**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

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



**BÁO CÁO BÀI TẬP LỚN**

**Học phần: An toàn và bảo mật thông tin**

**RSA và sản phẩm demo ứng dụng mã hóa  
trong truyền tin**

**Giảng viên hướng dẫn: TS.Lê Thị Anh**

**Mã lớp: 20231IT6001001**

**Sinh viên thực hiện: Nhóm 16**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | Mã sinh viên |
| 1 | Phùng Thị Trang Anh | 2021602928 |
| 2 | Nguyễn Minh Chiến | 2021606723 |
| 3 | Nguyễn Trọng Đạt | 2021600567 |
| 4 | Vũ Đức Dũng | 2021600819 |
| 5 | Đồng Phương Duyên | 2021601732 |

***Hà Nội, Tháng 12 – 2023***

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

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



**BÁO CÁO BÀI TẬP LỚN**

**Học phần: An toàn và bảo mật thông tin**

**RSA và sản phẩm demo ứng dụng mã hóa  
trong truyền tin**

**Giảng viên hướng dẫn: TS.Lê Thị Anh**

**Mã lớp: 20231IT6001001**

**Sinh viên thực hiện: Nhóm 16**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | Mã sinh viên |
| 1 | Phùng Thị Trang Anh | 2021602928 |
| 2 | Nguyễn Minh Chiến | 2021606723 |
| 3 | Nguyễn Trọng Đạt | 2021600567 |
| 4 | Vũ Đức Dũng | 2021600819 |
| 5 | Đồng Phương Duyên | 2021601732 |

***Hà Nội, Tháng 12 – 2023***

# MỤC LỤC

[MỤC LỤC 2](#_Toc154275591)

[DANH MỤC HÌNH ẢNH 3](#_Toc154275592)

[LỜI CẢM ƠN 4](#_Toc154275593)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 5](#_Toc154275594)

[1.1. Tổng quan đề tài 5](#_Toc154275595)

[1.2. Lý do chọn đề tài 6](#_Toc154275596)

[1.3. Hệ thống mật mã RSA 7](#_Toc154275597)

[1.3.1. Khái quát về hệ mật RSA 7](#_Toc154275598)

[1.3.2. Lĩnh vực hoạt động 8](#_Toc154275599)

[1.3.3. Các thuật toán hỗ trợ 10](#_Toc154275600)

[1.3.4. Cách thức hoạt động 12](#_Toc154275601)

[1.3.5. Mức độ an toàn 13](#_Toc154275602)

[1.4. Ngôn ngữ và công cụ sử dụng 14](#_Toc154275603)

[1.4.1. Ngôn ngữ python 14](#_Toc154275604)

[1.4.2. Môi trường Visual Studio Code 15](#_Toc154275605)

[1.4.3. Công cụ Qt Designer 16](#_Toc154275606)

[CHƯƠNG 2. THIẾT KẾ CHI TIẾT 18](#_Toc154275607)

[2.1. Mô tả về ứng dụng của RSA 18](#_Toc154275608)

[2.2. Cài đặt thuật toán RSA 19](#_Toc154275609)

[2.2.1. Cài đặt thuật toán hỗ trợ 19](#_Toc154275610)

[2.2.2. Cài đặt thuật toán tạo khóa, mã hóa và giải mã 20](#_Toc154275611)

[2.3. Thiết kế giao diện 22](#_Toc154275612)

[2.4. Xử lý các chức năng cho widget giao diện: 25](#_Toc154275613)

[CHƯƠNG 3. CHƯƠNG TRÌNH VÀ DEMO 28](#_Toc154275614)

[3.1. Giao diện chương trình 28](#_Toc154275615)

[3.2. Hướng dẫn và chạy chương trình 29](#_Toc154275616)

[CHƯƠNG 4. KẾT LUẬN VÀ KIẾN NGHỊ 31](#_Toc154275617)

[4.1. Những kiến thức và kỹ năng đã học được 31](#_Toc154275618)

[4.1.1. Kiến thức 31](#_Toc154275619)

[4.1.2. Kỹ năng 32](#_Toc154275620)

[4.2. Hướng phát triển 33](#_Toc154275621)

# DANH MỤC HÌNH ẢNH

[Hình 1.1: Mô tả quá trình truyền tin 5](#_Toc154275622)

[Hình 1.2: Thuật toán Euclid tìm USCLN 10](#_Toc154275623)

[Hình 1.3: Tính nghịch đảo modulo bằng Euclid mở rộng 11](#_Toc154275624)

[Hình 1.4: Tính bình phương và nhân 12](#_Toc154275625)

[Hình 1.5: Các phiên bản và hệ điều hành phù hợp với máy 16](#_Toc154275626)

[Hình 2.1: Thuật toán hỗ trợ 19](#_Toc154275627)

[Hình 2.2: Cài đặt thuật toán tạo khóa, mã hóa và giải mã 20](#_Toc154275628)

[Hình 2.3: Cài đặt các thư viện cần thiết 22](#_Toc154275629)

[Hình 2.4: Thiết lập giao diện bằng Python 23](#_Toc154275630)

[Hình 2.5: Dịch các chuỗi văn bản bằng Python 24](#_Toc154275631)

[Hình 2.6: Xử lý các chức năng cho widget (a) 26](#_Toc154275632)

[Hình 2.7: Xử lý các chức năng cho widget (b) 26](#_Toc154275633)

[Hình 3.1: Chương trình demo 28](#_Toc154275634)

[Hình 3.2: Minh họa bước 1 29](file:///C:\Users\ADMIN\Documents\Zalo%20Received%20Files\Bài%20tập%20lớn%20ATBM_Nhóm%2016%20(1).docx#_Toc154275635)

[Hình 3.3: Minh họa bước 2 29](#_Toc154275636)

[Hình 3.4: Minh họa bước 3 30](#_Toc154275637)

[Hình 3.5: Minh họa bước 4 30](file:///C:\Users\ADMIN\Documents\Zalo%20Received%20Files\Bài%20tập%20lớn%20ATBM_Nhóm%2016%20(1).docx#_Toc154275638)

# LỜI CẢM ƠN

Nhóm chúng em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đến các thầy cô của trường Đại học Công nghiệp Hà Nội, đặc biệt là các thầy cô công nghệ thông tin và kế toán - Kiểm toán của trường đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thầy cô nên đề tài nghiên cứu của chúng em mới có thể hoàn thiện tốt đẹp.

Đặc biệt, chúng em xin chân thành cảm ơn Cô Lê Thị Anh – người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn chúng em hoàn thành tốt bài báo cáo trong thời gian qua. Cô đã không ngừng động viên, góp ý và chỉ ra những điểm cần cải thiện cho bài báo cáo của chúng em. Cô cũng đã giúp chúng em hiểu rõ hơn về lý thuyết và ứng dụng của thuật toán RSA trong truyền tin.

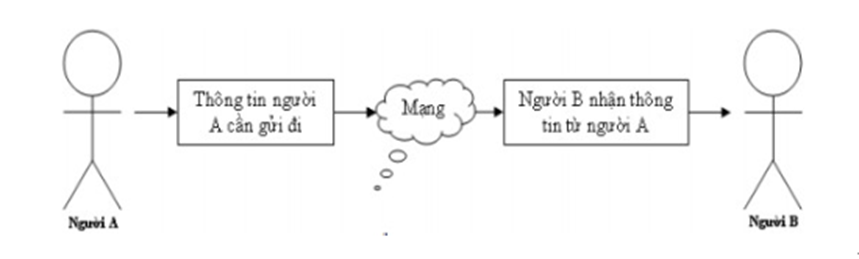
Bài báo cáo được thực hiện trong khoảng thời gian 5 tuần. Bài báo cáo nghiên cứu này cũng được hoàn thiện dựa trên cơ sở tham khảo các sách giáo trình, tạp chí của nhiều tác giả tại các trường đại học, các tổ chức nghiên cứu… cũng như học hỏi kinh nghiệm từ các nghiên cứu liên quan. Là một ngành học không chuyên về IT, chúng em còn nhiều bỡ ngỡ nên không tránh khỏi thiếu sót trong cách hiểu và lỗi trình bày, nhóm chúng em rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô để kiến thức của chúng em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao ý thức của mình và sẽ hoàn thành tốt bài báo cáo của môn học sắp tới.

*Chúng em xin chân thành cảm ơn!*

# CƠ SỞ LÝ THUYẾT

## Tổng quan đề tài

Trong mọi lĩnh vực kinh tế, chính trị, xã hội, quân sự,… luôn có nhu cầy trao đổi thông tin giữa các cá nhân, các công ty, tổ chức, hoặc giữa các quốc gia với nhau. Ngày nay, với sự phát triển của công nghệ thông tin đặc biệt là mạng internet thì việc truyền tải thông tin đã dễ dàng và nhanh chóng hơn.



Hình .: Mô tả quá trình truyền tin

Bảo mật thông tin luôn là vấn đề quan trọng hàng đầu trong các lĩnh vực tình báo, quân sự, ngoại giao, và đây cũng là một vấn đề đã được nghiên cứu hàng nghìn năm nay. Bảo mật thông tin là duy trì tính bảo mật, tính toàn vẹn và tính sẵn sàng của thông tin. Bảo mật nghĩa là đảm bảo thông tin chỉ được tiếp cận bởi những người được cấp quyền tương ứng.

Tính toàn vẹn là bảo vệ sự chính xác, hoàn chỉnh của thông tin và thông tin chỉ được thay đổi bởi những người được cấp quyền.

Tính sẵn sàng của thông tin là những người được quyền sử dụng có thể truy xuất thông tin khi họ cần.

Vấn đề bảo mật đang được nhiều người tập trung nghiên cứu và tìm mọi giải pháp để đảm bảo an toàn, an ninh cho hệ thống phần mềm, đặc biệt là các hệ thống thông tin trên mạng Internet cho phép mọi người truy cập, khai thác và chia sẻ thông tin. Mặt khác nó cũng là nguy cơ chính dẫn đến thông tin bị rò rỉ hoặc bị phá hoại. Lúc này việc bảo mật an toàn dữ liệu là vấn đề thời sự, là một chủ đề rộng có liên quan đến nhiều lĩnh vực và trong thực tế có nhiều phương pháp được thực hiện để đảm bảo dữ liệu.

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

## Lý do chọn đề tài

Bảo mật thông tin đóng vai trò cực kỳ quan trọng và thiết yếu trong thế giới kỹ thuật số ngày nay. Trong một xã hội mạng liên kết toàn cầu, thông tin là nguồn lực quý giá, và bảo mật thông tin là yếu tố quyết định đến sự thành công của các tổ chức, doanh nghiệp và cá nhân.

Một trong những khía cạnh quan trọng nhất của bảo mật thông tin là đảm bảo tính toàn vẹn, tức là đảm bảo rằng thông tin không bị sửa đổi hoặc biến đổi một cách trái phép. Khi thông tin không được bảo mật đúng cách, nó có thể bị thay đổi, làm giả, hoặc bị tấn công để tạo ra hậu quả nghiêm trọng. Điều này có thể dẫn đến những hậu quả đáng tiếc như mất mát tài sản, tiêu cực hóa hình ảnh, mất uy tín và thiệt hại cho doanh nghiệp và cá nhân.

Bảo mật thông tin cũng đóng vai trò quan trọng trong việc bảo vệ quyền riêng tư của cá nhân. Trong thời đại mạng xã hội và các dịch vụ trực tuyến, thông tin cá nhân được thu thập và sử dụng phổ biến. Bảo mật thông tin đảm bảo rằng thông tin cá nhân không bị tiếp cận trái phép hoặc lộ ra công chúng. Điều này đặc biệt quan trọng trong việc bảo vệ quyền riêng tư, giữ cho cá nhân không bị tác động bởi việc xâm phạm và lạm dụng thông tin cá nhân.

Sự ra đời mạnh mẽ của các hệ mật mã như RSA, Elgammal,… đã góp phần đảm bảo cho việc bảo mật thông tin trở lên thuận lợi dễ dàng hơn, đồng thời chúng cũng góp phần thúc đẩy ngành an toàn và bảo mật thông tin phát triển mạnh mẽ.

Nhóm chúng em nghiên cứu đề tài hệ thống mật mã RSA vì muốn tìm hiểu thêm về kiến thức cơ bản và ứng dụng của mật mã học trong việc bảo vệ thông tin và dữ liệu. Chúng em luôn muốn tìm hiểu và khám phá những lĩnh vực mới, đặc biệt là trong lĩnh vực công nghệ thông tin và an ninh mạng.

Với sự phát triển của công nghệ và Internet, thông tin và dữ liệu cá nhân trở nên quan trọng hơn bao giờ hết. Việc bảo mật thông tin và dữ liệu là vô cùng cần thiết, đặc biệt khi việc trao đổi thông tin và giao dịch trực tuyến đang ngày càng trở nên phổ biến. Hệ thống mật mã RSA là một trong những hệ thống mã hóa công khai phổ biến và quan trọng nhất, do đó chúng em muốn nghiên cứu về nó để có thể áp dụng trong thực tế và cải thiện kỹ năng của mình.

Ngoài ra, chúng em tin rằng việc tìm hiểu về hệ thống mật mã RSA sẽ giúp chúng em có thêm kiến thức liên quan đến bảo mật thông tin, an ninh mạng và kỹ thuật lập trình. Hi vọng rằng sau khi hoàn thành đề tài này, chúng em sẽ có một tầm nhìn rõ ràng hơn về công nghệ mã hóa và bảo mật thông tin, cũng như có thể áp dụng kiến thức này vào các tình huống thực tế trong tương lai.

## Hệ thống mật mã RSA

### Khái quát về hệ mật RSA

Một hệ thống mật mã là một hệ thống gồm 5 thành phần (P, C, K, E, D) như sau:

* 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 hoá có thể (hay còn gọi là không gian các hàm mã hoá).

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

Hệ mật mã gồm 2 loại:

* Hệ mật mã đổi 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 thông tin. Do đó khóa phải được giữ bí mật tuyệt đối. Một sổ thuật toán nổi tiếng trong mã hóa đổi xứng là: DES, Triple DES(3DES), RC4, AES,...
* Hệ mật mã bất đổi 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 sổ thuật toán mã hóa công khai nổi tiếng: Diffle- Hellman, Elgamal, RSA, Rabin,...

Hệ thống mật mã RSA được đề xuất vào năm 1977 bởi ba nhà toán học Ronald Rivest, Adi Shamir và Leonard Adleman. Đây là một hệ thống mã hóa công khai, tức là sử dụng hai khóa khác nhau cho quá trình mã hóa và giải mã. Khóa công khai được chia sẻ và bảo mật với bất kỳ ai, trong khi đó khóa bí mật chỉ được giữ bởi chủ sở hữu. Mật mã RSA dựa trên bài toán tháp nguyên tố, trong đó ta cần phân tích số một số tự nhiên lớn thành các thừa số nguyên tố riêng lẻ. Phương pháp này đã được biết đến từ lâu, nhưng lại được ứng dụng vào mật mã học bởi RSA.

Hệ thống mật mã RSA là một dạng mã hoá chữ thập (block cipher), có khả năng giải mã nhanh hơn so với các hệ thống mã hoá cùng loại. RSA cũng không yêu cầu sự tương tác giữa hai bên trước và sau khi truyền tải thông tin. Tuy nhiên, mật mã RSA chỉ phù hợp với việc mã hóa các tập tin có kích thước nhỏ hơn, vì sự an toàn của hệ thống phụ thuộc vào độ dài các khóa mã hóa. Hệ mã hóa này cũng là biến thể của hệ thống mã hóa tương tự, nhưng có hiệu quả và an toàn hơn. So với hệ thống mã hóa công khai khác, như ElGamal, RSA có tốc độ mã hóa và giải mã nhanh hơn, đặc biệt là với các khóa mã hóa dài. Tuy nhiên, việc khôi phục lại khóa bí mật trong mật mã RSA là một bài toán phức tạp hơn so với ElGammal.

### Lĩnh vực hoạt động

Mã hoá và giải mã: Hệ mật RSA đóng vai trò quan trọng trong việc bảo vệ tính an toàn của thông điệp khi mã hóa chúng và sau đó giải mã. Điều này tạo nền tảng chắc chắn cho việc truyền tải dữ liệu an toàn qua các kênh mạng, cung cấp giải pháp mật mã đáng tin cậy để ngăn chặn việc truy cập trái phép và bảo vệ tính toàn vẹn của thông tin quan trọng.

* Chữ ký số

Hệ mật RSA thường được ứng dụng mạnh mẽ trong việc tạo chữ ký số, có tác dụng không chỉ đảm bảo tính xác thực mà còn khẳng định nguồn gốc của thông điệp một cách mạnh mẽ. Sự đồng thuận của nó không chỉ giúp xác nhận tính toàn vẹn của thông điệp mà còn tạo ra một lớp bảo vệ chắc chắn, bảo vệ tính nhất quán và đáng tin cậy của thông tin trong quá trình truyền tải và giao tiếp điện tử.

* Bảo vệ dữ liệu truyền tải

Hệ mật RSA đóng một vai trò quan trọng trong việc bảo vệ dữ liệu truyền tải trên Internet, đặc biệt là trong các giao thức an toàn như HTTPS. Việc tích hợp RSA vào các hệ thống này không chỉ đảm bảo mức độ bảo mật cao mà còn thúc đẩy khả năng bảo vệ thông tin truyền tải, cung cấp một tầng an ninh đáng tin cậy để ngăn chặn tiềm ẩn của sự đánh cắp dữ liệu và đảm bảo an toàn cho giao tiếp trực tuyến.

* Quản lý khóa

Hệ mật RSA đóng một tầm quan trọng trong quản lý khóa cho các hệ thống mật mã, đóng góp tích cực vào việc bảo vệ thông tin mật. Vai trò này không chỉ giới hạn trong việc cung cấp các khóa an toàn mà còn mở rộng đến khả năng tạo, lưu trữ và quản lý khóa một cách hiệu quả. RSA không chỉ là một phương tiện mật mã mà còn là một cơ sở đáng tin cậy để bảo vệ quản lý khóa, đóng góp đáng kể vào sự an toàn của thông tin quan trọng trong các hệ thống bảo mật.

* Chứng thực

Hệ mật RSA đóng một vai trò quan trọng và không thể phụ qua trong quá trình chứng thực người dùng và thiết bị, đặc biệt là trong các hệ thống xác thực hai yếu tố. Việc tích hợp RSA vào quá trình chứng thực không chỉ cung cấp một cơ chế an toàn và đáng tin cậy mà còn đặt nền móng cho khả năng xác nhận đối tượng và thiết bị một cách mạnh mẽ. Điều này không chỉ làm tăng cường sự bảo mật mà còn giúp xây dựng một hệ thống chứng thực đa chiều, đảm bảo tính xác thực và quản lý truy cập hiệu quả trong môi trường an ninh cao.

* Bảo vệ email và tài khoản

Hệ mật RSA thường được hội nhập sâu rộng trong các hệ thống email và tài khoản, đóng vai trò quan trọng trong việc đảm bảo an toàn cho dữ liệu cá nhân và quá trình giao tiếp. Việc tích hợp RSA không chỉ mang lại một tầng bảo mật mạnh mẽ mà còn đề cao khả năng bảo vệ thông tin cá nhân và giao tiếp trực tuyến. Điều này giúp tăng cường sự tin cậy của hệ thống, bảo vệ thông tin cá nhân khỏi rủi ro và đóng góp vào việc xây dựng môi trường truyền thông an toàn và bảo mật.

* Tính toàn vẹn dữ liệu

Hệ mật RSA là một công cụ quan trọng được áp dụng để đảm bảo không chỉ tính toàn vẹn mà còn tính bảo mật của dữ liệu, đặc biệt là trong bối cảnh của các hệ thống lưu trữ trực tuyến và nền tảng chia sẻ tệp tin. Việc tích hợp RSA không chỉ mang lại một lớp bảo mật vững chắc mà còn nâng cao khả năng đối mặt với thách thức của việc duy trì sự toàn vẹn và quản lý an toàn cho thông tin trực tuyến. Điều này giúp tạo ra một môi trường lưu trữ và chia sẻ dữ liệu an toàn, đồng thời đảm bảo rằng thông tin quan trọng không bị thay đổi hay xâm phạm một cách không mong muốn.

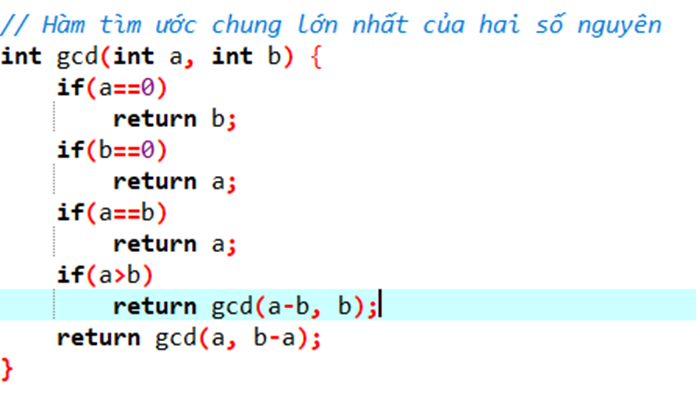
### Các thuật toán hỗ trợ

* Thuật toán Euclid

Thuật toán Euclid là một thuật toán cổ điển trong toán học được đặt theo tên nhà toán học Euclid, sống vào thế kỷ thứ 3 trước Công nguyên. Thuật toán Euclid được sử dụng để tìm ước chung lớn nhất (GCD - Greatest Common Divisor) của hai số nguyên dương. Ý tưởng chính của thuật toán Euclid là dựa trên việc áp dụng tính chất cơ bản của ước số chung. Theo đó, ước chung lớn nhất của hai số nguyên dương a và b cũng chính là ước chung lớn nhất của số b và phần dư khi a được chia cho b (hay a % b). Thuật toán Euclid được thực hiện các bước như sau:

* Bước 1: Bắt đầu với 2 số nguyên dương a và b. Giả sử: a>b
* Bước 2: Tính phần dư r khi lấy a chia cho b: r=a%b
* Bước 3: Gán a=b và b=r. Quay lại bước 2 cho đến khi b=0
* Bước 4: Khi bằng 0, UCSLN của a và b chính là a

Mô tả các bước bằng ngôn ngữ lập trình như sau:

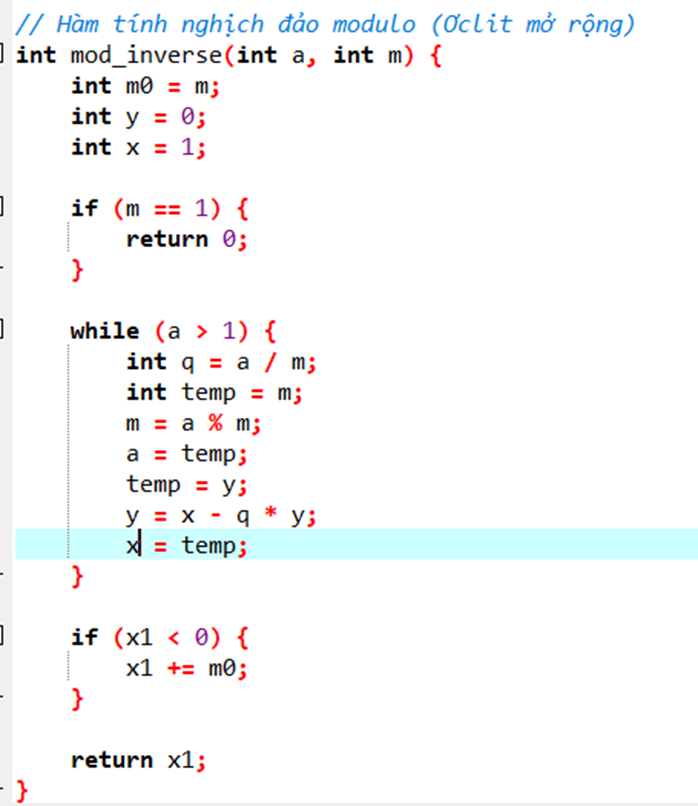


Hình .: Thuật toán Euclid tìm USCLN

* Thuật toán Euclid mở rộng

Thuật toán Euclid mở rộng, còn được gọi là thuật toán Euclid mở rộng dùng để tìm nghịch đảo modulo, là một biến thể của thuật toán Euclid cổ điển. Nó được sử dụng để tìm nghịch đảo của một số trong việc thực hiện các phép toán modulo.

Để hiểu thuật toán Euclid mở rộng, chúng ta cần nắm vững thuật toán Euclid cổ điển. Thuật toán Euclid mở rộng mở rộng chức năng của thuật toán Euclid cổ điển bằng cách tính toán UCLN của hai số nguyên a và b cùng với các hệ số Bézout (x và y) liên quan đến chúng. Cụ thể, nó tính toán USCLN và hai số nguyên x và y sao cho: ax + by = USCLN(a, b). Mô tả cài đặt thuật toán:



Hình .: Tính nghịch đảo modulo bằng Euclid mở rộng

Thuật toán Euclid và Euclid mở rộng được sử dụng rộng rãi trong các lĩnh vực như mật mã học, toán học, khoa học máy tính và các lĩnh vực liên quan đến số học. Nó cung cấp một phương pháp đơn giản và hiệu quả để tìm ước chung lớn nhất của hai số nguyên dương và nghịch đảo modulo.

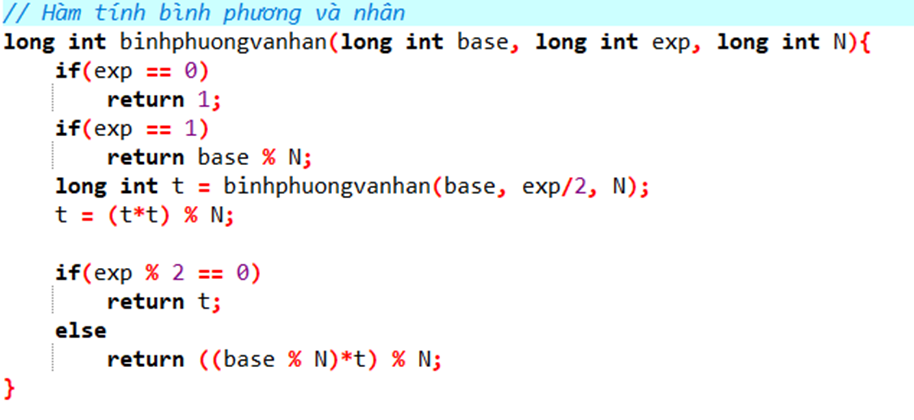
* Thuật toán bình phương và nhân

Thuật toán bình phương và nhân (Square and Multiply algorithm) là một phương pháp hiệu quả để tính lũy thừa của một số nguyên với một số nguyên dương. Đặc biệt, nó thường được sử dụng trong các hệ thống mật mã học, nơi tính toán các lũy thừa lớn là một phần quan trọng trong quá trình mã hóa và giải mã.

Thuật toán bình phương và nhân thực hiện việc tính toán lũy thừa bằng cách sử dụng các phép toán bình phương và nhân. Ý tưởng chính của thuật toán là biểu diễn số mũ dưới dạng nhị phân, sau đó lặp qua các bit của số mũ để tính toán kết quả. Cụ thể, các bước của thuật toán bình phương và nhân như sau:

* Bước 1: Đầu tiên, biểu diễn số mũ dưới dạng nhị phân.
* Bước 2: Khởi tạo một biến kết quả (result) với giá trị bằng 1.
* Bước 3: Lặp qua các bit của số mũ từ bit thứ cao đến bit thấp:
* Bước 4: Nếu bit hiện tại là 0, thực hiện phép toán bình phương cho kết quả: result = result \* result.
* Bước 5: Nếu bit hiện tại là 1, thực hiện phép toán nhân cho kết quả: result = result \* base.

Sau khi lặp qua tất cả các bit của số mũ, kết quả cuối cùng là giá trị của biến result.



Hình .: Tính bình phương và nhân

Thuật toán bình phương và nhân giúp tối ưu hóa quá trình tính toán lũy thừa, đặc biệt là khi số mũ rất lớn. Nó có ứng dụng quan trọng trong các thuật toán mật mã học như RSA và Diffie-Hellman, nơi tính toán các lũy thừa lớn là một yếu tố quan trọng để bảo vệ thông tin.

### Cách thức hoạt động

* Tạo khóa

Khóa công khai: Kpublic = {e,n}

Khóa cá nhân: Kprivate = {d,p,q}

Để tạo cặp khóa thực hiện các bước sau:

* Bước 1: Chọn 2 số nguyên tố lớn: p và q, với p # q và p, q >120 chữ số. Lựa chọn ngẫu nhiên và độc lập.
* Bước 2: Tính số n = p.q
* Bước 3: Tính số φ(n) = (p-1)\*(q-1) (hàm phi Euler)
* Bước 4: Chọn số e sao cho 1 < e < φ(n) và là số nguyên tố cùng nhau với φ(n) (tức là: gcd(e, φ(n)) = 1, để đảm bảo e-1 mod φ(n) tồn tại duy nhất)
* Bước 5: Tính số d = e-1 mod φ(N) Khi đó, ta có khóa là các bộ số:
* Mã hóa RSA

Người nhận sử dụng khóa công khai () để mã hóa.

* Bước 1: Người nhận sử dụng khóa để mã hóa bản rõ x.
* Bước 2: Tính (sử dụng thuật toán bình phương và nhân), trong đó:

x: Bản rõ

b: Thuộc khóa Kpublic

n: n= p\*q

Sau hai bước trên ta thu được bản mã y.

* Giải mã RSA

Người nhận sử dụng khóa cá nhân () để giải mã.

* Bước 1: Người nhận sử dụng khóa để giải mã bản mã y.
* Bước 2: Tính (sử dụng thuật toán bình phương và nhân), trong đó:

y: Bản mã

a: Thuộc khóa Kprivate

n: n= p\*q

Sau hai bước trên ta thu được bản rõ ban đầu.

### Mức độ an toàn

Mã hóa RSA là một hệ mã hóa bất đối xứng, có nghĩa là nó sử dụng hai khóa khác nhau để mã hóa và giải mã dữ liệu. Khóa công khai được dùng để mã hóa dữ liệu và khóa bí mật được dùng để giải mã dữ liệu. Mã hóa RSA có thể đảm bảo tính bảo mật, xác thực và toàn vẹn của dữ liệu truyền đi. Tuy nhiên, mã hóa RSA trong truyền tin có an toàn không phụ thuộc vào nhiều yếu tố, như độ dài khóa, phương pháp sinh khóa, phương pháp mã hóa và giải mã, cũng như các kỹ thuật tấn công có thể áp dụng.

*Ưu điểm của hệ mật RSA:*

* Tính bảo mật cao: Bởi sử dụng các số nguyên tố lớn, khó bị phân tích thành thừa số nguyên tố. Nếu có đủ độ dài khóa, mã hóa RSA có thể chống lại các kỹ thuật tấn công thông thường sử dụng máy tính cá nhân.
* Có thể kết hợp với các hệ mã hóa đối xứng có tốc độ cao như: DES, IDEA, … để tăng hiệu quả và tốc độ mã hóa. Khi đó, RSA chỉ dùng để mã hóa khóa đối xứng, còn khóa đối xứng dùng để mã hóa toàn bộ dữ liệu.
* Có thể dùng để tạo ra các chữ ký số, có thể chứng minh tính xác thực và toàn vẹn của dữ liệu. Chữ ký số được tạo bằng cách sử dụng khóa bí mật để mã hóa dữ liệu hoặc giá trị băm của dữ liệu, và được xác thực bằng cách sử dụng khóa công khai để giải mã và so sánh.

*Nhược điểm của RSA:*

* Tốc độ chậm hơn rất nhiều so với các hệ mã hóa đối xứng. Thường thì RSA chậm ít nhất 100 lần khi cài đặt bằng phần mềm và có thể chậm hơn từ 1000 đến 10000 lần khi cài đặt bằng phần cứng.
* Kích thước khóa lớn hơn rất nhiều so với các hệ mã hóa đối xứng. Điều này làm tăng dung lượng dữ liệu mã hóa và giảm hiệu suất truyền tin.
* Có thể bị tấn công bằng các phương pháp tiên tiến hơn, như sử dụng các siêu máy tính hay các máy tính lượng tử. Nếu có thể phân tích được các số nguyên tố lớn, hoặc tìm được khóa bí mật từ khóa công khai, thì mã hóa RSA sẽ bị phá vỡ.

## Ngôn ngữ và công cụ sử dụng

### Ngôn ngữ python

Python là một ngôn ngữ lập trình được phát triển bởi Guido van Rossum và được phát hành lần đầu vào năm 1991. Được thiết kế để đơn giản hóa quá trình lập trình và tăng cường tính dễ đọc của mã, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

Python là một ngôn ngữ lập trình đa năng và có thể được sử dụng để phát triển nhiều loại ứng dụng khác nhau, từ các ứng dụng desktop đến các ứng dụng web và trò chơi điện tử. Nó cũng có thể được sử dụng để phân tích dữ liệu, xử lý hình ảnh và thực hiện các tác vụ khác liên quan đến trí tuệ nhân tạo. Một số lợi ích của việc sử dụng Python là:

* Dễ học và sử dụng: Python được thiết kế để đơn giản hóa quá trình lập trình và tăng cường tính dễ đọc của mã, giúp cho các lập trình viên mới có thể nhanh chóng bắt đầu.
* Hỗ trợ đa nền tảng: Python có thể được sử dụng trên nhiều nền tảng khác nhau, bao gồm Windows, Linux và Mac OS.
* Hỗ trợ thư viện đa dạng: có rất nhiều thư viện Python cho phép các lập trình viên phát triển các ứng dụng một cách nhanh chóng và hiệu quả.
* Tính linh hoạt: Python có thể được sử dụng để phát triển nhiều loại ứng dụng khác nhau, từ các ứng dụng desktop đến các ứng dụng web và trò chơi điện tử.
* Tính tương thích và kết nối: Python có thể kết nối với nhiều thư viện khác nhau và các hệ thống quản lý cơ sở dữ liệu Oracle, MySQL và SQLite.

Một số nhược điểm của Python là tốc độ chậm hơn so với một số ngôn ngữ khác và cú pháp của Python không phù hợp với một số lập trình viên. Tuy nhiên, những nhược điểm này ít quan trọng khi so sánh với những ưu điểm của Python.

Để sử dụng ngôn ngữ Python, nhóm sử dụng trên công cụ Visual Studio Code.

### Môi trường Visual Studio Code

Visual Studio là một môi trường phát triển tích hợp (Integrated Development Environment - IDE) được phát triển bởi Microsoft. Visual Studio là một công cụ rất phổ biến, được sử dụ rộng rãi trong ngành công nghiệp phần mềm để phát triển ứng dụng Windows, web di động, trò chơi và ứng dụng đám mây.

Visual Studio cung cấp một loạt các công cụ và tính năng giúp các nhà phát triển phát triển ứng dụng một cách nhanh chóng và dễ dàng. IDE này cho phép bạn viết mã, thực hiện mã hoá, gỡ lỗi, làm việc với các thư viện, kiểm thử và triển khai ứng dụng.

Visual Studio hỗ trợ nhiều ngôn ngữ lập trình, bao gồm Python, C++, C#, Visual Basic, F#, JavaScript và TypeScript. Nó cũng đi kèm với các công cụ để phân tích mã và kiểm tra mã.

Một vài tính năng nổi bật của Visual Studio:

* Tích hợp các công cụ như trình biên dịch, trình gỡ lỗi, quản lý phiên bản và trình nhóm làm việc.
* Hỗ trợ cho nhiều nền tảng và thiết bị.
* Có thể kết hợp với nhiều công cụ bên thứ ba, bao gồm các trình biên dịch thông minh và các công cụ phân tích mã.
* Cung cấp các mẫu lập trình và thư viện để giúp các nhà phát triển tăng tốc quá trình phát triển ứng dụng.

Visual Studio được phát triển cho Windows và có các phiên bản miễn phí và có phí để tải về. Nó cũng được tích hợp trong nhiều sản phẩm Microsoft khác, bao gồm Azure và Office 365.

* Để tải và cài đặt Visual Studio Code, ta truy cập vào đường link: *https://code.visualstudio.com/download*

**A screenshot of a computer program

Description automatically generated with low confidence**

Hình .: Các phiên bản và hệ điều hành phù hợp với máy

* Sau khi cài đặt Visual Studio Code, chúng ta cần phải cài đặt thêm thư viện Pyqt6 để thực hiện lập trình giao diện người dùng :
* Vào Terminal trong Visual Studio Code, gõ câu lệnh: ‘pip install pyqt6’, hệ thống sẽ tiến hành cài đặt Pyqt6 và Qt Designer.
* Ngay tại Terminal, gõ câu lệnh: ‘pip install pyqt 6-tools’, hệ thống tiến hành tải các công cụ hỗ trợ để lập trình giao diện.

### Công cụ Qt Designer

Qt Designer là một công cụ của Qt để thiết kế và xây dựng giao diện người dùng đồ họa (GUI) với các widget của Qt. Bạn có thể sắp xếp và tùy chỉnh các cửa sổ hoặc hộp thoại của bạn theo cách nhìn thấy là gì nhận được (WYSIWYG), và kiểm tra chúng với các kiểu và độ phân giải khác nhau. Các widget và form được tạo bằng Qt Designer tích hợp một cách liền mạch với mã lập trình, sử dụng cơ chế tín hiệu và khe của Qt, để bạn có thể dễ dàng gán hành vi cho các thành phần đồ họa. Tất cả các thuộc tính được thiết lập trong Qt Designer có thể được thay đổi động trong mã. Hơn nữa, các tính năng như thăng tiến widget và plugin tùy chỉnh cho phép bạn sử dụng các thành phần của riêng bạn với Qt Designer.

Có thể sử dụng Qt Designer cùng với Python thông qua PyQt hoặc PySide 6 để tạo các ứng dụng dựa trên Qt và Python. PyQt và PySide6 là hai bộ liên kết Python cho Qt, cho phép bạn truy cập vào các lớp và phương thức của Qt từ mã Python. Bạn có thể sử dụng Qt Designer để tạo các tệp .ui, sau đó sử dụng công cụ pyuic hoặc pyside-uic để chuyển đổi chúng thành mã Python.

Một số ưu điểm của việc sử dụng Qt Designer với Python là:

* Có thể tạo giao diện người dùng một cách nhanh chóng và dễ dàng bằng cách kéo và thả các widget và sắp xếp chúng theo các bố cục khác nhau.
* Có thể xem trước giao diện người dùng của bạn với các kiểu và độ phân giải khác nhau, và kiểm tra tính tương thích của nó với các nền tảng khác nhau.
* Có thể sử dụng các widget tùy chỉnh của riêng bạn với Qt Designer, hoặc sử dụng các plugin để mở rộng chức năng của nó.
* Có thể dễ dàng kết nối các tín hiệu và khe của Qt, để gán hành vi cho các thành phần đồ họa.
* Có thể thay đổi động các thuộc tính được thiết lập trong Qt Designer trong mã Python, hoặc sử dụng các biểu mẫu phong cách để tùy chỉnh giao diện người dùng của bạn.

Một số nhược điểm của việc sử dụng Qt Designer với Python là:

* Phải cài đặt và duy trì các bộ liên kết Python cho Qt, và đảm bảo rằng chúng tương thích với phiên bản Qt của bạn.
* Cần chuyển đổi các tệp .ui thành mã Python mỗi khi bạn thay đổi giao diện người dùng của bạn, hoặc sử dụng các phương thức để tải các tệp .ui trực tiếp trong mã Python.
* Phải tuân thủ các giấy phép của PyQt và PySide6 khi sử dụng chúng. PyQt là phần mềm thương mại, người sử dụng phải mua một giấy phép để sử dụng nó trong các ứng dụng thương mại. PySide 6 là phần mềm mã nguồn mở và tuân thủ các điều khoản của giấy phép LGPL khi sử dụng nó.

# THIẾT KẾ CHI TIẾT

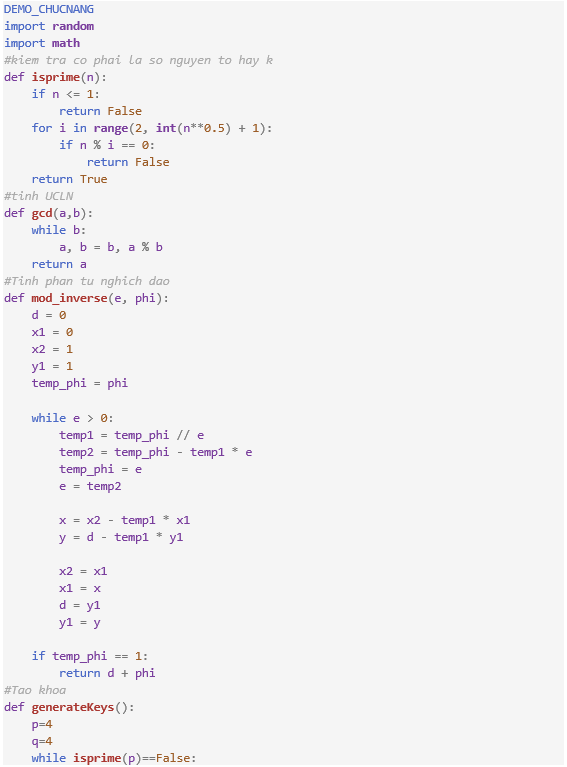
## Mô tả về ứng dụng của RSA

Sản phẩm demo của chúng em là một ứng dụng truyền tin, sử dụng thuật toán RSA để mã hóa và giải mã các tin nhắn giữa hai người dùng. Ứng dụng này có thể đảm bảo rằng chỉ có người gửi và người nhận mới có thể đọc được nội dung của tin nhắn, và không ai khác có thể can thiệp hoặc gián điệp được. Cách hoạt động của ứng dụng truyền tin của chúng em như sau:

* Khi người dùng sử ứng dụng, ứng dụng sẽ tạo ra một cặp khóa ngẫu nhiên gồm khóa công khai (public key) và khóa bí mật (private key), chúng sẽ tạo ra mỗi cặp khóa ngẫu nhiên mới với mỗi lần gửi tin.
* Khi người dùng muốn gửi một tin nhắn, ứng dụng sẽ sử dụng khóa phiên để mã hóa đoạn tin nhắn đó. Sau đó, ứng dụng sẽ gửi tin nhắn đã mã hóa đến bên người nhận.
* Khi người nhận nhận được tin nhắn, ứng dụng sẽ sử dụng khóa phiên để giải mã tin nhắn. Sau đó, ứng dụng sẽ hiển thị tin nhắn gốc cho người nhận. Chỉ có người gửi và người nhận mới có thể xem được nội dung tin nhắn.
* Ứng dụng này cũng giúp chúng ra mã hóa cả file doc hc txt. Ngoài ra, với mỗi lần gửi tin nhắn, thì ứng dụng cũng sẽ in ra thông tin khóa công khai và khóa bí mật của lần gửi tin đó.

## Cài đặt thuật toán RSA

### Cài đặt thuật toán hỗ trợ



Hình .: Thuật toán hỗ trợ

* Hàm isprime(n) kiểm tra xem một số nguyên n có phải là số nguyên tố hay không. Nếu n <= 1, hàm trả về False. Nếu n > 1, hàm sử dụng một vòng lặp for để kiểm tra xem n có chia hết cho bất kỳ số nguyên nào từ 2 đến căn bậc hai của n hay không. Nếu có, hàm trả về False. Nếu không, hàm trả về True. Hàm này có thể được sử dụng để chọn các số nguyên tố p và q cho RSA.
* Hàm gcd(a,b) tính ước chung lớn nhất (UCLN) của hai số nguyên a và b. Hàm sử dụng thuật toán Euclid, một thuật toán lặp để tìm UCLN của hai số nguyên bằng cách sử dụng phép chia lấy dư. Hàm này có thể được sử dụng để kiểm tra xem hai số nguyên có nguyên tố cùng nhau hay không, tức là UCLN của chúng bằng 1.
* Hàm mod\_inverse(e, phi) tính phần tử nghịch đảo của e theo modulo phi, tức là số nguyên d sao cho d \* e = 1 (mod phi). Hàm sử dụng thuật toán mở rộng Euclid, một thuật toán lặp để tìm phần tử nghịch đảo của e bằng cách sử dụng các phương trình đồng dư. Hàm này có thể được sử dụng để tìm số d cho RSA.

### Cài đặt thuật toán tạo khóa, mã hóa và giải mã

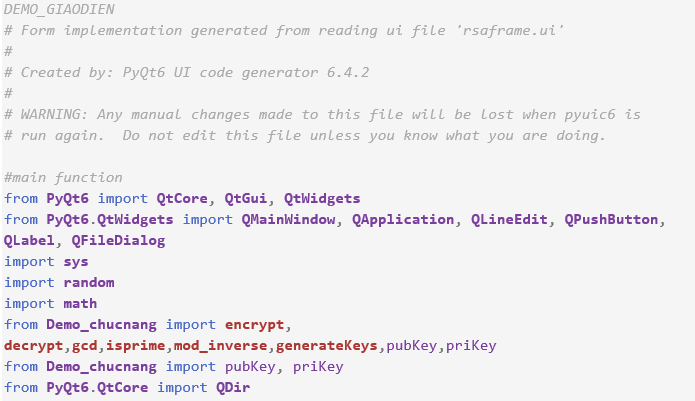


Hình .: Cài đặt thuật toán tạo khóa, mã hóa và giải mã

* Hàm generateKeys() tạo ra khóa công khai và khóa bí mật cho RSA. Hàm này thực hiện các bước sau:
  + Khởi tạo hai số nguyên p và q bằng 4.
  + Sử dụng hàm isprime() để kiểm tra xem p và q có phải là số nguyên tố hay không. Nếu không, sử dụng hàm random.randint() để tạo ra hai số nguyên tố ngẫu nhiên trong khoảng từ 1000000000000 đến 9000000000000.
  + Nếu p và q bằng nhau, sử dụng hàm random.randint() để tạo ra hai số nguyên tố ngẫu nhiên khác nhau trong cùng khoảng.
  + Tính n bằng tích của p và q.
  + Tính phi\_n bằng tích của p-1 và q-1.
  + Sử dụng hàm random.randrange() để tạo ra số nguyên e ngẫu nhiên trong khoảng từ 1 đến phi\_n.
  + Sử dụng hàm gcd() để kiểm tra xem e và phi\_n có phải là nguyên tố cùng nhau hay không. Nếu không, sử dụng hàm random.randrange() để tạo ra số nguyên e ngẫu nhiên khác trong cùng khoảng.
  + Sử dụng hàm mod\_inverse() để tìm số nguyên d là phần tử nghịch đảo của e theo modulo phi\_n.
  + Khai báo hai biến toàn cục pubKey và priKey để lưu trữ khóa công khai và khóa bí mật. Khóa công khai là một cặp số (e,n) và khóa bí mật là một cặp số (d,n).
* Hàm encrypt(msg,pubKey) mã hóa một chuỗi msg bằng khóa công khai pubKey. Hàm này thực hiện các bước sau:
  + Giải nén khóa công khai thành hai số e và n.
  + Khởi tạo một chuỗi rỗng banMa để lưu trữ bản mã.
  + Duyệt qua từng ký tự i trong chuỗi msg.
  + Sử dụng hàm ord() để chuyển ký tự i thành một số nguyên temp.
  + Sử dụng hàm pow() để tính lũy thừa temp mũ e theo modulo n. Kết quả là một số nguyên là bản mã của ký tự i.
  + Nối kết quả vào chuỗi banMa với một khoảng trắng phân cách.
  + Trả về chuỗi banMa là bản mã của chuỗi msg.
* Hàm decrypt(cipher,priKey) giải mã một chuỗi cipher bằng khóa bí mật priKey. Hàm này thực hiện các bước sau:
  + Giải nén khóa bí mật thành hai số d và n.
  + Khởi tạo một chuỗi rỗng msg để lưu trữ bản rõ.
  + Sử dụng hàm split() để tách chuỗi cipher thành một danh sách các phần tử. Mỗi phần tử là một số nguyên là bản mã của một ký tự.
  + Duyệt qua từng phần tử part trong danh sách.
  + Nếu part không rỗng, sử dụng hàm int() để chuyển part thành một số nguyên c.
  + Sử dụng hàm pow() để tính lũy thừa c mũ d theo modulo n. Kết quả là một số nguyên là bản rõ của ký tự c.
  + Sử dụng hàm chr() để chuyển kết quả thành một ký tự và nối vào chuỗi msg.
  + Trả về chuỗi msg là bản rõ của chuỗi cipher.

## Thiết kế giao diện

* Import các thư viện cần thiết:



Hình .: Cài đặt các thư viện cần thiết

* Định nghĩa một lớp Ui\_MainWindow có phương thức setupUi để thiết lập các thành phần giao diện:



Hình .: Thiết lập giao diện bằng Python

* MainWindow: là cửa sổ chính của ứng dụng, có kích thước 842x600 pixel và font chữ 12.
* centralwidget: là widget trung tâm của cửa sổ, chứa các widget khác.
* label: là một nhãn hiển thị tiêu đề “Ứng dụng mã hóa RSA” ở vị trí (310, 10) và kích thước (300, 31).
* line: là một đường thẳng dọc phân chia giao diện thành hai phần, ở vị trí (380, 60) và chiều cao 321 pixel.
* label\_2: là một nhãn hiển thị chữ “Nhập bản rõ” ở vị trí (10, 50) với font chữ 10, in đậm và gạch chân.
* label\_public\_key: là một nhãn hiển thị khóa công khai ở vị trí (60, 330) và kích thước (321, 31).
* label\_3: là một nhãn hiển thị chữ “P =” ở vị trí (10, 80) với font chữ 10.
* label\_4: là một nhãn hiển thị chữ “Q =” ở vị trí (10, 280) với font chữ 10.
* nhap\_BR: là một trường nhập liệu cho bản rõ, ở vị trí (60, 80) và kích thước (321, 121).

Đoạn code này cũng nhập các thư viện và hàm cần thiết cho mã hóa RSA từ file Demo\_chucnang.py, và khai báo các biến toàn cục pubKey và priKey cho khóa công khai và khóa bí mật. Và tương tự, chúng ta cũng thiết lập dược label\_5, label\_6, label\_7.

* Dịch các chuỗi văn bản của các widget giao diện:



Hình .: Dịch các chuỗi văn bản bằng Python

* retranslateUi: là một phương thức để dịch các chuỗi văn bản của các widget giao diện theo ngôn ngữ được chọn, sử dụng hàm \_translate của lớp QCoreApplication. Phương thức này cũng đặt tiêu đề cho cửa sổ chính là “MainWindow” và các nhãn và nút bấm theo nội dung tương ứng.
* connectSlotsByName: là một phương thức để kết nối các sự kiện của các widget giao diện với các hàm xử lý tương ứng, dựa trên tên của các widget. Phương thức này được gọi sau khi thiết lập các widget giao diện.
* btn\_maHoa.clicked.connect(self.runEn): là một dòng lệnh để kết nối sự kiện nhấn nút bấm “Mã Hoá” với hàm runEn của lớp Ui\_MainWindow, hàm này sẽ thực hiện mã hóa bản rõ bằng thuật toán RSA.
* btn\_Chuyen.clicked.connect(self.chuyen): là một dòng lệnh để kết nối sự kiện nhấn nút bấm “Gửi” với hàm chuyen của lớp Ui\_MainWindow, hàm này sẽ chuyển bản mã từ phần người gửi sang phần người nhận.
* btn\_giaiMa.clicked.connect(self.runDe): là một dòng lệnh để kết nối sự kiện nhấn nút bấm “Giải Mã” với hàm runDe của lớp Ui\_MainWindow, hàm này sẽ thực hiện giải mã bản mã bằng thuật toán RSA.
* file\_MH.clicked.connect(self.filemh): là một dòng lệnh để kết nối sự kiện nhấn nút bấm “File” ở phần người gửi với hàm filemh của lớp Ui\_MainWindow, hàm này sẽ mở một hộp thoại để chọn file chứa bản rõ.
* file\_GM.clicked.connect(self.filegm): là một dòng lệnh để kết nối sự kiện nhấn nút bấm “File” ở phần người nhận với hàm filegm của lớp Ui\_MainWindow, hàm này sẽ mở một hộp thoại để chọn file chứa bản mã.
* luu\_MH.clicked.connect(self.luumh): là một dòng lệnh để kết nối sự kiện nhấn nút bấm “Lưu” ở phần người gửi với hàm luumh của lớp Ui\_MainWindow, hàm này sẽ lưu bản mã vào file.
* luu\_GM.clicked.connect(self.luugm): là một dòng lệnh để kết nối sự kiện nhấn nút bấm “Lưu” ở phần người nhận với hàm luugm của lớp Ui\_MainWindow, hàm này sẽ lưu bản rõ vào file.

## Xử lý các chức năng cho widget giao diện:



Hình .: Xử lý các chức năng cho widget (a)



Hình .: Xử lý các chức năng cho widget (b)

Đoạn code này tiếp tục định nghĩa các hàm xử lý cho các sự kiện của các widget giao diện cho ứng dụng mã hóa RSA, bao gồm:

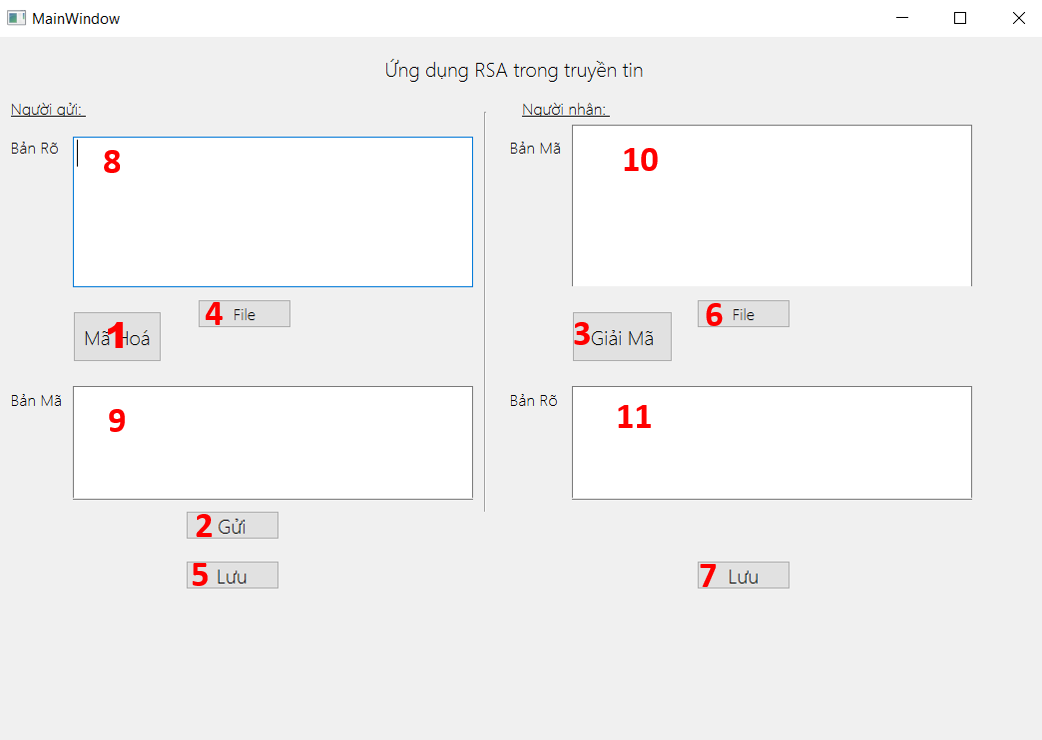
* runEn: là hàm để mã hóa bản rõ bằng thuật toán RSA, bao gồm các bước: tạo khóa công khai và khóa bí mật, lưu khóa vào file, mã hóa bản rõ và hiển thị bản mã.
* savePublicKeyToFile: là hàm để lưu khóa công khai vào file có tên “public\_key.txt” ở thư mục hiện tại, và hiển thị khóa công khai trên nhãn label\_public\_key.
* savePrivateKeyToFile: là hàm để lưu khóa bí mật vào file có tên “private\_key.txt” ở thư mục hiện tại, và hiển thị khóa bí mật trên nhãn label\_public\_key.
* chuyen: là hàm để chuyển bản mã từ phần người gửi sang phần người nhận, bằng cách lấy nội dung của trường show\_BM và đặt vào trường nhap\_BM.
* runDe: là hàm để giải mã bản mã bằng thuật toán RSA, bằng cách lấy nội dung của trường nhap\_BM và giải mã bằng khóa bí mật, sau đó hiển thị bản rõ trên trường show\_BR.
* filemh: là hàm để mở một hộp thoại để chọn file chứa bản rõ, và đọc nội dung của file và đặt vào trường nhap\_BR.
* filegm: là hàm để mở một hộp thoại để chọn file chứa bản mã, và đọc nội dung của file và đặt vào trường nhap\_BM.
* luumh: là hàm để lưu bản mã vào file, bằng cách mở một hộp thoại để chọn tên file và ghi nội dung của trường show\_BM vào file.
* luugm: là hàm để lưu bản rõ vào file, bằng cách mở một hộp thoại để chọn tên file và ghi nội dung của trường show\_BR vào file.

Đoạn code này cũng định nghĩa một đoạn code chạy chính để khởi tạo ứng dụng, tạo một cửa sổ chính, tạo một đối tượng của lớp Ui\_MainWindow, thiết lập giao diện cho cửa sổ chính, hiển thị cửa sổ chính và chờ sự kiện. Đoạn code này cũng xoá các biến toàn cục pubKey và priKey sau khi chạy xong.

# CHƯƠNG TRÌNH VÀ DEMO

## Giao diện chương trình

Sau khi thực hiện chạy đoạn code, giao diện chương trình sẽ hiện ra như sau:



Hình .: Chương trình demo

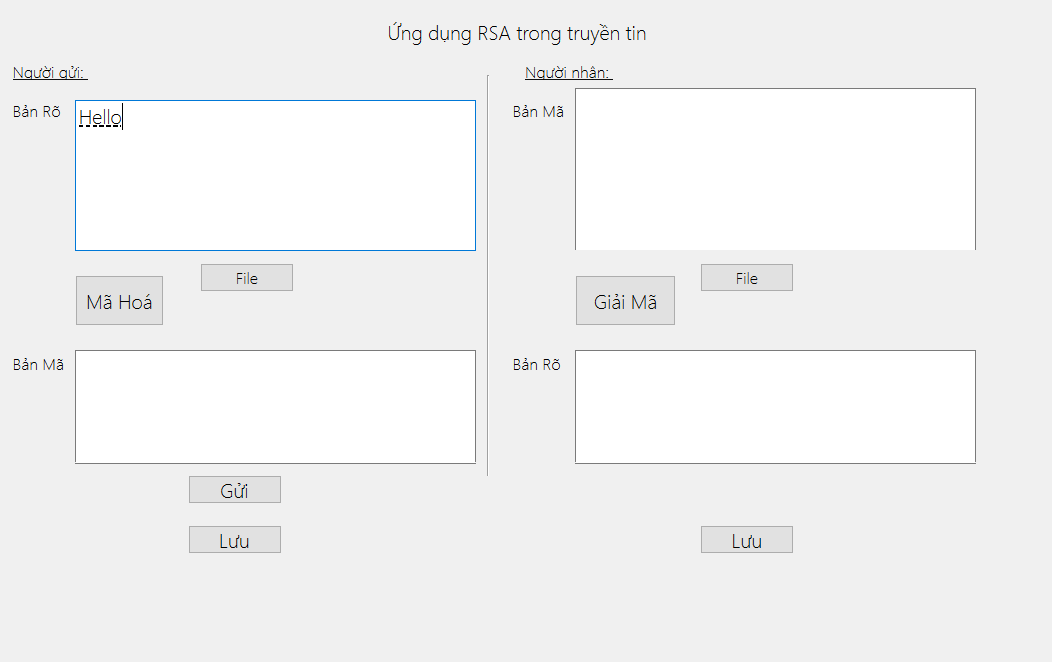
* Chú thích:

1. Mã hóa: Người dùng ấn nút này để thực hiện mã hóa bản rõ đầu vào(8) và đưa thông tin bản mã vừa mã hóa sang bản mã đầu ra(9)
2. Gửi: Thực hiện chuyển toàn bộ bản mã đầu ra(9) sang bản mã đầu vào(10).
3. Giải mã: Thực hiện giải mã bản mã đầu vào(10) và đưa bản rõ được giải mã vào bản rõ đầu ra(11).
4. Chọn file bản rõ đầu vào: Người dùng ấn nút này để thực hiện chọn 1 file(txt hoặc docx) bản rõ có sẵn trên máy để mã hóa .
5. Lưu file bản mã: Thực hiện lưu bản mã vừa được mã hóa thành 1 file(txt hoặc docx)
6. Chọn file bản mã đầu vào: thực hiện chọn file bản mã có sẵn trên máy để giải mã
7. Lưu file bản rõ: Người nhận có thể nhấn nút 7 để thực hiện lưu bản rõ vừa giải mã thành 1 file(txt hoặc docx)
8. 8-9-10-11: Ô văn bản để thực hiện nhập và hiển thị dữ liệu đầu vào/ra.

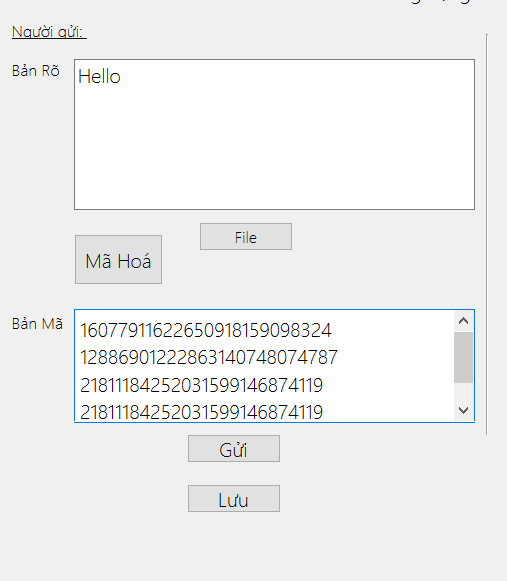
## Hướng dẫn và chạy chương trình

* Bước 1: Nhập vào dữ liệu đầu vào ở ô Bản rõ để nhập vào bản rõ cần mã hóa hoặc ấn File để chọn file văn bản chứa bản rõ. Ví dụ: “ Hello”

Hình 3.2: Minh họa bước 1

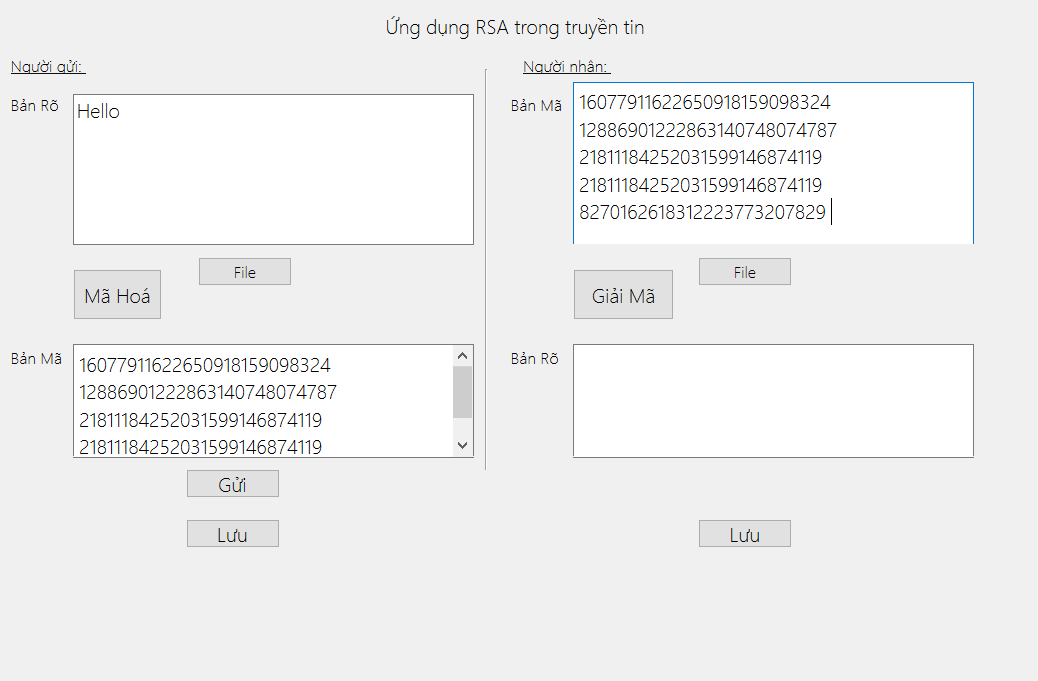


* Bước 2: Ấn nút Mã hóa để thực hiện mã hóa văn bản trong ô Bản rõ và hiện bản mã hiện vào ô Bản mã.

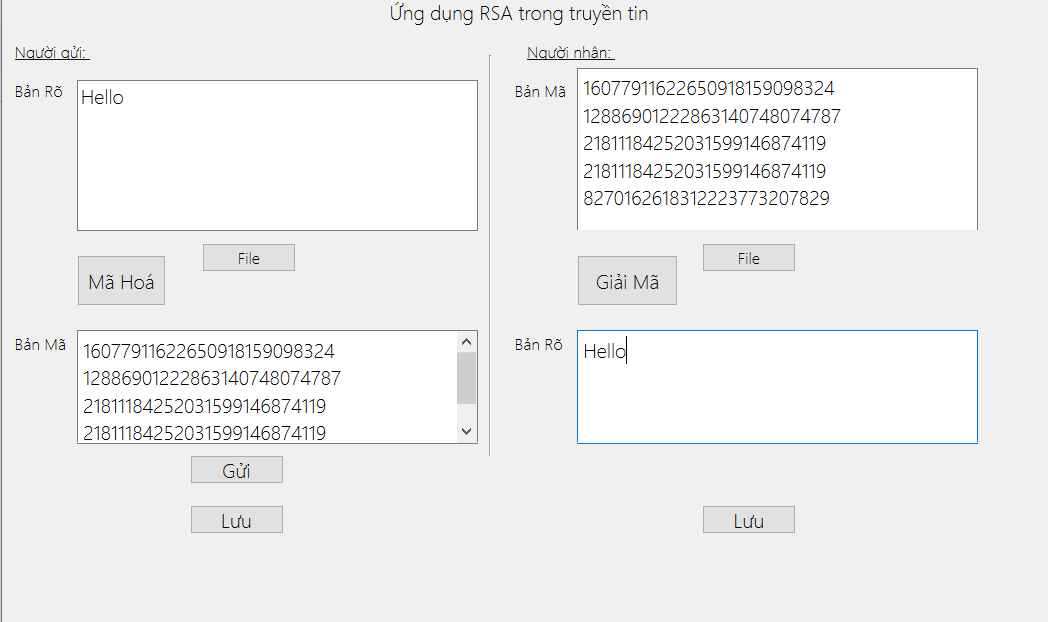


Hình .: Minh họa bước 2

* Bước 3: Ấn nút Gửi để thực hiện đưa văn bản ô Bản mã phần Mã hóa sang ô Bản mã phần Giải mã. Hoặc ấn nút Lưu để lưu bản mã vào 1 file văn bản.



Hình .: Minh họa bước 3

* Bước 4: Ấn nút Giải mã để thực hiện mã hóa văn bản trong ô Bản mã và hiện bản mã hiện vào ô Bản rõ, đây chính là thông tin mà người gửi đã gửi.

Hình 3.5: Minh họa bước 4

* Bước 5: Ấn lưu để lưu bản mã vào 1 file văn bản.

# KẾT LUẬN VÀ KIẾN NGHỊ

Trên đây nhóm 16 đã trình bày toàn bộ nội dung về RSA và ứng dụng của mã hóa RSA trong truyền tin. Để tìm hiểu về cách hoạt động của mật mã khóa công khai, nhóm 16 đã tìm hiểu về cơ sở toán học, thuật toán Ơ-clid mở rộng, thuật toán bình phương và nhân, thuật toán tạo khóa và các thuật toán để mã hóa và giải mã RSA. Chương trình mã hóa và giải mã được xây dựng trên ngôn ngữ Python.

Do điều kiện thời gian có hạn cùng với khả năng còn hạn chế nên không tránh khỏi thiếu sót. Nhóm chúng em rất mong nhận được những nhận xét và góp ý từ cô để có thể hoàn thiện hơn bài báo cáo và học hỏi thêm nhiều kiến thức bổ ích. Một lần nữa chúng em xin chân thành cảm ơn cô Lê Thị Anh, cô giáo hướng dẫn trực tiếp, tận tình giúp đỡ và tạo điều kiện để nhóm em hoàn thành bài đồ án này.

## Những kiến thức và kỹ năng đã học được

### Kiến thức

RSA là một hệ mã hóa bất đối xứng được phát triển bởi Ron Rivest, Adi Shamir và Leonard Adleman (tên của nó cũng chính là tên viết tắt của 3 tác giả này) và được sử dụng rộng rãi trong công tác mã hoá và công nghệ chữ ký điện tử. Trong hệ mã hóa này, public key có thể chia sẻ công khai cho tất cả mọi người. Hoạt động của RSA dựa trên 4 bước chính: sinh khóa, chia sẻ key, mã hóa và giải mã.

* Sinh khóa
* Chọn ngẫu nhiên 2 số nguyên tố p và q phân biệt
* Tính n = p.q. Sau này, n sẽ được dùng làm modulus trong cả public key và private key.
* Tính giá trị số giả nguyên tố bằng phi hàm Euler(Φ(n))
* Φ(n) = (p – 1) (q – 1)
* Chọn một số tự nhiên b trong khoảng (2, Φ(n)) sao cho gcd(b, Φ(n))= 1, tức là b và Φ(n) là số nguyên tố cùng nhau
* Tính toán số a sao cho a ≡ b-1 (mod Φ(n)) hay viết dễ hiểu hơn thì ab ≡ 1 (mod Φ(n)). Số a được gọi là số nghịch đảo modulo của b (theo modulo mod Φ(n))
* Mã hóa và giải mã

Nếu chúng ta có bản rõ M, chúng ta cần chuyển nó thành một số tự nhiên m trong khoảng (0, n) sao cho m, n nguyên tố cùng nhau. Việc này rất dễ dàng thực hiện bằng cách thêm một các kỹ thuật padding. Tiếp theo, chúng ta sẽ mã hóa m, thành c như sau:

**c= mb mod n**

Sau đó giá trị c sẽ được chuyển cho người nhận, ở phía người nhận, họ sẽ giải mã từ c để lấy được m như sau:

**m= ca mod n**

Từ m có thể lấy lại được bản rõ bằng cách đảo ngược padding

### Kỹ năng

* Kỹ năng giao tiếp: Ngoài những kiến thức, kinh nghiệm mà bạn đã học thì kỹ năng giao tiếp là một trong những kỹ năng quan trọng nhất để bạn dẫn đến thành công. Giao tiếp là một phương tiện cầu nối để bạn có thể giao lưu với mọi người, bạn có thể thuyết phục mọi người chấp nhận ý kiến của bạn và bạn có thể bày tỏ được nhu cầu của bạn.
* Kỹ năng thuyết trình đám đông: Muốn thành công trong sự nghiệp thì bạn cần phải có kỹ năng thuyết trình để mọi người có thể chấp nhận ý kiến của bạn, chấp nhận hướng giải quyết vấn đề đúng đắn của bạn. Bạn có thể tạo dựng cho mình một thương hiệu cá nhân trong mắt người khác thông qua việc thuyết trình.
* Kỹ năng làm việc nhóm: có thể hiểu 1 cách đơn giản là nhiều người cùng nhau kết hợp các ưu điểm của mình để thực tốt một nhiệm vụ hướng tới một mục tiêu chung. Cách làm việc này sẽ giúp các cá nhân bổ sung những thiếu sót cho nhau và hoàn thiện bản thân mình. Để công việc của nhóm đạt kết quả cao nhất, các thành viên phải có kỹ năng làm việc nhóm thuần thục. Ngoài ra, làm việc nhóm (sức mạnh của teamwork) giúp cho mỗi cá nhân đề cao tinh thần tập thể, nâng cao hiệu quả công việc và sự gắn bó.
* Kỹ năng lãnh đạo: là một kỹ năng vô cùng quan trọng. Bạn phải lãnh đạo làm sao mà nhân viên phải kính trọng và nghe theo lời bạn. Muốn trở thành một nhà lãnh đạo tài ba thì bạn phải là một người tài giỏi, có tầm nhìn xa trông rộng, dự đoán và đưa ra quyết định đúng đắn. Bạn phải phát hiện ra khả năng của thành viên nhóm mình và phát huy hết tài năng của nhân viên đó. Để lãnh đạo giỏi không còn cách nào khác là bạn phải kiên trì học tập và rèn luyện.
* Kỹ năng giải quyết vấn đề: Trong công việc, có rất nhiều vấn đề xảy ra một cách đột ngột mà bạn không thể biết trước được. Để giải quyết vấn đề một cách ổn thỏa bạn phải phân tích, xem xét thật kĩ. Bạn phải tìm ra nguyên nhân của vấn đề và từng bước một giải quyết chúng. Nếu thành thạo kỹ năng này thì bạn không còn phải lo lắng mỗi khi có vấn đề gì xảy ra bất ngờ nữa. Kỹ năng này giúp cho công việc của bạn tiến triển nhanh hơn
* Kỹ năng lập trình : Nhờ vào bài tập lớn, các thành viên trong nhóm thành thạo ngôn ngữ lập trình Python, cách sử dụng các thư viện hữu ích và giao diện của ngôn ngữ.

## Hướng phát triển

* RSA và giải pháp bảo mật hai yếu tố

Trên cơ sở mức độ phức tạp của thuật toán phân tích số nguyên thành thừa số, có thể đảm bảo tính an toàn cao của giải pháp bảo mật sử sụng thuật toán RSA. Giải pháp bảo mật sử dụng thuật toán RSA thỏa mãn 4 yêu cầu:

* Độ bảo mật cao (để giải mã được mà không biết khóa sẽ mất nhiều thời gian)
* Có thể dùng chung
* Thực hiện nhanh (mã hóa và giải mã tốn ít thời gian)
* Chống từ chối (xác định dõ người thực hiện)

Hiện nay, trên thị trường có nhiều giải pháp bảo mật để tăng cường tính an toàn cho hệ thống công nghệ thông tin đặc biệt cho ngành Tài chính – Ngân hàng. Tuy nhiên, xét về mức độ an toàn dựa trên thuật toán, thì giải pháp sử dụng thuật toán RSA luôn cho độ tin cậy cao. Đây là điểm đáng lưu tâm trong bối cảnh nguy cơ rủi ro an toàn thông tin như hiện nay.

Nhận thức được tầm quan trọng và bảo mật thông tin, Ngân hàng Đầu tư và Phát triển Việt Nam (BIDV) đã nghiên cứu và đầu tư bảo mật hệ thống công nghệ thông tin, đặc biệt, là giải pháp xác thực hai yếu tố RSA cho hệ thống cốt lõi ngân hàng và một số bảo mật khác trong các giao dịch của ngân hàng…

* Tạo chữ ký số cho văn bản

Thuật toán RSA được dùng để tạo “chữ ký số” cho văn bản, giả sử người A muốn gửi cho người B một văn bản chữ ký của mình. Để làm việc này, A tạo ra môt giá trị băm (hash Value) của văn bản cần ký và tính giá trị d mod n của nó (giống như việc A thực hiện giải mã). Gía trị tìm được cuối cùng chính là chữ ký điện tử của văn bản đang xét. Khi B nhận được văn bản cùng chứ ký điện tử, anh ta tính giá trị e mod n sửa chữ ký đồng thời với việc tính giá trị băm của văn bản. Nếu 2 giá trị như nhau thì chữ ký của A không bị thay đổi.

* Ứng dụng của RSA trong công nghệ thông tin

Trong ngôn ngữ lập trình Java, các nhà lập trình viên thường sử dụng những đoạn code chứa RSA để tăng tính bảo mật cho trang web và ứng dụng cũng như đảm bảo an toàn cho người sử dụng. Các đoạn code RSA này có thể hoạt động dưới bất kỳ sự thay đổi nào của môi trường. Ngoài ra, các lập trình viên cũng sử dụng các ngôn ngữ lập trình khác bên cạnh Java có thể tìm hiểu và ứng dụng những tính năng của RSA trong hoạt động làm việc và bảo mật thông tin. Ngày nay việc sử dụng các ứng dụng, trang web trên internet ngày càng gia tăng khiến cho vấn đề bảo mật dữ liệu càng được chú trọng. Những dữ liệu này có thể là những thông tin bí mật cá nhân, thông tin về tài chính,... gây không ít nguy hại cho người sử dụng. Cũng chính vì lý do này mà thuật toán RSA được biết đến và sử dụng nhiều hơn trong tất cả các lĩnh vực đặc biệt là trong ngành ngân hàng.

* Ứng dụng RSA trong sử dụng cửa cuốn

Trong tay điều khiển cửa cuốn là sử dụng các mã số ngẫu nhiên và mã hóa đối xứng để đảm bảo tính bảo mật cửa dữ liệu được truyền qua sóng radio giữa tay điều khiển và motor cửa cuốn. Khi tay điều khiển nhấn một nút trên nó, nó sẽ tạo ra một số ngẫu nhiên mới để mã hóa dữ liệu. Sau đó, nó sẽ sử dụng khóa công khai của motor của cuốn để mã hóa mã số ngẫu nhiên này và gửi qua sóng radio.

Khi motor cửa cuốn nhận được dữ liệu, nó sẽ sử dụng khóa bí mật của mình để giải mã mã số ngẫu nhiên này để giải mã các lệnh từ tay điều khiển. Các mã số ngẫu nhiên đảm bảo được tạo ra và sử dụng một cách độc lập cho mỗi lần gửi , đảm bảo là duy nhất và khó bị đoán trước.