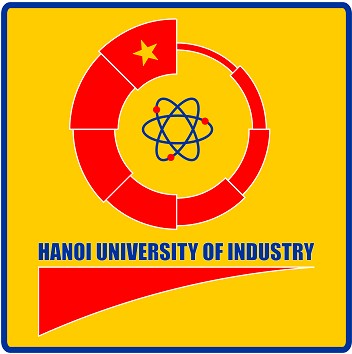
**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN MÔN AN TOÀN VÀ BẢO MẬT THÔNG TIN**

**ĐỀ TÀI: XÂY DỰNG CHƯƠNG TRÌNH MÃ HÓA VÀ GIẢI MÃ RSA**

**CBHD:** ThS. Trần Phương Nhung

**Nhóm:**  14

**Thành viên nhóm:**

1. Đào Thanh Tùng - 2019604965

2. Bùi Thị Tươi - 2019605102

3. Trần Thị Yến - 2019697911

Hà Nội – 2022

**MỤC LỤC**

[LỜI MỞ ĐẦU 4](#_Toc113545294)

[CHƯƠNG 1. TỔNG QUAN 5](#_Toc113545295)

[1.1. Giới thiệu về hệ mật mã 5](#_Toc113545296)

[1.2. Các hệ mật mã 6](#_Toc113545297)

[1.3. Hệ mật mã công khai 7](#_Toc113545298)

[1.4. Giới thiệu chung về hệ mật mã RSA 8](#_Toc113545299)

[1.5. Cơ sở lý thuyết 9](#_Toc113545300)

[1.6. Thuật toán tạo khóa, mã hóa và giải mã hệ mật RSA 9](#_Toc113545301)

[1.6.1. Thuật toán tạo khóa 9](#_Toc113545302)

[1.6.2. Thuật toán mã hóa 10](#_Toc113545303)

[1.6.3. Thuật toán giải mã 10](#_Toc113545304)

[1.7. Ưu và nhược điểm của hệ mật mã RSA 10](#_Toc113545305)

[1.7.1. Ưu điểm 10](#_Toc113545306)

[1.7.2. Nhược điểm 10](#_Toc113545307)

[1.8. Độ an toàn của hệ mật mã RSA 10](#_Toc113545308)

[CHƯƠNG 2. KẾT QUẢ NGHIÊN CỨU 11](#_Toc113545309)

[2.1. Giới thiệu 11](#_Toc113545310)

[2.2. Nội dung thuật toán 11](#_Toc113545311)

[2.3. Thiết kế và cài đặt chương trình demo thuật toán 12](#_Toc113545312)

[2.3.1. Giao diện chương trình demo 12](#_Toc113545313)

[2.3.1.1. Giao diện demo theo ngôn ngữ Java 12](#_Toc113545314)

[2.3.1.2. Giao diện demo theo ngôn ngữ Python 13](#_Toc113545315)

[2.3.1.3. Giao diện demo theo ngôn ngữ C# 14](#_Toc113545316)

[2.4. Cài đặt và triển khai 14](#_Toc113545317)

[2.4.1. Giới thiệu công cụ 14](#_Toc113545318)

[2.4.1.1. Giới thiệu công cụ Eclipse 14](#_Toc113545319)

[2.4.1.2. Giới thiệu công cụ Visual Studio 14](#_Toc113545320)

[2.4.2. Hướng dẫn cài đặt và chạy chương trình demo đã cài đặt 15](#_Toc113545321)

[2.4.2.1. Cài đặt Eclipse và chạy chương trình demo 15](#_Toc113545322)

[2.4.2.2. Cài đặt Visual Studio Code và chạy chương trình demo Python 15](#_Toc113545323)

[2.4.2.4. Cài đặt Visual Studio và chạy chương trình demo C# 16](#_Toc113545324)

[2.5. Thực hiện bài toán 16](#_Toc113545325)

[2.5.1. Phân công công việc 16](#_Toc113545326)

[2.5.2. Đào Thanh Tùng - abc 17](#_Toc113545327)

[2.5.3. Bùi Thị Tươi - abc 17](#_Toc113545328)

[2.5.4. Yến - Tìm hiểu về nguyên lý hoạt động của RSA và cài đặt chương trình demo 17](#_Toc113545329)

[CHƯƠNG 3. KIẾN THỨC LĨNH HỘI VÀ BÀI HỌC KINH NGHIỆM 18](#_Toc113545330)

[3.1. Nội dung đã thực hiện 18](#_Toc113545331)

[3.1.1. Các kiến thức đã học được thông qua thực hiện bài tập lớn 18](#_Toc113545332)

[3.1.2. Các kỹ năng đã học được thông qua thực hiện bài tập lớn 18](#_Toc113545333)

[3.1.3. Những bài học kinh nghiệm được rút ra sau khi kết thúc bài tập lớn 18](#_Toc113545334)

[DANH MỤC TÀI LIỆU THAM KHẢO 21](#_Toc113545335)

LỜI MỞ ĐẦU

Ngay từ thuở sơ khai của lịch sử, vấn đề bảo mật và mã hóa thông tin đã tồn tại và được nghiên cứu suốt chiều dài văn minh nhân loại. Từ xa xưa, trước cả sự tồn tại của những khái niệm về máy tính sơ khai nhất, con người đã sáng tao ra các hệ mật mã cổ điển với bút và giấy, thậm chí là hỗ trợ từ những dụng cụ cơ khí đơn giản.

Vào đầu thế kỷ 20, sự xuất hiện của các cơ cấu cơ khí và điện cơ, chẳng hạn như máy Enigma, đã cung cấp những cơ chế phức tạp và hiệu quả hơn cho việc mật mã hóa. Sự ra đời và phát triển mạnh mẽ của ngành điện tử và máy tính, cũng như vai trò ngày càng quan trọng của chúng trong đời sống cũng như nền văn minh làm bùng nổ tầm quan trọng của việc bảo mật và mã hóa thông tin, nhờ đó tạo điều kiện để mật mã học có một bước nhảy vọt lớn. Nhờ sự trợ giúp của máy tính, rất nhiều hệ mật mã hiện đại đã ra đời dựa trên cơ sở đại số Modulo và các thuật toán logarithm rời rạc, đều là những hệ mật mã có tính bảo mật cao vượt trội. Một trong số các hệ mật mã hiện đại phổ biến nhất vẫn được sử dụng cho đến ngày hôm nay chính là hệ mật mã RSA.

Dựa trên sự hướng dẫn của giảng viên - ***ThS. Trần Phương Nhung***, các thành viên trong nhóm đã tiến hành tìm hiểu về các thuật toán mã hóa và giải mã hệ mật mã RSA. Bên cạnh đó nhóm chúng em đã tiến hành xây dựng các chương trình demo hệ mật RSA bằng nhiều ngôn ngữ khác nhau. Trong suốt quá trình hoàn thiện bài tập lớn, nhóm chúng em đã cố gắng tuy nhiên không thể tránh khỏi sai sót nên chúng em rất mong nhận được sự góp ý từ phía cô để bài tập lớn của nhóm được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

*Nhóm sinh viên thực hiện.*

CHƯƠNG 1. TỔNG QUAN

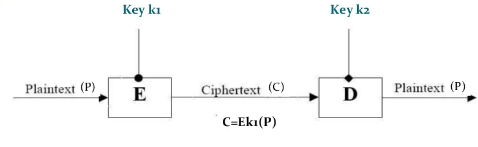
1.1. Giới thiệu về hệ mật mã

Để đảm bảo việc truyền tin an toàn và kiểm tra tính tồn vẹn của thông tin, người ta thường mã hóa thông tin trước khi truyền đi bằng các một số các hệ mật như DES, Triple DES(3DES), RC4, AES, RSA, Rabin, Diffle-Hellman, Elgamal, ... Một hệ thống mật mã là một hệ bao gồm 5 thành phần (P, C, K, E, D) thỏa mãn các tính chất:

* P (Plaintext) là tập hợp hữu hạn các bản rõ có thể (hay còn gọi là không gian bản rõ).
* C (Ciphertext) là tập hợp hữu hạn các bản mã có thể (hay còn gọi là không gian bản mã).
* K (Key) là tập hợp các bản khóa có thể (hay còn gọi là không gian khóa).
* E (Encryption) là tập hợp các quy tắc mã hóa có thể (hay còn gọi là không gian các hàm mã hóa).
* D (Decryption) là tập hợp các quy tắc giải mã có thể (hay còn gọi là không gian các hàm giải mã)

Quá trình mã hóa được tiến hành bằng cách áp dụng hàm tóan học E lên thông tin P để trở thành thông tin đã mã hóa C.

Quá trình giải mã được tiến hành ngược lại: áp dụng hàm D lên thông tin C để được thông tin đã giải mã.



Hình 1. Quá trình mã hóa và giải mã

1.2. Các hệ mật mã

Hệ mật mã gồm 2 loại:

* Hệ mật mã đối xứng (hay còn gọi là hệ mật mã khóa bí mật): là những hệ mật dùng chung một khóa cả trong quá trình mã hóa và giải mã thơng tin. Do đó khóa phải được giữ bí mật tuyệt đối. Một sổ thuật toán nổi tiếng trong mã hóa đối xứng là: DES, Triple DES(3DES), RC4, AES, ...
* Hệ mật mã bất đối xứng (hay còn gọi là mật mã khóa công khai): Các hệ mật này dùng một khóa để mã hóa sau đó dùng một khóa khác để giải mã, nghĩa là khóa để mã hóa và khóa để giải mã là khác nhau.

Các khóa này tạo nên từng cặp chuyển đổi ngược nhau và không có khóa nào có thể suy được ra khóa còn lại. Khóa dùng để mã hóa có thể công khai nhưng khóa dùng để giải mã thì giữ bí mật. Do đó trong thuật toán này có hai loại khóa: khóa dùng để mã hóa được gọi là khóa công khai-Public Key còn khóa để giải mã được gọi là khóa bí mật Private Key. Một số thuật toán mã hóa công khai nổi tiếng: DiffleHellman, Elgamal, RSA, Rabin, ...

1.3. Hệ mật mã công khai

Hệ mật mã công khai là bước tiến lớn của ngành mật mã. Hệ mật mã ra đời đã phá bỏ tư duy cũ về mật mã, đồng thời có nhiều ứng dụng to lớn như: phân phối khóa, chữ ký số v.v... Mặc dù giải quyết được điểm yếu logic của hệ mật mã khóa đối xứng nhưng hệ mật mã khóa công khai (bất đối xứng) đồng thời biến điểm mạnh của hệ mật mã khóa bí mật thành điểm yếu của mình. Hệ khóa công khai tính toán chậm do liên tục xử lý các số lớn (ở Elgamal là tính toán với số nguyên tố lớn). Để đảm bảo tính an toàn của hệ mật mã Elgamal. Một quy luật tự nhiên là yếu ở đâu thì ta khắc phục ở đó. Để giảm thời gian tính toán của hệ mật mã khóa cơng khai, ta đang cổ gắng thực hiện Giảm độ dài khóa đồng thời vẫn giữ được tính an tồn của hệ mật mã. Đây chính là xu thế của mật mã học hiện nay.

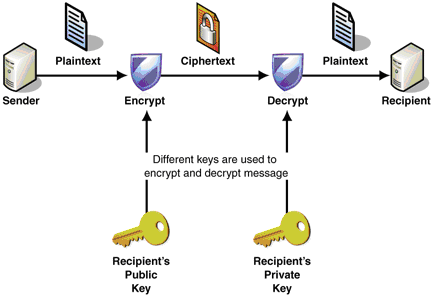
Xuất phát từ mong muổn tìm hiểu, tạo nên một chưong trình demo mã hóa và giải mã thông tin và lợi ích mà việc mã hóa mang lại như trên cùng với yêu cầu bài tập lớn của môn an toàn và bảo mật thông tin nên nhóm 14 chúng em đã lựa chọn tìm hiểu về hệ mã hóa công khai (hệ mật mã bất đối xứng) RSA với mục đích hiểu rõ phưong pháp, cách thức thực hiện mã hóa, giải mã và các thuật toán dùng để mã hóa và giải mã thông tin. Qua đó ta có thể thấy được tầm quan trọng của hệ mã hóa công khai RSA đối với việc truyền tin an toàn và kiểm tra tính toàn vẹn của thông tin.

1.4. Giới thiệu chung về hệ mật mã RSA

RSA là thuật toán mã hóa công khai đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa. Nó đánh dấu một sự tiến bộ vượt bậc của lĩnh vực mật mã học trong việc sử dụng khóa công cộng. RSA đang được sử dụng phổ biến trong thương mại điện tử và được cho là đảm bảo an toàn với điều kiện độ dài khóa đủ lớn.

Thuật toán 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 có hai khóa: khóa công khai (public key) và khóa bí mật (private key). 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 (private key) mới có thể giải mã được.



Hình 2. Sơ đồ mã hóa và giải mã hệ mật mã RSA

1.5. Cơ sở lý thuyết

Mấu chốt cơ bản của việc sinh khóa trong RSA là tìm được bộ 3 số tự nhiên *e, d* và *n* sao cho:

và một điểm không thể bỏ qua là cần bảo mật cho d sao cho dù biết e, n hay thậm chí cả m cũng không thể tìm ra d được. Điều này đạt được bằng cách chọn e,d đủ lớn sao cho với năng lực tính toán của máy tính hiện thời việc tìm ra e,d từ n gần như bất khả.

1.6. Thuật toán tạo khóa, mã hóa và giải mã hệ mật RSA

1.6.1. Thuật toán tạo khóa

* B1: Chọn 2 số nguyên tố p,q đủ lớn
* B2: Tính , sau này n sẽ được dùng làm modulus trong cả public key và private key.
* B3: Tính một số giả nguyên tố là hàm phi euler của n: . Giá trị này sẽ được giữ bí mật
* B4: Chọn ngẫu nhiên khóa mã hóa b thỏa mã sao cho
* B5: Tìm khóa giải mã a sao cho (tức a là nghịch đảo modulo của b theo )
* Khi đó khóa công khai và

1.6.2. Thuật toán mã hóa

Để mã hóa thông tin M ta làm theo các bước:

* B1: Với bản rõ M, cần chuyển nó thành một số tự nhiên x trong khoảng (0,n) sao cho x,n nguyên tố cùng nhau.
* B2: Tính . Khi đó y chính là bản mã sẽ được chuyển cho người nhận

1.6.3. Thuật toán giải mã

Để giải mã thông tin từ bản mã y ta làm như sau:

* Bl: Tính . Khi đó x chính là bản rõ ban đầu

1.7. Ưu và nhược điểm của hệ mật mã RSA

1.7.1. Ưu điểm

- Tính an toàn cao trong trường hợp p,q đủ lớn. Việc tính ngược lại p và q từ n là chuyện hầu như không thể với hai số nguyên tố 2048 bít .

- Cơ chế public – private key là một cơ chế rất hiệu quả để sử dụng trong các bài toán thương mại điện tử và chữ ký số.

1.7.2. Nhược điểm

- Tốc độ chậm do phải xử lý với số nguyên lớn.

- Dung lượng bộ nhớ dùng cho việc lưu trữ khóa yêu cầu phải lớn.

- Dễ bị bẻ gãy nếu n không đủ lớn

1.8. Độ an toàn của hệ mật mã RSA

Tính an toàn của RSA chủ yếu dựa vào bố tạo số ngẫu nhiên sinh ra 2 số nguyên tố p và q ban đầu.

Việc tính ngược lại p và q từ n là chuyện hầu như không thể với hai số nguyên tố 2048 bít .

Nhưng việc tính ra d từ p và q là việc rất dễ dàng.

Do vậy, nếu như một bên nào đó đoán ra được hoặc tìm ra lỗ hổng của bộ sinh số ngẫu nhiên đó thì coi RSA bị hóa giải.

Tuy nhiên, với sự phát triển của công nghệ, các siêu máy tính xuất hiện ngày càng nhiều. Cùng với chúng ta máy tính lượng tử cho phép tính toán với tốc độ cao hơn rất nhiều có thể sẽ phá vỡ sự bảo mật của RSA.

Ngày từ năm 1993, thuật toán Shor đã được phát triển và chỉ ra rằng máy tính lượng tử có thể giải bài toán phân tích ra thừa số trong thời gian đa thức. Rất may là những điều này mới chỉ là lý thuyết vì đến thời điểm hiện tại và trong vài năm tới, máy tính lượng tử vẫn chưa hoàn thiện.

CHƯƠNG 2. KẾT QUẢ NGHIÊN CỨU

2.1. Giới thiệu

- Tên đề tài nghiên cứu: Xây dựng chương trình mã hóa và giải mã RSA

- Các bước thực hiện triển khai đề tài bao gồm:

* Nghiên cứu nội dung các thuật toán
* Thiết kế và cài đặt chương trình demo thuật toán

- Hình thức sản phẩm: Sản phẩm ứng dụng

- Kết quả đạt được: Nghiên cứu, Cài đặt demo thuật toán

2.2. Nội dung thuật toán

Bước 1: Tạo khoá

* Chọn p,q là số nguyên tố.
* Tính ,
* Chọn b sao cho sao cho
* Tính

Từ đó ta tạo được khoá:



Bước 2: Mã hóa

* Gọi phần tử của bản rõ là x, phần tử của bản mã là y.
* Khi đó

Bước 3: Giải mã



2.3. Thiết kế và cài đặt chương trình demo thuật toán

2.3.1. Giao diện chương trình demo

2.3.1.1. Giao diện demo theo ngôn ngữ Java

Graphical user interface, application, Word

Description automatically generated

2.3.1.2. Giao diện demo theo ngôn ngữ Python

Graphical user interface

Description automatically generated

2.3.1.3. Giao diện demo theo ngôn ngữ C#

Graphical user interface, application, Teams

Description automatically generated

2.4. Cài đặt và triển khai

2.4.1. Giới thiệu công cụ

2.4.1.1. Giới thiệu công cụ Eclipse

Eclipse là một mơi trường phát triển tích hợp (IDE) cho Java. Eclipse cho phép các ứng dụng được phát triển từ một tập hợp các thành phần phần mềm được gọi là modules. Eclipse chạy trên Windows, macOS, Linux và Solaris.

Giao diện dễ sử dụng vời người lập trình. Có đầy đủ các thư viện của ngôn ngữ Java giúp người lập trình sử dụng các API một cách dễ dàng. Ngồi ra Eclipse còn hỗ trợ người lập trình dễ dàng thiết kế giao diện với thư viện Java Swing hay Java Awt.

* ***Ưu điểm của công cụ Eclipse***
* Nền tảng đa: có thể chạy trên cả Windows và Linux điều hành
* Không bị hạn chế về các nhà cung cấp công cụ, bao gồm tất cả các nhà cung cấp phần mềm độc lập (ISV)
* Hỗ trợ sử dụng nhiều công cụ lập trình
* Tạo lợi ích cho tích hợp các mạch công cụ bên trong và xuyên qua nhiều loại nội dung và các nhà cung cấp công cụ khác nhau.
* Hỗ trợ các công cụ thao tác lập trình ngôn ngữ như: [HTML](https://vn.got-it.ai/blog/ngon-ngu-html-la-gi-tim-hieu-cau-truc-html) , Java, C,….
* GUI môi trường hỗ trợ phát triển không dựa trên GUI.
* Tính biến phổ của ngôn ngữ Java (ngôn ngữ sử dụng để viết các công cụ).
* Tải nhanh hơn sử dụng SWT / Jface
* ***Nhược điểm của công cụ Eclipse***
* Cài đặt khá phức tạp, phần cứng và máy bộ nhớ
* Nhiều plugin đến quán nhất tính thiếu

2.4.1.2. Giới thiệu công cụ Visual Studio

Microsoft Visual Studio là một môi trường tích hợp (IDE) từ Microsoft. Microsoft Visual Studio còn được gọi là "Trình soạn thảo mã nhiều người sử dụng nhất thế giới ", được dùng để lập trình C++ và C# là chính. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web.

Từ khi ra đời đến nay, Visual Studio đã có rất nhiều các phiên bản sử dụng khác nhau. Điều đó, giúp cho người dùng có thể lựa chọn được phiên bản tương thích với dòng máy của mình cũng như cấu hình sử dụng phù hợp nhất.

Những điểm mạnh của Visual Studio:

-– Visual Studio hỗ trợ lập trình trên nhiều ngôn ngữ như: C/C++, [C#](https://www.w3schools.com/cs/), F#, Visual Basic, HTML, CSS, JavaScript.

– Là một công cụ hỗ trợ việc Debug một cách dễ dàng và mạnh mẽ như: Break Point, xem giá trị của biến trong quá trình chạy, hỗ trợ debug từng câu lệnh.

– Giao diện Visual Studio rất dễ sử dụng đối với người mới bắt đầu [lập trình](https://www.youtube.com/watch?v=So0luoTtl_o).

– Visual Studio hỗ trợ phát triển các ứng dụng: desktop MFC, Windows Form, Universal App, ứng dụng mobile Windows Phone 8/8.1, Windows 10, …

– Visual Studio hỗ trợ xây dựng ứng dụng một cách chuyên nghiệp bằng các công cụ kéo thả.

– Visual Studio được đông đảo lập trình viên trên thế giới sử dụng.

2.4.2. Hướng dẫn cài đặt và chạy chương trình demo đã cài đặt

2.4.2.1. Cài đặt Eclipse và chạy chương trình demo

- Cài đặt Eclipse

* Bước 1: Truy cập trang https https://www.oracle.com/ để tải JDK và công cụ Eclipse IDE về máy
* Bước 2: Mở file sau khi đã tải thành công
* Bước 3: Ấn theo hướng dẫn cài đặt của phần mềm
* Bước 4: Thiết lập biến môi trường
* Bước 5: Mở công cụ và hoàn tất cài đặt

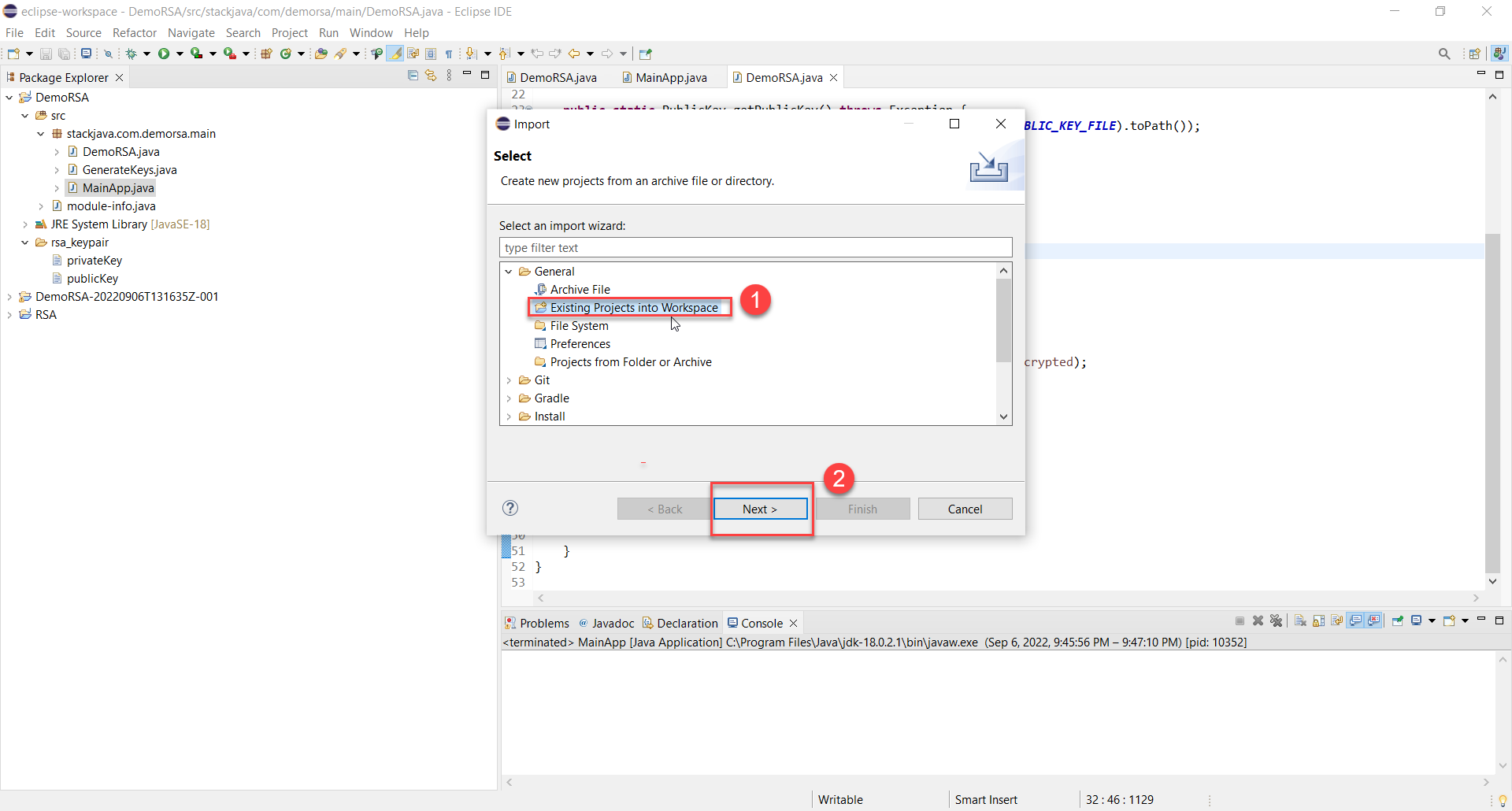
- Chạy chương trình demo

* **Bước 1**: Ở giao diện làm việc của Eclipse IDE, vào **File** chọn **Import**.

Graphical user interface, application

Description automatically generated

* **Bước 2**: Ở hộp thoại chọn vào **Existing Projects into Workspace** và chọn **Next**. Hộp thoại tiếp theo sẽ mở ra.



* **Bước 3**: Ở hộp thoại tiếp theo chọn ***Select root directory*** và trỏ đến thư mục project. Chọn ***Finish***.

Graphical user interface, application

Description automatically generated

* **Bước 4**: Thực hiện nhấn vào Run hoặc icon  trên thanh công cụ để thực hiện chạy chương trình

Graphical user interface, application

Description automatically generated

***\*\* Giao diện chương trình***

Graphical user interface, application, Word

Description automatically generated

2.4.2.2. Cài đặt Visual Studio Code và chạy chương trình demo Python

**Cài đặt Visual Studio Code và thiết lập môi trường:**

B1: Truy cập vào trang [Visual Studio Code - Code Editing. Redefined](https://code.visualstudio.com/), tải xuống và cài đặt VSCode theo hướng dẫn của chương trình.

B2: Cài đặt biến môi trường (Enviroment Path) cho VSCode

B3: Mở VSCode, cài đặt tiện ích mở rộng có tên Python.

B4: Restart VSCode, ở terminal của VSCode, gõ lệnh ‘pip install PyQt5’. Hoàn tất thiết lập môi trường chạy chương trình.

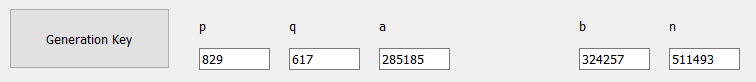
**Chạy chương trình demo**

* B1: Ở giao diện làm việc của VSCode, vào file chọn open folder rồi chọn folder cần chạy.
* B2: Click vào nút run trên thanh công cụ hoặc nhấp chuột phải vào bất cứ đâu ở cửa sổ editor và chọn ‘Run Python File in Terminal’
* B3: Lúc này chương trình demo bắt đầu chạy. Đây là giao diện chưong trình:

Graphical user interface

Description automatically generated

* B4: Để sử dụng chương trình mã hóa thì người dùng ấn nút Generation Key để chương trình tự động sinh ra giá trị cho p, q và b



* B5: Tiếp theo người dùng sẽ nhập thông tin cần mã hóa, hoặc nhấn open file rồi chọn file chứa thông điệp muốn mã hóa

**Cách 1: Nhập trực tiếp vào text box**

Graphical user interface, application

Description automatically generated with medium confidence

**Cách 2: Click vào nút Open File, chọn file \*.txt**

Graphical user interface, text, application

Description automatically generated

* B6: Click vào nút Encrypt ta được thông tin đã mã hoá

Graphical user interface, text, application

Description automatically generated

* B7: Click vào nút Send Ciphertext để chuyển bản mã sang phần giải mã hoặc click vào Save File để lưu bản mã
* **Cách 1: Click vào Send Ciphertext**

Graphical user interface, application

Description automatically generated

**Cách 2: Click vào Save File, lưu file dưới dạng .txt**

Graphical user interface, text, application, email

Description automatically generated

* B8: Click vào Receive Ciphertext hoặc Get File để nhận bản mã

Graphical user interface, application

Description automatically generated

* B9: Click vào Decrypt để thực hiện giải mã bản mã thành bản rõ như ban đầu

Graphical user interface, text, application

Description automatically generated

* B10: Click vào nút Clear để xóa toàn bộ thông tin trên cửa sổ chương trình, chương trình sẽ trở về ban đầu như khi ta khởi động chương trình.

2.4.2.4. Cài đặt Visual Studio và chạy chương trình demo C#

- Cài đặt Visual Studio

* B1: Truy cập trang <https://visualstudio.microsoft.com/> để download bộ cài Visual Studio.
* B2: Mở file sau khi đã tải thành công
* B3: Ấn theo hướng dẫn cài đặt của phần mềm
* B4: Thiết lập biến môi trường, cài đặt gói hỗ trợ C#
* B5: Mở công cụ và hoàn tất cài đặt

- Chạy chương trình demo

Demo giao diện:

Graphical user interface, application, Teams

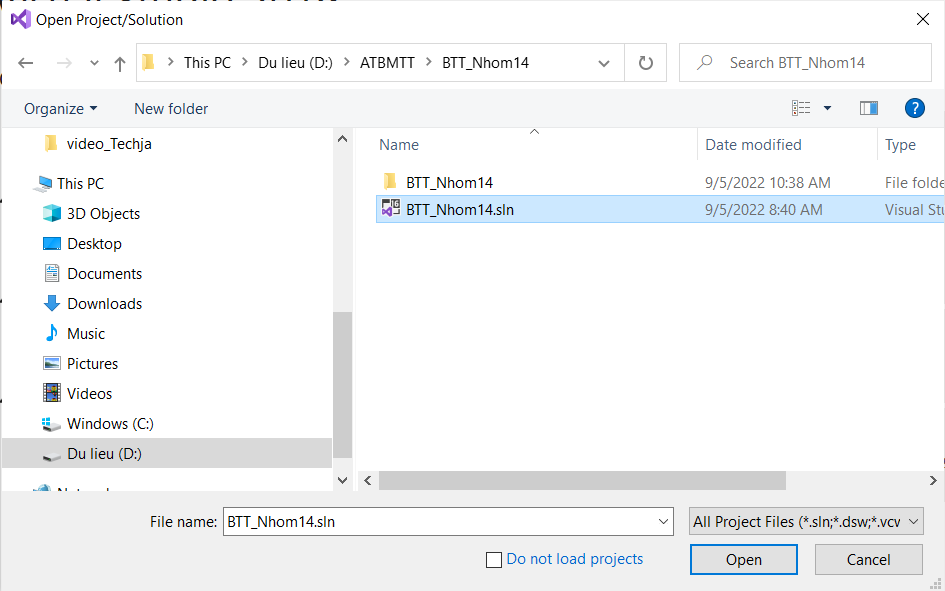
Description automatically generated

* B1: Ở giao diện làm việc của Visual Studio, chọn open project or solution rồi chọn project cần chạy.

Graphical user interface, application

Description automatically generated

* B2: Chọn project, chọn file .sln và nhấn **Open**



2

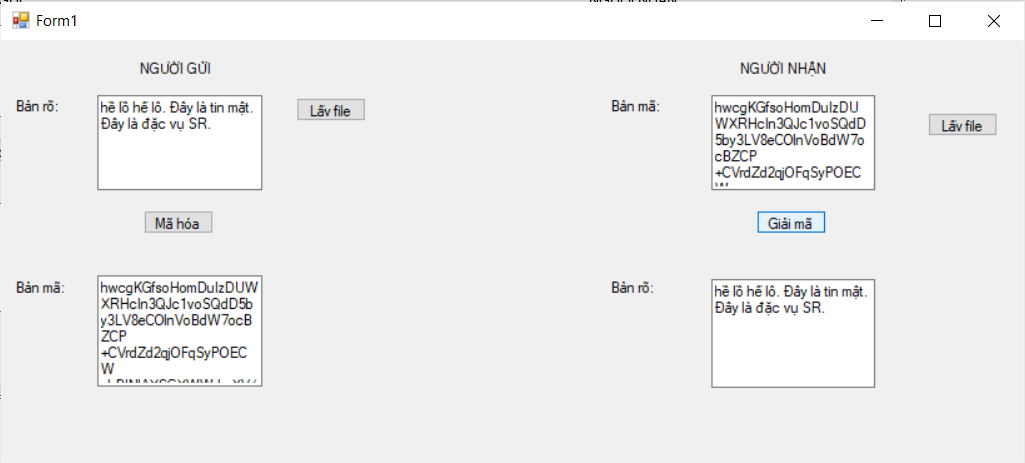
1

* B3: Nhấn “Start” để chạy chương trình

Graphical user interface, application

Description automatically generated

* B4: Thêm vào văn bản hoặc tệp văn bản, thực hiện chạy chương trình và kiểm tra kết quả:



2.5. Thực hiện bài toán

2.5.1. Phân công công việc

|  |  |
| --- | --- |
| **Tên sinh viên** | **Tên công việc** |
| Đào Thanh Tùng | - Tìm hiểu về quá trình tạo khóa  - Tìm hiểu về độ an toàn của mã hóa RSA  - Cài đặt các chức năng trên môi trường Python:   * Mã hóa * Giải mã |
| Bùi Thị Tươi | - Tìm hiểu về ưu nhược điểm của mã hóa RSA  - Tìm hiểu về quá trình mã hóa của RSA  - Cài đặt các chức năng trên môi trường C#:   * Mã hóa * Giải mã |
| Yến | - Tìm hiểu về nguyên lý hoạt động của RSA  - Tìm hiểu về các cách tấn công và phòng chống  - Cài đặt các chức năng trên môi trường Java:   * Mã hóa * Giải mã |

2.5.2. Đào Thanh Tùng - abc

* Tìm hiểu về quá trình tạo khóa
* Tìm hiểu về độ an toàn của hệ mật mã RSA
* Viết chương trình demo với ngôn ngữ python

2.5.3. Bùi Thị Tươi - abc

* Tìm hiểu về ưu, nhược điểm của hệ mật RSA
* Tìm hiểu về quá trình mã hóa
* Viết chương trình demo với ngôn ngữ C#

2.5.4. Yến - Tìm hiểu về nguyên lý hoạt động của RSA và cài đặt chương trình demo

* Tìm hiểu về nguyên lý hoạt động của RSA
* Viết chương trình demo với ngôn ngữ Java

CHƯƠNG 3. KIẾN THỨC LĨNH HỘI VÀ BÀI HỌC KINH NGHIỆM

3.1. Nội dung đã thực hiện

3.1.1. Các kiến thức đã học được thông qua thực hiện bài tập lớn

* Thuật toán mã hóa và giải mã RSA
* Thuật toán hình thành tham số và khóa
* Cơ sở lý thuyết xây dựng hệ mật mã RSA
* Quá trình tạo khóa
* Quá trình mã hóa
* Quá trình giải mã
* Tính đúng đắn của hệ mật mã RSA
* Ưu và nhược điểm của hệ mật mã RSA
* Độ an toàn của hệ mật mã RSA

3.1.2. Các kỹ năng đã học được thông qua thực hiện bài tập lớn

* Đánh giá được vai trò của bảo mật thông tin, các cơ chế, chính sách bảo mật, các kiểu tấn công và phương pháp phòng chống.
* Phân tích được các kỹ thuật sử dụng để mã hóa và xác thực thông tin.
* Hiểu và áp dụng các thuật toán liên quan đến hệ mã hóa RSA như (thuật toán sinh khóa, thuật toán mã hóa, thuật toán giải mã cùng vời các thuật toán liên quan như thuật toán nghịch đảo của phép nhân modulo hay thuật toán bình phưong và nhân trong modulo) vào việc mã hóa và giải mã để giải quyết bài toán có tính ứng dụng vào thực tiễn.
* Tổ chức được hoạt động nhóm.
* Áp dụng được các phương pháp thuyết trình hiệu quả trong công việc.

3.1.3. Những bài học kinh nghiệm được rút ra sau khi kết thúc bài tập lớn

* Các tài liệu quan trọng và chính thống liên quan đến hệ mật mã RSA chủ yếu được viết bằng tiếng anh nên các thành viên trong nhóm phải trau rèn khả năng ngoại ngữ để có thể đọc hiểu được một số tài liệu liên quan đến hệ mã hóa RSA và công cụ, ngôn ngữ lập trình.
* Trong quá trình thực hiện viết chương trình các thành viên trong nhóm gặp rất nhiều vấn đề về việc mã hóa và giải mã thông qua file, đặc biệt là lấy bản rõ bằng cách open file docx. Do đó các thành viên trong nhóm đã đúc kết được kinh nghiệm về việc xử lý file và hiểu được trách nhiệm phải quan tâm nhiều hơn đến dữ liệu người dùng nhập vào, người dùng có thể nhập bản rõ vào phần mềm bằng nhiều cách khác nhau và phải xử lý để các các dữ liệu đó hoạt động thật mượt mà và trơn tru.

**3.2. Hướng phát triển**

**3.2.1. Tính khả thi của chủ đề nghiên cứu**

Chủ đề nghiên cứu của nhóm chúng em khá phù hợp với thời gian được cho phép để hoàn thiện bài tập lớn, bên cạnh đó các thuật toán và mã hóa đều đã có sẵn được thử nghiệm bởi các nhà nghiên cứu bảo mật nên trong thời gian nhóm nghiên cứu, nhóm nhận thấy cần phải thực sự hiểu rõ về hệ mật mã RSA và có kỹ thuật lập trình ở mức khá là đã có thể hoàn thiện đề tài nghiên cứu

**3.2.2. Những thuận lợi, khó khăn trong quá trình nghiên cứu**

**- Thuận lợi:**

* Do đã có kiến thức lập trình từ năm học trước nên vấn đề về phân chia ngôn ngữ lập trình cho các thành viên trong nhóm khá dễ dàng, và việc chuyển từ ngôn ngữ sở trường sang các ngôn ngữ khác đều không gặp nhiều vấn đề
* Có thể đọc hiểu được tài liệu tiếng anh nên có thể dễ dàng tiếp cận các nguồn tài liệu chính thống
* Các thuật toán đã có sẵn, chỉ cần áp dụng một chút kỹ thuật xử lý về Form, File là đã có thể hoàn thành bài toán của đề tài
* Các thành viên trong nhóm hòa đồng, cởi mở, tương tác với các thành viên khác trong nhóm khá sôi nổi nên các công việc liên quan đến cả nhóm thường diễn ra khá suôn sẻ

**- Khó khăn:**

* Công đoạn thiết giao diện phần mềm mã hóa chưa được bắt mắt do chưa có nhiều kinh nghiệm trong kỹ thuật xử lý giao diện
* Ở phần xử lý về File Docx sử dụng để lấy bản rõ khá mới mẻ với các thành viên trong nhóm nên giai đoạn hoàn thiện chức năng tốn khá nhiều thời gian

**3.2.3. Hướng phát triển và mở rộng của đề tài**

Nhận thấy việc sử dụng hệ mật mã RSA vô cùng hữu ích trong việc bảo mật thông tin, tuy nhiên sản phẩm của nhóm chỉ có thể mã hóa 1 thông điệp 1 lúc và chỉ 1 thông điệp từ 1 người gửi đến cho 1 người nên khá bất tiện trong nhiều trường hợp. Cho nên nhóm có đề xuất hướng phát triển là sẽ xây dựng thêm chức năng nhập file excel để có thể nhập nhiều bản rõ cùng một lúc và mã hóa cùng lúc. Điều này sẽ rất thuận tiện trong nhiều trường hợp trong thực tế.

Hơn nữa, với triển vọng của việc phát triển máy tính lượng tử, trong tương lai mã hóa RSA sẽ cần có những cải tiến vượt trội. Nhóm cũng có thể tìm đọc và mở rộng kiến thức liên quan đến những vấn đề của mã hóa RSA khi máy tính lượng tử được phổ cập. Từ đó có thể có được cái nhìn sâu và tổng quan hơn về mã hóa nói riêng cũng như ngành mật mã nói chung.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] [RSA (cryptosystem) - Wikipedia](https://en.wikipedia.org/wiki/RSA_(cryptosystem)) - Wikipedia

[2] RSA Security's official guide to cryptography by Steve Burnett

[3] Cryptanalysis of RSA and Its Variants by M. Jason Hinek

K. Giuliani, “Attacks on the Elliptic Curve Discrete

Logarithm Problem”, Master of Mathematics,

University of Waterloo, Ontario, Canada, 1999.

[2] J. H. Silverman, “The Arithmetic of Elliptic

Curves”,GTM 106, Springer-Verlag, 1986.

[3] A. Menezes, T. Okamoto and S. Vanstone,

“Reducing Elliptic CurveLogarithm to Logarithms

in a Finite Field”, IEEE Transactions on

Information Theory, vol. 39, pp. 1639-1646, 1993.

K. Giuliani, “Attacks on the Elliptic Curve Discrete

Logarithm Problem”, Master of Mathematics,

University of Waterloo, Ontario, Canada, 1999.

[2] J. H. Silverman, “The Arithmetic of Elliptic

Curves”,GTM 106, Springer-Verlag, 1986.

[3] A. Menezes, T. Okamoto and S. Vanstone,

“Reducing Elliptic CurveLogarithm to Logarithms

in a Finite Field”, IEEE Transactions on

Information Theory, vol. 39, pp. 1639-1646, 1993.

K. Giuliani, “Attacks on the Elliptic Curve Discrete

Logarithm Problem”, Master of Mathematics,

University of Waterloo, Ontario, Canada, 1999.

[2] J. H. Silverman, “The Arithmetic of Elliptic

Curves”,GTM 106, Springer-Verlag, 1986.

[3] A. Menezes, T. Okamoto and S. Vanstone,

“Reducing Elliptic CurveLogarithm to Logarithms

in a Finite Field”, IEEE Transactions on

Information Theory, vol. 39, pp. 1639-1646, 1993.

K. Giuliani, “Attacks on the Elliptic Curve Discrete

Logarithm Problem”, Master of Mathematics,

University of Waterloo, Ontario, Canada, 1999.

[2] J. H. Silverman, “The Arithmetic of Elliptic

Curves”,GTM 106, Springer-Verlag, 1986.

[3] A. Menezes, T. Okamoto and S. Vanstone,

“Reducing Elliptic CurveLogarithm to Logarithms

in a Finite Field”, IEEE Transactions on

Information Theory, vol. 39, pp. 1639-1646, 1993.

[4] RSA and Public-Key Cryptography by Richard A. Mollin