhật vào thuộc tính liên quan của đối tượng đang dùng.
- `void ChonFileGiaiMa()` : dùng để chọn địa chỉ lưu file cần giải mã bất cứ là loại file nào.
- `void ChonFileMaHoa()` : dùng để chọn địa chỉ lưu file cần mã hóa.
- `void ChonNoiLuuFileMaHoa()` : chọn folder để lưu file kết quả sau khi mã hóa.
- `void ChonNoiLuuFileDaGiaiMa()` : chọn folder để lưu file kết quả sau khi giải mã.
- `void RSA_Algorithm(string inputFile, string outputFile, RSAParameters RSAKeyInfo, bool isEncrypt)` : dùng để mã hóa và giải mã.
- `static string SHA256(string path)` : dùng để băm file với thuật toán băm SHA-256 và trả về chuỗi kết quả băm.
- `static RSAParameters ExtractPrivateKeyFromXml(string privateKeyXmlFilePath)` : dùng để lấy dữ liệu trong file lưu khóa và trả về một `RSAParameters`.
- `byte[] SignString(string plaintext)` : dùng để trả về kết quả băm đã được ký bởi khóa riêng tư của người gửi.
- `void WriteSignatureToFileInFolder(byte[] signature, string tenChuKy = "sign")` : dùng tạo file lưu trữ chữ ký đã được ký cho tài liệu cần gửi.
- `void ReadSignatureFromFile(string filePath)` : dùng để đọc chữ ký lưu trong file.
- `void ChonFileCanXacMinhTinhToanVen()` : dùng để chọn file cần xác minh tính toàn vẹn của file bằng khóa công khai của người gửi.
- `void ChonFileLuuChuKy()` : dùng để lấy địa chỉ lưu file có chữ ký cần xác minh.
- `bool VerifyString(byte[] signature)` : dùng để xác nhận chữ ký.



- void KiemTraTruocKhiGiaiMaVaXuLy() :dùng để kiểm tra xem có đúng đuôi mở rộng .lhde để giải mã.
- void KiemTraTruocKhiMaHoaVaXuLy() :dùng để kiểm tra xem đã chọn địa chỉ file cần mã hóa hay chưa, các thông tin cần thiết để mã hóa .
- void MaHoa() :dùng để kiểm tra địa chỉ thư mục cần lưu file đã mã hóa và đã có khóa chưa.
- void GiaiMa() :dùng để kiểm tra địa chỉ thư mục lưu file sau khi giải mã.

## 2.4 Cải tiến

- Trước đây code sẽ không mã hóa được các khối dữ liệu dài hơn độ dài khóa

Đoạn code trước:

```
private void RSA_Algorithm(string inputFile, string
outputFile, RSAParameters RSAKeyInfo, bool isEncrypt)
{
    try
    {
        FileStream fsInput = new FileStream(inputFile,
        FileMode.Open, FileAccess.Read); //Đọc file input
        FileStream fsCipherText = new
        FileStream(outputFile, FileMode.Create, FileAccess.Write);
        //Tạo file output
        fsCipherText.SetLength(0);
        byte[] bin, encryptedData;
        long rdlen = 0;
        long totlen = fsInput.Length;
        int len;
        RSA.ImportParameters(RSAKeyInfo); //Nhập thông tin
        khoá RSA (bao gồm khoá riêng)
        int maxBytesCanEncrypted;
        //RSA chỉ có thể mã hóa các khối dữ liệu ngắn hơn
        độ dài khóa, chia dữ liệu cho một số khối và sau đó mã hóa
        từng khối và sau đó hợp nhất chúng
        if (isEncrypt)
            maxBytesCanEncrypted = ((RSA.KeySize - 384) /
            8) + 37; // + 7: OAEP - Đệm mã hóa bất đối xứng tối ưu
        else
            maxBytesCanEncrypted = (RSA.KeySize / 8);
        //Read from the input file, then encrypt and write
        to the output file.
```

```

        if (isEncrypt) encryptedData =
RSA.Encrypt(bin, false); //Mã Hoá
        else encryptedData = RSA.Decrypt(bin, false);
//Giải mã

        fsCiperText.Close(); //save file
        fsInput.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Failed: trong lúc xử lý " +
ex.Message);
    }
}

```

Đoạn code sau khi cải tiến: (đoạn code cải tiến sử dụng phương pháp "chế độ mã hóa với đệm" (padding mode) như PKCS#1.5 . Chế độ mã hóa với đệm cho phép mã hóa dữ liệu có kích thước lớn hơn kích thước của khóa bằng cách thêm dữ liệu đệm vào trước khi mã hóa và loại bỏ nó khi giải mã.)

```

private void RSA_Algorithm(string inputFile, string
outputFile, RSAParameters RSAKeyInfo, bool isEncrypt)
{
    try
    {

        FileStream fsInput = new FileStream(inputFile,
        FileMode.Open, FileAccess.Read); //Đọc file input
        FileStream fsCiperText = new
        FileStream(outputFile, FileMode.Create, FileAccess.Write);
//Tạo file output
        fsCiperText.SetLength(0);
        byte[] bin, encryptedData;
        long rdlen = 0;
        long totlen = fsInput.Length;
        int len;
        RSA.ImportParameters(RSAKeyInfo); //Nhập thông
tin khoá RSA (bao gồm khoá riêng)
        int maxBytesCanEncrypted;
        //RSA chỉ có thể mã hóa các khối dữ liệu ngắn hơn
độ dài khóa, chia dữ liệu cho một số khối và sau đó mã hóa
từng khối và sau đó hợp nhất chúng
        if (isEncrypt)
            maxBytesCanEncrypted = ((RSA.KeySize - 384) /
8) + 37; // + 7: OAEP - Đệm mã hóa bất đối xứng tối ưu
        else
            maxBytesCanEncrypted = (RSA.KeySize / 8);
    }
}

```

```

        //Read from the input file, then encrypt and
        write to the output file.
        while (rdlen < totlen)
        {
            if (totlen - rdlen < maxBytesCanEncrypted)
maxBytesCanEncrypted = (int)(totlen - rdlen);
            bin = new byte[maxBytesCanEncrypted];
            len = fsInput.Read(bin, 0,
maxBytesCanEncrypted);

            if (isEncrypt) encryptedData =
RSA.Encrypt(bin, false); //Mã Hoá
            else encryptedData = RSA.Decrypt(bin, false);
//Giải mã

            fsCiperText.Write(encryptedData, 0,
encryptedData.Length);
            rdlen = rdlen + len;
        }

        fsCiperText.Close(); //save file
        fsInput.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Failed: trong lúc xử lý " +
ex.Message);
    }
}

```

Ứng dụng quản lý chữ ký số: Được xây dựng trên mô hình 3 lớp (BLL,DAL,DTO,GUI)

❖ TrangChu.cs :

- void OpenChildForm(Form childForm): dùng để mở form con .
- void setBackgroundColorDefault() : dùng để set màu nền chỉ định cho các control trong form .
- void setBackgroundColorMacDinhChoTatCaButton(): dùng để set màu forecolor cho các control.
- void setBackgroundColor\_ForeColor\_MacDinh(Button btn): set màu nền cho các nút đang được kích hoạt.

❖ DanhSachTheCCCD.cs :

- void LoadDuLieuBieuDoGioiTinh() : dùng để hiện thị dữ liệu về giới tính lấy được ở tầng BLL và hiện thị lên chart.
- void LoadDuLieuBieuDoNoiCap() : dùng để hiện thị dữ liệu về nơi cấp lấy được ở tầng BLL và hiện thị lên chart.
- Dictionary<string, string> renderCapKhoa(string name = "user", int doDaiBitKhoa = 1024) : dùng để tạo cặp khóa cho người dùng

❖ ChiTietAnh.cs :

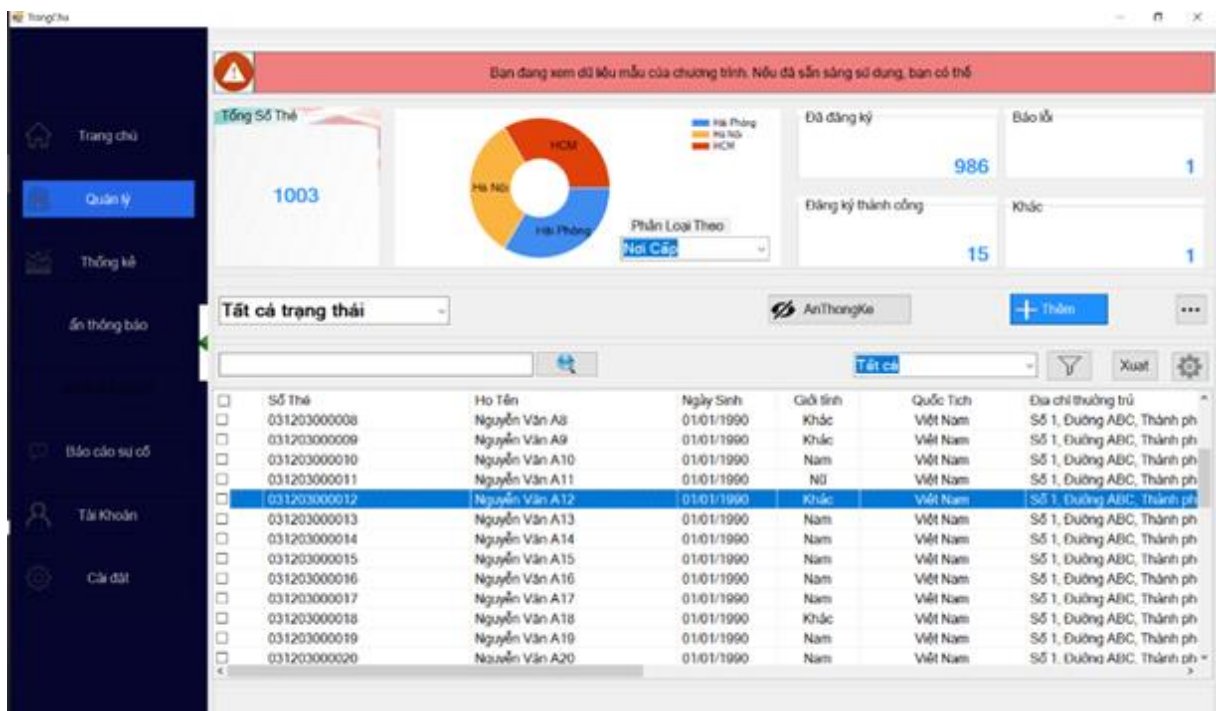
- ChiTietAnh\_Load(object sender, EventArgs e) : dùng để hiện thị chi tiết ảnh.

❖ Edit.cs: (form dùng để xem, thêm, sửa) .

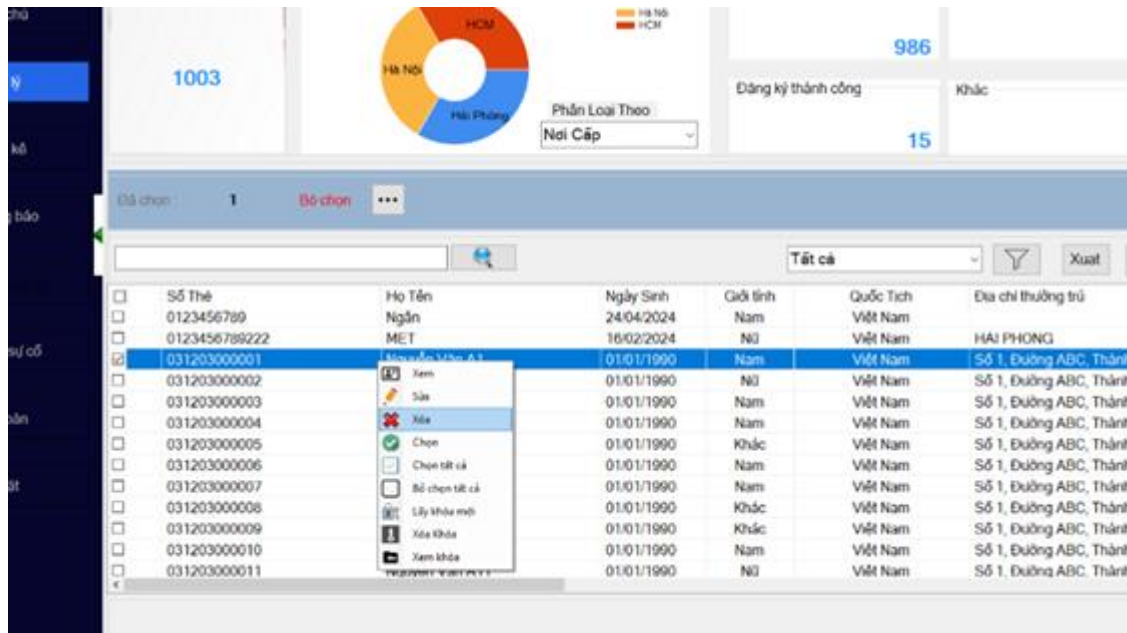
❖ DAL:

- int LayTongSoDongBangTheCanCuocHandle(Dictionary<string, string> option): dùng để lấy tổng số dòng có option kèm theo.
- The LayThongTinChiTietTheHandle(string maTheCanLay) :Lấy thông tin chi tiết một thẻ từ db.
- List<The> TimKiemHandle(string noiDungCanTim,Dictionary<string, string> option) :Dùng để trả về Danh sách thẻ thỏa mãn điều kiện tìm kiếm.
- Dictionary<string, int> ThongKeTheoNoiCapHandle() :dùng để trả về một dictionary có dữ liệu số lượng đăng ký của từng nơi cấp.
- Dictionary<string, string> ThongKeTheoGioiTinhVaTrangThaiDangKyHandle(): dùng để trả về một dictionary có dữ liệu số lượng đăng ký của từng giới tính.
- List<The> LayToanBoDanhSachTheHandle(int trangHienTai, int soDongMoiTrang,Dictionary<string,string> option):lấy toàn bộ danh sách thẻ trong db.
- string ThemTheHandle(The theCanThem) :Dùng để thêm thông tin thẻ vào db.
- string SuaTheHandle(The theCanSua): Dùng để sửa thông tin thẻ vào db.

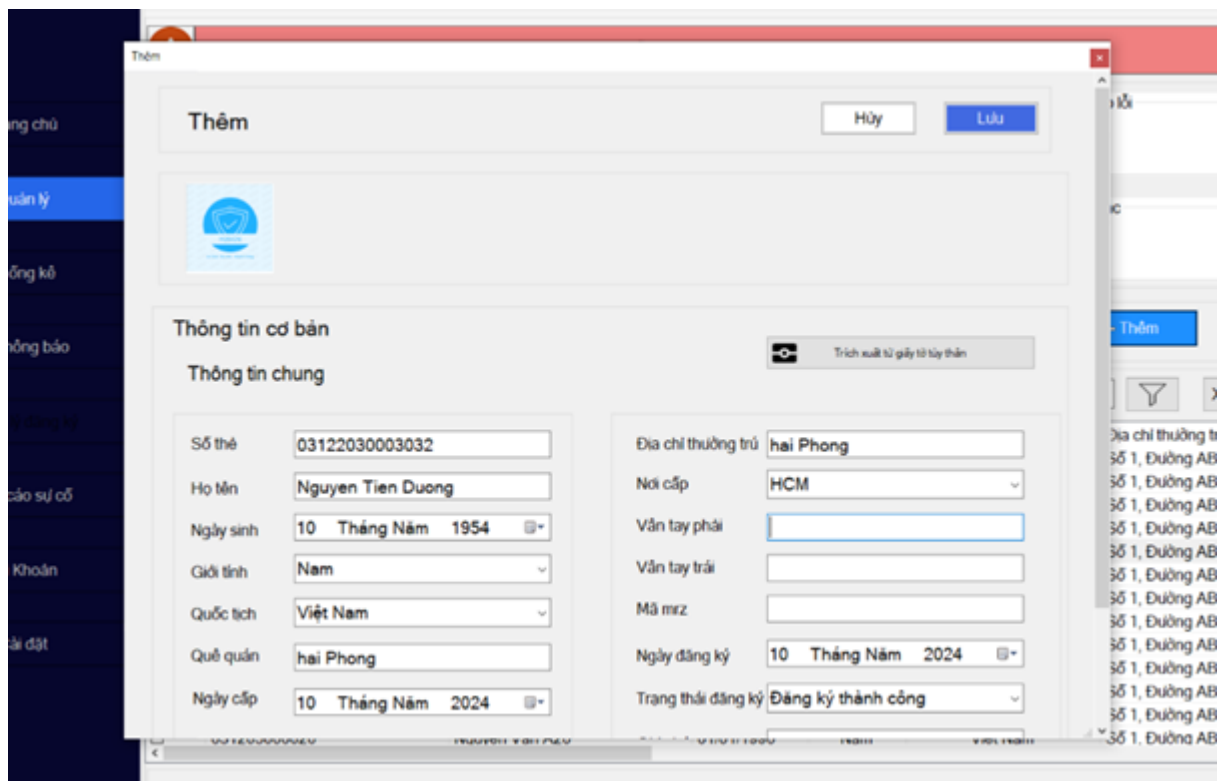
- string XoaDongHandle(string soTheCanXoa) : Dùng để xóa thông tin thẻ trong db.
- List<string> LayDanhSachNoiCapHandle() : Dùng để lấy danh sách nơi cấp từ db.
- string SuaTrangThaiDangKyHandle(string maThe, string trangThaiDangKy, string ghiChu): dùng để sửa trạng thái đăng ký.
- string LayCapKhoaMoiHandle(string maTheCanLayKhoa, Dictionary<string, string> capKhoa) : dùng để ghi cập khóa mới cho người dùng.
- string XoaCapKhoaHandle(string maTheCanXoa) : dùng để xóa cập khóa trong db và cả file lưu trữ .



Hình 2.1. Giao diện quản lý



Hình 2.2. Giao diện để xóa



Hình 2.3. Giao diện thêm

**Sửa** Hủy Lưu

**Thông tin cơ bản**

**Thông tin chung**

**Số thẻ** 031203000002

**Họ tên** Nguyễn Văn A2

**Ngày sinh** 01 Tháng Giêng 1990

**Giới tính** Nữ

**Quốc tịch** Việt Nam

**Quê quán** Hà Nội

**Ngày cấp** 01 Tháng Giêng 2010

**Địa chỉ thường trú** Số 1, Đường ABC, Thành phố X

**Nơi cấp** Hải Phòng

**Vân tay phải** van\_tay\_phai\_a

**Vân tay trái** van\_tay\_trai\_a

**Mã mrz** MRZ00002

**Ngày đăng ký** 24 Tháng Tư 2024

**Trạng thái đăng ký** Đăng ký thành công

**Trích xuất từ giấy tờ tùy thân**

Hình 2.4. Giao diện sửa

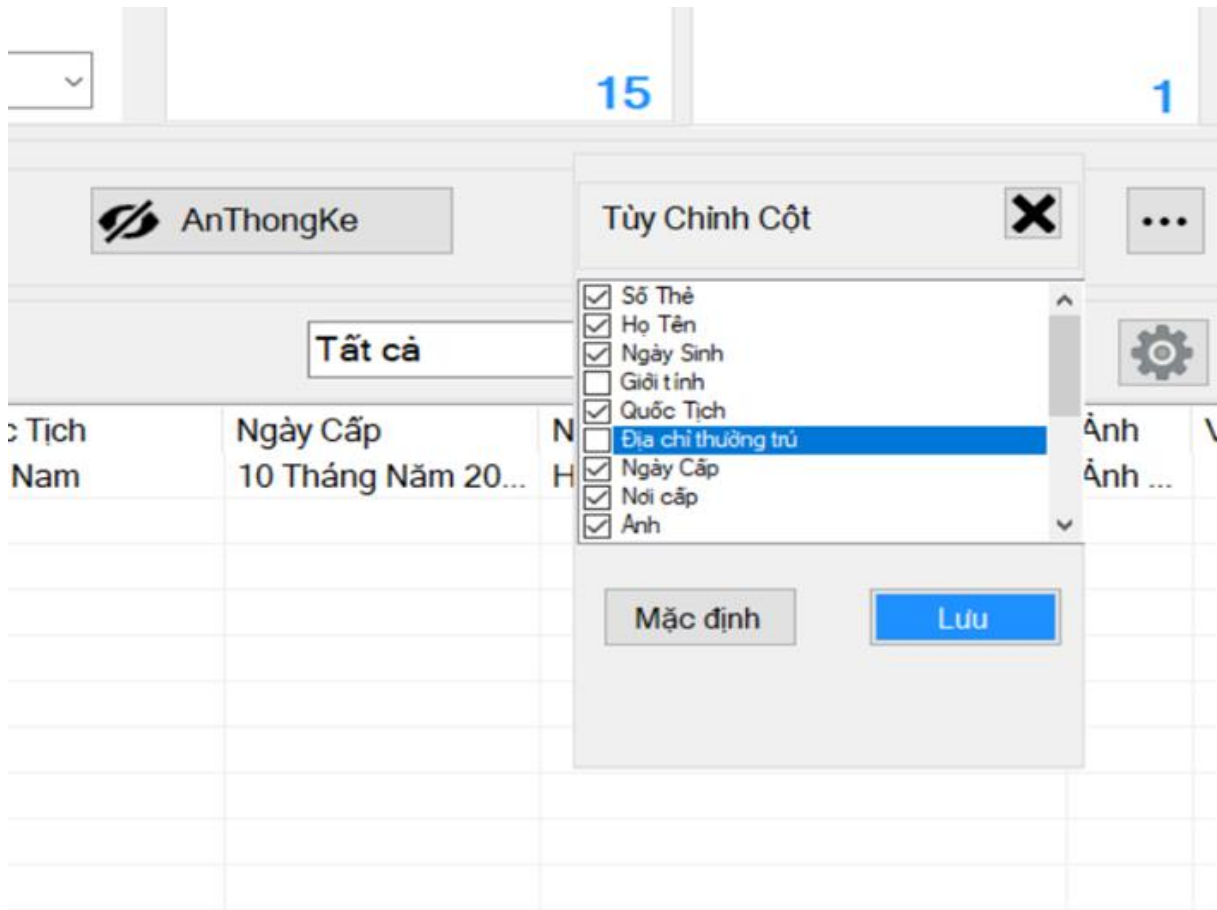
**Tất cả trạng thái**

**AnThongKe** + Thêm

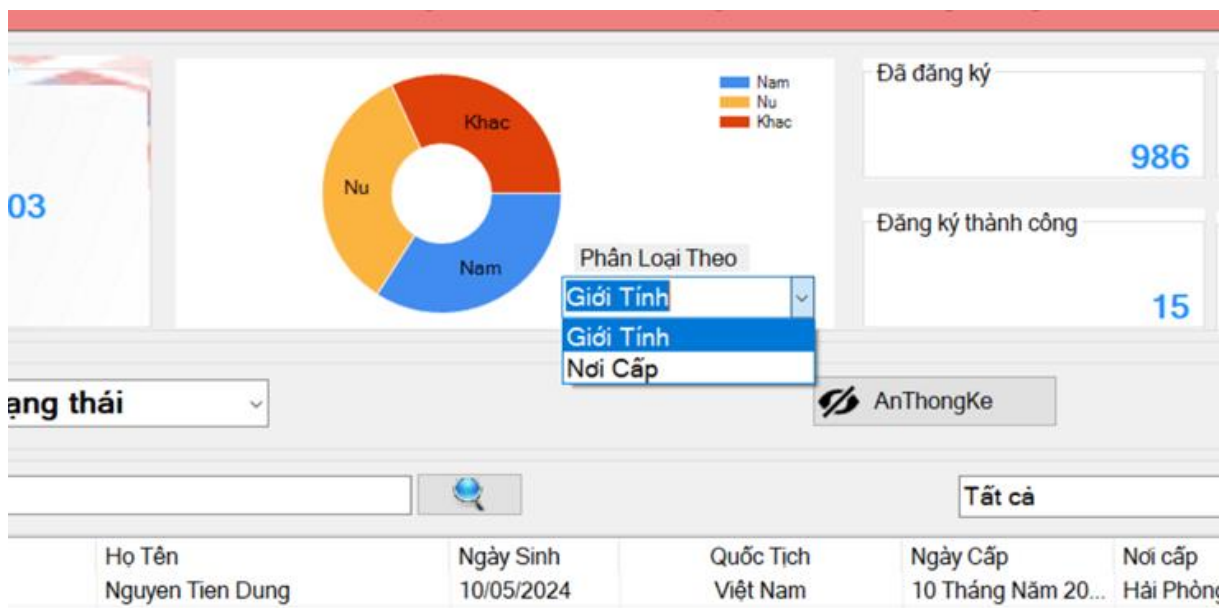
**dung** **Tất cả** **Xuat**

<input type="checkbox"/>	Số Thẻ	Họ Tên	Ngày Sinh	Giới tính	Quốc Tịch	Địa chỉ thường trú
<input type="checkbox"/>	031203007302	Nguyen Tien Dung	10/05/2024	Nam	Việt Nam	

Hình 2.5. Chức năng tìm kiếm



Hình 2.6. Giao diện tùy chỉnh hiển thị cột

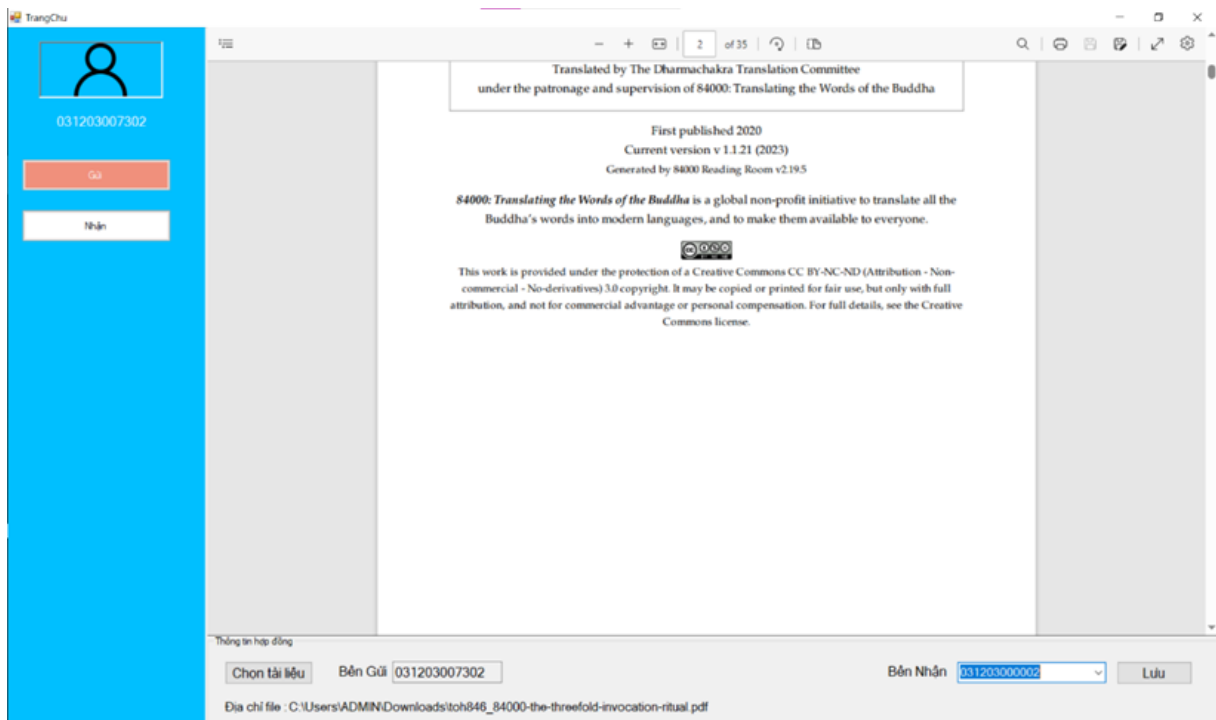


Hình 2.7. Giao diện lọc theo giới tính và nơi cấp

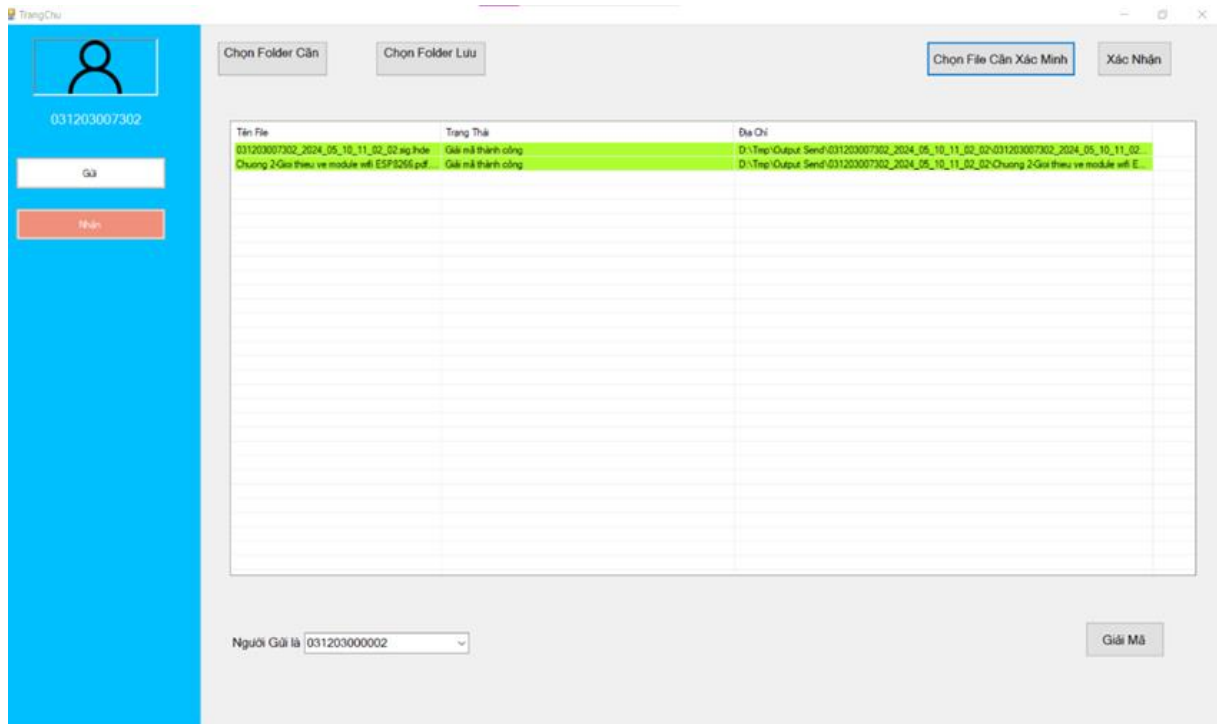




Hình 2.8. Giao diện đăng nhập H2sign



Hình 2.9. Giao diện để tạo file gửi



Hình 2.10. Giao diện để giải mã và xác nhận chữ ký

Ứng dụng H2Sign :

Dùng class User ở trên để xây dựng và có chỉnh sửa để phù hợp với mô hình dữ liệu

- ❖ static RSAParameters ExtractPrivateKeyFromXmlString(string stringXML) : dùng để tạo ra đối tượng RSAParameters từ string xml

1. Form đăng nhập (DangNhap.cs)
2. Form tạo file để ký và gửi (HopDongDaKy.cs)
3. Form giải mã và xác nhận chữ ký (Hopdongdoiky.cs)

## CHƯƠNG 3: KẾT LUẬN VÀ ĐỊNH HƯỚNG

### 3.1 Kết luận

Qua quá trình tìm hiểu và thực hành, chúng ta có thể thấy rằng việc ứng dụng chữ ký RSA vào việc gửi nhận file và chữ ký điện tử mang lại nhiều lợi ích đáng kể. Đầu tiên, RSA cung cấp một lớp bảo mật mạnh mẽ cho dữ liệu, giúp ngăn chặn việc truy cập trái phép và đảm bảo rằng chỉ những người có quyền mới có thể giải mã thông tin. Thứ hai, chữ ký điện tử bằng RSA giúp xác thực nguồn gốc của dữ liệu, đảm bảo rằng nó không bị thay đổi trong quá trình truyền tải.

Tuy nhiên, cũng cần lưu ý rằng việc sử dụng RSA đòi hỏi một sự hiểu biết kỹ thuật nhất định và việc triển khai không đúng cách có thể dẫn đến các lỗ hổng bảo mật. Do đó, việc đào tạo và nâng cao nhận thức về bảo mật cho người dùng là vô cùng quan trọng.

Kết luận, việc ứng dụng chữ ký RSA vào việc gửi nhận file và chữ ký điện tử đã và sẽ tiếp tục đóng một vai trò quan trọng trong việc bảo vệ thông tin và dữ liệu trực tuyến. Với sự phát triển không ngừng của công nghệ, chúng ta có thể kỳ vọng vào những tiến bộ mới trong lĩnh vực này, giúp tăng cường bảo mật và quyền riêng tư cho người dùng trên toàn cầu.

### 3.2 Định hướng

Trong bối cảnh sự phát triển không ngừng của máy tính lượng tử, hệ mật mã RSA có thể đối mặt với những rủi ro về an ninh. Máy tính lượng tử có khả năng giải mã các hệ mật mã hiện tại nhanh chóng hơn rất nhiều so với máy tính cổ điển, điều này đặt ra thách thức lớn cho việc bảo mật thông tin trong tương lai.

Do đó, việc tìm hiểu và phát triển các hệ mật mã mới, cũng như cải tiến RSA để đối phó với thách thức từ máy tính lượng tử, trở thành một hướng đi cần thiết và cấp bách. Điều này không chỉ đòi hỏi sự tiến bộ về mặt công nghệ, mà còn cần sự thấu hiểu sâu sắc về cách hoạt động của RSA và các hệ mật mã khác.

Ngoài ra, việc nghiên cứu sâu hơn về cách hoạt động của RSA cũng rất quan trọng, nhằm tối ưu hóa hiệu suất và đảm bảo an toàn cho hệ thống. Việc này không chỉ giúp chúng ta hiểu rõ hơn về cách RSA bảo vệ dữ liệu, mà còn giúp chúng ta tìm ra cách để cải tiến và tối ưu hóa RSA trong tương lai.

Kết luận, trong thời đại số hóa ngày nay, việc đảm bảo an toàn cho dữ liệu là vô cùng quan trọng. Với sự phát triển của máy tính lượng tử, chúng ta cần phải không ngừng nghiên cứu và cải tiến các hệ mật mã, bao gồm RSA, để đảm bảo an toàn cho dữ liệu trong tương lai."

### Tài Liệu Tham Khảo

- [1] fastca.vn, “RSA là gì? Cách thức hoạt động của RSA với chữ ký số,” [Trực tuyến]. Available: <https://fastca.vn/rsa-la-gi-cach-thuc-hoat-dong-cua-rsa-voi-chu-ky-so/>. [Đã truy cập 6 5 2024].
- [2] itnavi, “Cơ chế hoạt động và ứng dụng của RSA trong công nghệ thông tin,” 29 6 2021. [Trực tuyến]. Available: <https://itnavi.com.vn/blog/rsa-la-gi/>. [Đã truy cập 6 5 2024].
- [3] mi2, “Tại sao lại cần an toàn thông tin? Các giải pháp bảo vệ an toàn thông tin,” 28 8 2021. [Trực tuyến]. Available: <https://mi2.com.vn/tai-sao-lai-can-an-toan-thong-tin-cac-giai-phap-bao-ve-an-toan-thong-tin/>. [Đã truy cập 5 5 2024].
- [4] “Nguyên tắc bảo đảm an toàn thông tin được thực hiện như thế nào? An toàn thông tin được hiểu ra sao?,” [Trực tuyến]. Available: [https://thuvienphapluat.vn/phap-luat/nguyen-tac-bao-dam-an-toan-thong-tin-duoc-thuc-hien-nhu-the-nao-an-toan-thong-tin-duoc-hieu-ra-sao-32534.html#google\\_vignette](https://thuvienphapluat.vn/phap-luat/nguyen-tac-bao-dam-an-toan-thong-tin-duoc-thuc-hien-nhu-the-nao-an-toan-thong-tin-duoc-hieu-ra-sao-32534.html#google_vignette). [Đã truy cập 2 5 2024].
- [5] H. Nguyễn, “RSA là gì? Mã hóa RSA hoạt động như thế nào?,” 30 01 2024. [Trực tuyến]. Available: <https://vietnix.vn/rsa/>. [Đã truy cập 03 05 2024].
- [6] manhhomienbienthuy, “Hệ mã hóa RSA và chữ ký số,” 23 2 2017. [Trực tuyến]. Available: <https://viblo.asia/p/he-ma-hoa-rsa-va-chu-ky-so-6J3ZgkgMZmB>. [Đã truy cập 04 5 2024].