

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI



BÀI TẬP LỚN

Môn: An toàn và bảo mật thông tin

XÂY DỰNG CHƯƠNG TRÌNH

MÃ HÓA VÀ GIẢI MÃ RSA

GVHD : *TS. Nghiêm Bá Nghiễn*

Nhóm : **04**

Lớp : **20232IT6001006**

Thành viên nhóm :

- | | |
|------------------------------|---------------------|
| 1. Đinh Đăng Duy Anh | - 2021602842 |
| 2. Nguyễn Phương Long | - 2021601030 |
| 3. Vũ Thị Thanh Lan | - 2021601616 |
| 4. Đinh Thanh Sáng | - 2021602908 |
| 5. Lâm Hoàng An | - 2021601655 |

Hà Nội, Năm 2024

MỤC LỤC

MỤC LỤC	1
DANH MỤC HÌNH ẢNH.....	3
LỜI CẢM ƠN.....	4
LỜI NÓI ĐẦU	5
CHƯƠNG 1: TỔNG QUAN	6
1.1. Giới thiệu về hệ mật mã.....	6
1.2. Các hệ mật mã	7
1.3. Hệ mật mã công khai	7
1.4. Giới thiệu chung về hệ mật mã RSA	8
1.5. Cơ sở lý thuyết.....	9
1.5.1. Số học đồng dư (modulo)	10
1.5.2. Thuật toán Euclid.....	10
1.5.3. Định lý Fermat.....	15
1.5.4. Hàm số Euler	15
1.5.5. Thuật toán Miller-Rabin	16
1.6. Thuật toán tạo khóa, mã hóa và giải mã hệ mật RSA	18
1.6.1. Thuật toán tạo khóa	19
1.6.2. Thuật toán mã hóa	20
1.6.3. Thuật toán giải mã	20
1.7. Ưu và nhược điểm của hệ mật mã RSA	20
1.7.1. Ưu điểm	20
1.7.2. Nhược điểm	21
1.8. Độ an toàn của hệ mật mã RSA.....	21
CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU	22
2.1. Giới thiệu	22
2.2. Nội dung thuật toán	22
2.3. Thiết kế, cài đặt chương trình demo thuật toán.....	23
2.3.1. Giao diện chương trình demo theo ngôn ngữ Java.....	23
2.3.2. Giao diện chương trình demo theo ngôn ngữ JavaScript	24
2.3.3. Giao diện chương trình demo theo ngôn ngữ PHP	25
2.3.4. Giao diện chương trình demo theo ngôn ngữ C#	26
2.3.5. Giao diện chương trình demo theo ngôn ngữ Python	27

2.4. Cài đặt và triển khai.....	27
2.4.1. Giới thiệu công cụ	27
2.4.1.1. Giới thiệu công cụ Visual Studio Code.....	27
2.4.1.2. Giới thiệu công cụ Eclipse.....	28
2.4.1.3. Giới thiệu công cụ Visual Studio 2022	29
2.4.2. Hướng dẫn cài đặt và chạy chương trình.....	30
2.4.2.1. Cài đặt Eclipse và chạy chương trình demo	30
2.4.2.2. Cài đặt Visual Studio 2022 và chạy chương trình demo.....	33
2.4.2.3. Cài đặt Visual Studio Code và chạy chương trình demo	33
2.4.2.4. Cài đặt Visual Studio Code và chạy chương trình demo	34
2.5. Thực hiện bài toán	38
2.5.1. Phân công công việc	38
2.5.2. Cài đặt chương trình	40
CHƯƠNG 3: KIẾN THỨC LĨNH HỘI VÀ BÀI HỌC KINH NGHIỆM.....	43
3.1. Nội dung đã thực hiện	43
3.1.1. Các kiến thức đã lĩnh hội.....	43
3.1.2. Các kỹ năng đã tiếp thu	43
3.1.3. Bài học kinh nghiệm.....	44
3.2. Hướng phát triển.....	44
3.2.1. Tính khả thi của chủ đề nghiên cứu.....	44
3.2.2. Những thuận lợi, khó khăn	45
3.2.3. Hướng phát triển và mở rộng của đề tài	45
KẾT LUẬN	47
TÀI LIỆU THAM KHẢO	48

DANH MỤC HÌNH ẢNH

<i>Hình 1. Quá trình mã hóa và giải mã.....</i>	<i>6</i>
<i>Hình 2. Sơ đồ mã hóa và giải mã hệ mật mã RSA.....</i>	<i>9</i>
<i>Hình 3. Mã hóa và giải mã RSA ứng dụng cho bảo mật.....</i>	<i>20</i>
<i>Hình 4: Giao diện tạo khóa bằng ngôn ngữ Java.....</i>	<i>23</i>
<i>Hình 5: Giao diện mã hóa bằng ngôn ngữ Java.....</i>	<i>23</i>
<i>Hình 6: Giao diện giải mã bằng ngôn ngữ Java.....</i>	<i>24</i>
<i>Hình 7: Giao diện tạo khóa bằng ngôn ngữ JavaScript.....</i>	<i>24</i>
<i>Hình 8: Giao diện mã hóa bằng ngôn ngữ JavaScript.....</i>	<i>25</i>
<i>Hình 9: Giao diện giải mã bằng ngôn ngữ JavaScript.....</i>	<i>25</i>
<i>Hình 10: Giao diện chương trình demo bằng ngôn ngữ PHP.....</i>	<i>25</i>
<i>Hình 11: Giao diện chương trình demo bằng ngôn ngữ C#.....</i>	<i>26</i>
<i>Hình 12: Giao diện chương trình demo bằng ngôn ngữ Python</i>	<i>27</i>

LỜI CẢM ƠN

Bài báo cáo học phần An Toàn Và Bảo Mật Thông Tin với đề tài “*Xây dựng chương trình mã hóa và giải mã RSA*” là kết quả của quá trình học tập không ngừng nghỉ của các thành viên trong nhóm và nhận được sự giúp đỡ tận tình, động viên, khích lệ của thầy cô, bạn bè. Qua đây, nhóm chúng em xin gửi lời cảm ơn chân thành đến những người đã giúp đỡ chúng em trong thời gian học tập, nghiên cứu vừa qua.

Trước hết, chúng em xin chân thành cảm ơn ban giám hiệu cùng toàn thể quý thầy, cô khoa Công nghệ thông tin trường Đại học Công nghiệp Hà Nội đã tạo điều kiện cho nhóm chúng em hoàn thành tốt bài tiểu luận này.

Em xin trân trọng gửi đến thầy Nghiêm Bá Nghiễm - người đã trực tiếp hướng dẫn cũng như cung cấp tài liệu, thông tin khoa học cần thiết để hoàn thành bài báo cáo này lời cảm ơn chân thành và sâu sắc nhất. Đó là những góp ý hết sức quý báu không chỉ trong quá trình thực hiện bài tiểu luận này mà còn là hành trang tiếp bước trong quá trình học tập và lập nghiệp sau này.

Và cuối cùng, xin gửi lời cảm ơn đến gia đình, bạn bè, tập thể lớp 20232IT6001006, những người luôn sẵn sàng sẻ chia và giúp đỡ chúng em trong học tập và cuộc sống. Mong rằng, chúng ta sẽ mãi mãi gắn bó với nhau.

Xin chúc những điều tốt đẹp nhất sẽ luôn đồng hành cùng mọi người.

Chúng em xin chân thành cảm ơn!

Nhóm sinh viên thực hiện

LỜI NÓI ĐẦU

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

Ngày nay với sự phát triển của khoa học công nghệ, 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. Do đó nhu cầu về an toàn và bảo mật thông tin trên máy tính càng nhiều và việc sử dụng mật mã mã hoá càng phổ biến hơn.

Từ nhu cầu thực tế đó, nhằm có cái nhìn bao quát về các hệ mã hóa, cách mã hóa, giải mã dữ liệu và ứng dụng thực tế của các hệ mã hóa, nhóm chúng em lựa chọn thực hiện đề tài “*Xây dựng chương trình mã hóa và giải mã RSA*”.

Bài báo cáo gồm 3 chương:

Chương 1. Tổng quan

Chương 2. Kết quả nghiên cứu

Chương 3. Kiến thức lĩnh hội và bài học kinh nghiệm

Đề tài được hoàn thành bằng sự cộng tác của các thành viên nhóm cùng sự hướng dẫn của thầy Nghiêm Bá Nghiễn . Nội dung đề tài được hoàn thiện dựa trên lý thuyết đã học về chuẩn dữ liệu RSA cùng nhiều tài liệu tham khảo khác tuy nhiên không tránh khỏi thiếu sót mong nhận thêm phản ánh và góp ý từ phía giảng viên và quý bạn đọc .

CHƯƠNG 1: TỔNG QUAN

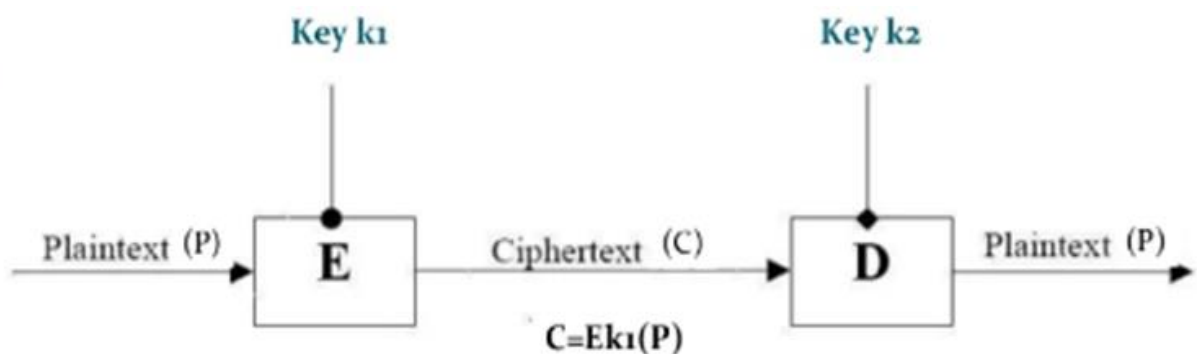
1.1. Giới thiệu về hệ mật mã

Để đảm bảo việc truyền tin an toàn và kiểm tra tính toàn vẹn của thông tin, người ta thường mã hóa thông tin trước khi truyền đi bằng các một số các hệ mật như DES, Triple DES(3DES), RC4, AES, RSA, Rabin, Diffie-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 toán 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 thông tin. Do đó khóa phải được giữ bí mật tuyệt đối. Một số 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: Diffie-Hellman, 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 công khai, ta đang cố gắng thực hiện Giảm độ dài khóa đồng thời vẫn giữ được tính an toàn của hệ mật mã. Đây chính là xu thế của mật mã học hiện nay.

Xuất phát từ mong muốn tìm hiểu, tạo nên một chương 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 phương 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.

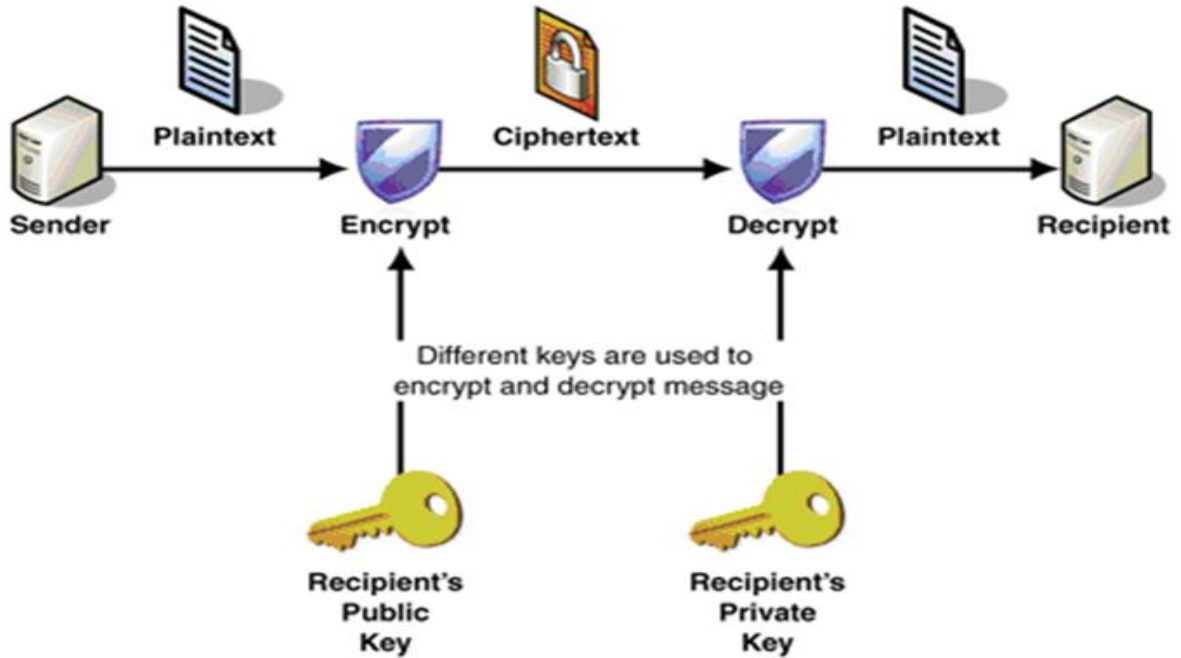
1.4. Giới thiệu chung về hệ mật mã RSA

Ý tưởng về một hệ mật khóa công khai đã được Diffie và Hellman đưa ra vào 1976. Còn việc hiện thực hóa hệ mật khóa công khai thì do Rivest, Shamir và Adleman đưa ra đầu tiên vào 1977 tại Học viện Công nghệ Massachusetts (MIT), họ đã tạo nên hệ mật RSA nổi tiếng. Tên của thuật toán lấy từ 3 chữ cái đầu của tên 3 tác giả.

Đây là thuật toán 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 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:

$$m^{ed} \equiv m \pmod{n}$$

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.

1.5.1. Số học đồng dư (modulo)

- Định nghĩa modulo: Cho số tự nhiên n và số nguyên a . Ta định nghĩa $a \bmod n$ là phần dư dương khi chia a cho n .

VD: $5 \bmod 3 = 2$

- Phép đồng dư ký hiệu $a \equiv b \bmod n$, a và b có phần dư như nhau khi chia cho n .

VD: $5 \bmod 3 = 2$, $8 \bmod 3 = 2 \Rightarrow 5 \equiv 8 \bmod 3$

- Ước số chung lớn nhất (UCLN) của hai hay nhiều số là số lớn nhất trong tập hợp các ước chung. Ước chung lớn nhất của a và b ký hiệu là $\text{GCD}(a, b)$.

VD: $\text{GCD}(6, 36) = 6$ vì $6 \bmod 6 = 0$ và $36 \bmod 6 = 0$

- Số nguyên tố là số tự nhiên chỉ có đúng hai ước số là 1 và chính nó. Các số tự nhiên lớn hơn 1 không phải là số nguyên tố được gọi là hợp số.

VD: 3 là số nguyên tố vì $3 = 3 * 1$

- Nguyên tố cùng nhau: a và b được gọi là nguyên tố cùng nhau nếu

$\text{GCD}(a, b) = 1$

VD: $a = 8, b = 5$ là nguyên tố cùng nhau vì $\text{GCD}(8, 5) = 1$.

1.5.2. Thuật toán Euclid

Định nghĩa thuật toán

- Thuật toán Euclid dùng để tìm ước số chung lớn nhất của hai số nguyên a và b . Ta ký hiệu ước số chung lớn nhất này là $\text{GCD}(a, b)$. Thuật toán này dựa trên định lý sau

- Định lý: với mọi số nguyên $a \geq 0$ và $b > 0$ thì:

$$\text{gcd}(a, b) = \text{gcd}(b, a \bmod b).$$

- Thuật toán Euclid tìm ước chung lớn nhất

Bài toán: Cho 2 số nguyên dương A và B. Tìm ƯCLN(A, B)

If(A=0) return B

If(B=0) return A

While(B>0){

R=A mod B;

A=B; B=R;

}

return A

Nhận xét: Như vậy để tìm ước số chung lớn nhất của 1 cặp số cho trước, ta đưa về bài toán ƯCLN của cặp số gồm số nhỏ hơn gồm thương 2 số đó và phần dư của số lớn hơn khi chia cho số nhỏ hơn. Thuật toán Euclid tạo nên vòng lặp, ở mỗi bước ta áp dụng tính chất trên khi phần dư đó còn khác 0.

Ví dụ vận dụng: Hãy áp dụng thuật toán Euclid tìm ước chung lớn nhất của 270 và 192.

Ta có: $A = 270, B = 192, A \neq 0, B \neq 0$

o Ta có: $270 / 192 = 1$ dư 78

- Theo thuật toán Euclid ta có

o $\text{GCD}(270, 192) = \text{GCD}(192, 78)$

- Tìm $\text{GCD}(192, 78)$

o $A = 192, B = 78, A \neq 0, B \neq 0$

o Ta có: $192 / 78 = 2$ dư 36

- Theo Euclid ta có:

o $\text{GCD}(192, 78) = \text{GCD}(78, 36)$

- Tìm $\text{GCD}(78, 36)$

o $A = 78, B = 36, A \neq 0, B \neq 0$

o Ta có: $78 / 36 = 2$ dư 6

- Theo thuật toán Euclid ta có:

o $\text{GCD}(78, 36) = \text{GCD}(36, 6)$

- Tìm $\text{GCD}(36, 6)$

o $A = 36, B = 6, A \neq 0, B \neq 0$

o Ta có: $36 / 6 = 6$ dư 0

- Theo thuật toán Euclid ta có $\text{GCD}(36, 6) = \text{GCD}(6, 0) = 6$

Vậy $\text{GCD}(270, 192) = 6$

Thuật toán Euclid mở rộng tìm phần tử nghịch đảo

Định nghĩa

Cho $a \in \mathbb{Z}_n$, nếu $\exists b \in \mathbb{Z}_n$ sao cho $a \otimes b = 1 \pmod{n}$. Khi đó b được gọi là phần tử nghịch đảo của a . Ký hiệu: $a^{(-1)} = b$.

Định lý về sự tồn tại của phần tử nghịch đảo

Nếu $\text{GCD}(a, n) = 1$ thì tồn tại duy nhất một số $b \in \mathbb{Z}_n$ là phần tử nghịch đảo của a thỏa mãn $a \otimes b = (a \times b) \pmod{n} = 1$.

Bổ đề Bezout

Cho 2 số nguyên r_0, r_1 , tồn tại 2 số nguyên khác s và t sao cho

$s \times r_0 + t \times r_1 = \text{gcd}(r_0, r_1) = d$. Khi đó $t = r_1^{(-1)}$ theo mod r_0 .

Thuật toán

- Input: r_1, r_0

- Output: $r_1^{(-1)} \pmod{r_0} = ?$ (nếu tồn tại)

1) Dùng thuật toán Euclid mở rộng để tìm các số nguyên tố s và t sao cho

$$s.r_0 + t.r_1 = \text{GCD}(r_0, r_1) = d$$

2) Nếu $d > 1$ thì $r_1^{-1} \bmod r_0$ không tồn tại. Ngược lại nếu $d = 1$ thì return (t) .

Công thức tính s và t :

$$i=0 \quad s_0=1, t_0=0$$

$$i=1 \quad s_1=0, t_1=1$$

$$i \quad s_i = s_{(i-2)} - q_{(i-1)} * s_{(i-1)}, t_i = t_{(i-2)} - q_{(i-1)} * t_{(i-1)}$$

- Trong đó:

$$+ i = 0, 1, 2, 3, \dots$$

$$+ r_i = q_{(i+1)} * r_{(i+1)} + r_{(i+2)}$$

- Thuật toán dừng lại khi $r_{(i+2)} = 0$

Ví dụ vận dụng: Tìm $r_1^{-1} \bmod r_0 = 8^{-1} \bmod 29$

Bước i	r_i	$q_{(i+1)}$	$r_{(i+1)}$	$r_{(i+2)}$	s_i	t_i
0	29	3	8	5	1	0
1	8	1	5	3	0	1
2	5	1	3	2	1	-3
3	3	1	2	1	-1	4
4	2	2	1	0	2	-7
5					-3	11

Bước $i = 0$ có $r_i = r_0 = 29$, $r_{i+1} = r_1 = 8$

Lấy $r_0 = 29$ chia cho $r_1 = 8$ được thương $q_{i+1} = q_1 = 3$, dư $r_{i+2} = r_2 = 5$

Bước $i = 1$ có $r_i = r_1 = 8$, $r_{i+1} = r_2 = 5$

Lấy $r_1 = 8$ chia cho $r_2 = 5$ được thương $q_{i+1} = q_2 = 1$, dư $r_{i+2} = r_3 = 3$

Bước $i = 2$ có $r_i = r_2 = 5$, $r_{i+1} = r_3 = 3$

Lấy $r_2 = 5$ chia cho $r_3 = 3$ được thương $q_{i+1} = q_3 = 1$, dư $r_{i+2} = r_4 = 2$

$$s_2 = s_0 - q_1 \times s_1 = 1 - 3 \times 0 = 1$$

$$t_2 = t_0 - q_1 \times t_1 = 0 - 3 \times 1 = -3$$

Bước $i = 3$ có $r_i = r_3 = 3$, $r_{i+1} = r_4 = 2$

Lấy $r_3 = 3$ chia cho $r_4 = 2$ được thương $q_{i+1} = q_4 = 1$, dư $r_{i+2} = r_5 = 1$

$$s_3 = s_1 - q_2 \times s_2 = 0 - 1 \times 1 = -1$$

$$t_3 = t_1 - q_2 \times t_2 = 1 - 1 \times (-3) = 4$$

Bước $i = 4$ có $r_i = r_4 = 2$, $r_{i+1} = r_5 = 1$

Lấy $r_4 = 2$ chia cho $r_5 = 1$ được thương $q_{i+1} = q_5 = 2$, dư $r_{i+2} = r_6 = 0$

$$s_4 = s_2 - q_3 \times s_3 = 1 - 1 \times (-1) = 2$$

$$t_4 = t_2 - q_3 \times t_3 = (-3) - 1 \times 4 = -7$$

Bước $i = 5$ có $s_5 = s_3 - q_4 \times s_4 = (-1) - 1 \times 2 = -3$

$$t_5 = t_3 - q_4 \times t_4 = 4 - 1 \times (-7) = 11$$

Vậy $t = t_5 = 11 = 8^{-1} \bmod 29$.

1.5.3. Định lý Fermat

Định lý Fermat: Nếu p là một số nguyên tố và a là một số nguyên thỏa mãn điều kiện $\text{GCD}(a, p)=1$ (tức là a và p là số nguyên tố cùng nhau), thì:

$$a^{p-1} \equiv 1 \pmod{p}$$

Trong đó:

- p : số nguyên tố
- a : số nguyên bất kỳ khác bội của p , $\text{GCD}(a, p)=1$

Ví dụ:

- Tính $2^{7-1} \pmod{7}$:
 - Ta thấy 7 là số nguyên tố và $\text{GCD}(2, 7) = 1$.
 - Theo định lý Fermat nhỏ thì:

$$2^{7-1} \pmod{7} = 1$$
 - Kiểm tra lại:

$$2^{7-1} \pmod{7} = 2^6 \pmod{7} = 64 \pmod{7} = 1$$

1.5.4. Hàm số Euler

a. Định nghĩa

- **Hàm phi Euler** của một số nguyên dương n là số các số nguyên tố cùng nhau với n và nhỏ hơn n

- Ký hiệu: $\phi(n)$.

b. Tính chất của hàm phi Euler

- Tính chất 1: Nếu n là số nguyên tố thì $\phi(n) = n - 1$.
- Tính chất 2: Nếu p và q là hai số nguyên tố cùng nhau thì $\phi(p \times q) = \phi(p) \times \phi(q)$.
- Tính chất 3: Nếu p là số nguyên tố thì $\phi(p^n) = p^n - p^{n-1}$.

c. Định lý Euler

- Nếu a, m là nguyên tố cùng nhau thì:

$$a^{\phi(m)} \equiv 1 \pmod{m}$$

Ví dụ vận dụng

- Cho $m = 6, a = 5$:
 - $\text{GCD}(5, 6) = 1$.
 - Theo định lý Euler:

$$5^{\phi(6)} \equiv 1 \pmod{6}$$
- Chứng minh:
 - $\phi(6) = \phi(3 \times 2)$.
 - Ta có: $\text{GCD}(3, 2) = 1$.
 - Theo tính chất 2, ta có:

$$\phi(3 \times 2) = \phi(3) \times \phi(2)$$
 - Vì 3 và 2 là số nguyên tố nên theo tính chất 1:

$$\phi(3) = 3 - 1 = 2$$

$$\phi(2) = 2 - 1 = 1$$

1.5.5. Thuật toán Miller-Rabin

Thuật toán Miller-Rabin là thuật toán kiểm tra số nguyên tố dựa trên phương pháp chứng minh theo xác suất, có độ chính xác khá cao ngay cả khi đầu vào là số Carmichael.

Định lý

Giả sử p là một số nguyên tố lẻ. Gọi k, m là hai số thỏa mãn $p - 1 = 2^k \times m$ trong đó m là một số lẻ. Gọi $1 \leq a \leq p$. Ta có:

$$a^m \equiv 1 \pmod{p} \text{ hoặc}$$

Tồn tại $i, 1 \leq i \leq k$, thỏa mãn :

- $a^{2^i \times m} \equiv 1 \pmod{p}$ và
- $a^{2^{i-1} \times m} \equiv -1 \pmod{p}$

Giải thuật kiểm tra tính nguyên tố của một số nguyên p

Miller_Rabin(p):

1. Tìm k, q với $k > 0, q$ lẻ thỏa mãn $p = 2^k \cdot q + 1$.
2. Chọn số ngẫu nhiên a trong khoảng $[2, p - 1]$.
3. Nếu $a^q \pmod{p} = 1$:
 - return "p có thể là số nguyên tố".
4. For $j = 0$ to $k - 1$:
 - Nếu $a^{2^j \cdot q} \pmod{p} \neq 1$ và $a^{2^j \cdot q} \pmod{p} \neq p - 1$:
 - return "p không phải là số nguyên tố".
5. return "p có thể là số nguyên tố".

Xác suất sai và nguyên tắc kiểm tra:

- Người ta đã tính được xác suất để trong trường hợp p là hợp số, thuật toán Miller-Rabin đưa ra khẳng định "không phải là số nguyên tố" là 75%. Trong 25% còn lại, Miller_Rabin không xác định được p là nguyên tố hay hợp số.
 - Nếu chúng ta áp dụng thuật toán ttt lần (mỗi lần với các giá trị aaa khác nhau) thì xác suất không xác định (trong cả t lần) là 0.25^t Với $t=10$, xác suất trên là rất bé, nhỏ hơn 0.000001 .

Tóm lại nguyên tắc kiểm tra tính nguyên tố của số nguyên p thực hiện như sau:

1. Thực hiện thuật toán Miller-Rabin 10 lần với 10 số a ngẫu nhiên khác nhau.
2. Nếu cả 10 lần thuật toán cho ra kết quả "có thể là số nguyên tố", thì ta khẳng định p là số nguyên tố.
3. Chỉ cần một lần thuật toán cho ra kết quả "không phải là số nguyên tố" thì ta khẳng định p là hợp số.

Ví dụ vận dụng

$$p = 41 = 23 * 5 + 1 \text{ do đó } k = 3, q = 5, p-1 = 40$$

a	$a^q \bmod p$	$a^{2q} \bmod p$	$a^{4q} \bmod p$
7	38	9	40
8	9	40	
9	9	40	
12	3	9	40
13	38	9	40
16	1		
24	14	32	40
25	40		
31	40		
37	1		

Vậy 41 là số nguyên tố.

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

RSA hoạt động theo các bước cơ bản như sau:

Mỗi thực thể cần chia sẻ khóa công khai với nhau và thiết lập cặp khóa của riêng mình. Để thông tin liên lạc được đảm bảo bảo mật, hai thực thể cần giữ bí mật các chìa khóa riêng tư.

Sau khi người gửi đã có được chìa khoá riêng tư của người nhận, họ có thể mã hoá dữ liệu mà họ muốn bảo mật bằng chìa khoá này. Tuy nhiên, khi các dữ liệu đã được mã hoá bằng khoá công khai thì chỉ có khoá riêng tư đi kèm với khoá công khai mới có thể giải mã chúng. Tất cả những điều này được hình thành do chức năng của đây.

Khi người nhận nhận được tin nhắn đã được mã hoá, họ có thể truy cập vào dữ liệu bằng cách sử dụng khoá riêng của mình. Nếu người nhận muốn đảm bảo mức độ an toàn cho việc liên lạc thì họ có thể sử dụng khoá công khai của bên mà mình liên lạc để mã hoá tin nhắn. Và tất nhiên, cách duy nhất để đối phương truy cập được vào thông tin bị mã hoá bằng khoá công khai là sử dụng chìa khoá riêng tư phù hợp.

Bằng cách này, mã hoá RSA được sử dụng khi các bên không biết gửi dữ liệu một cách an toàn.

1.6.1. Thuật toán tạo khóa

- B1: Chọn 2 số nguyên tố p, q đủ lớn
- B2: Tính $n = p * q$, 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 :

$$\Phi(n) = (p - 1)(q - 1)$$

Giá trị này sẽ được giữ bí mật

- B4: Chọn ngẫu nhiên khóa mã hóa b thỏa mãn $1 < b < \Phi(n)$

–

sao cho $GCD(b, \Phi(n)) = 1$

- B5: Tìm khóa giải mã a sao cho

$$a \equiv b^{-1} \text{ mod } \Phi(n)$$

(tức a là nghịch đảo modulo của b theo $\Phi(n)$)

- Khi đó khóa công khai và khóa riêng tư có dạng :

$$K_{pub} = \{b, n\} \text{ và } K_{pr} = \{a, p, q\}$$

1.6.2. Thuật toán mã hóa

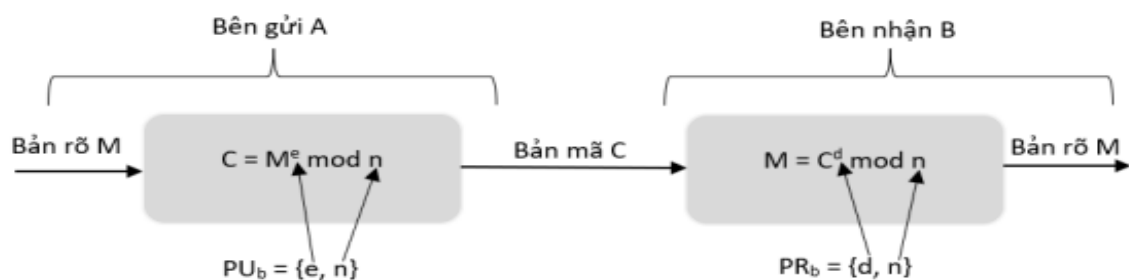
Để mã hóa thông tin M ta làm theo các bước:

- 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 $y = x^b \bmod n$. 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:

- B1: Tính $x = y^a \bmod n$. Khi đó x chính là bản rõ ban đầu



Hình 3. Mã hóa và giải mã RSA ứng dụng cho bảo mật

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 bit.

- 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 hàu như không thể với hai số nguyên tố 2048 bit .

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 kiến thức.
- Tìm hiểu thuật toán.
- Thiết kế và cài đặt chương trình.

- 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 $n = p * q$, $\Phi(n) = (p - 1)(q - 1)$
- Tính b sao cho $1 < b < \Phi(n)$ sao cho $GCD(b, \Phi(n)) = 1$
- Tính $a \equiv b^{-1} \text{ mod } \Phi(n)$

Từ đó ta tạo được khóa :

- $K_{pub} = \{b, n\}$
- $K_{pr} = \{a, p, q\}$

Bước 2: Mã hóa

- Gọi phần tử của bản rõ là x , phần tử của bản mã là y .
- Khi đó $y = x^b \text{ mod } n$

Bước 3: Giải mã

- $x = y^a \text{ mod } n$

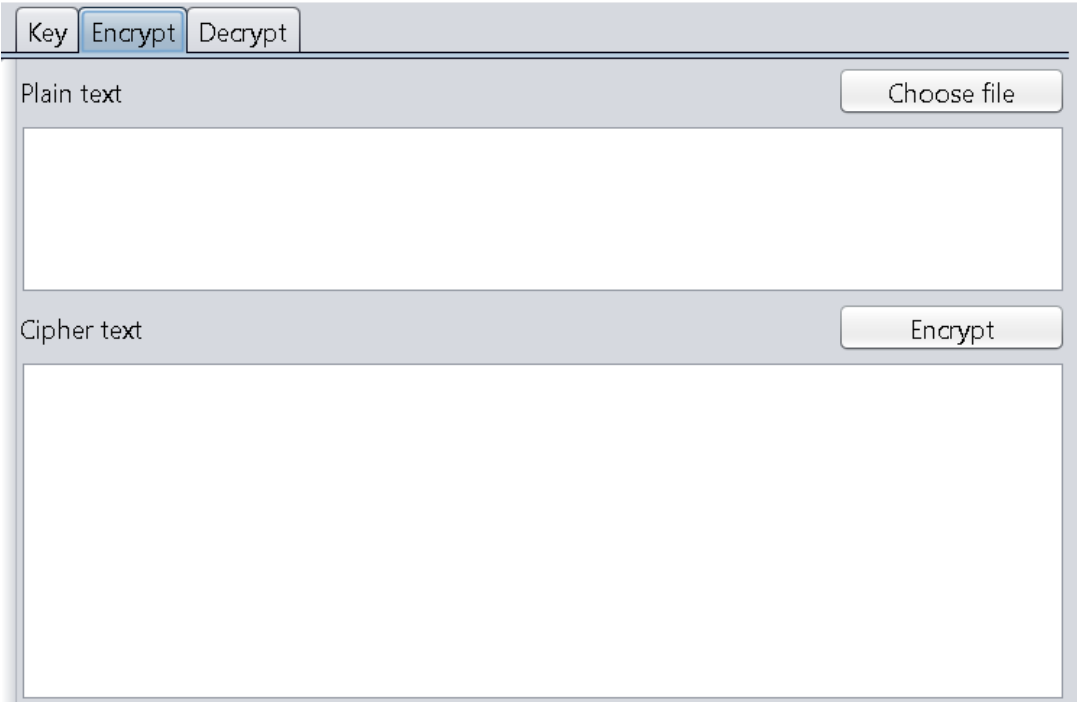
2.3. Thiết kế, cài đặt chương trình demo thuật toán

2.3.1. Giao diện chương trình demo theo ngôn ngữ Java



The screenshot shows a Java application window for key generation. At the top, there are three tabs: "Key" (selected), "Encrypt", and "Decrypt". Below the tabs, the interface is divided into two main sections. The top section is labeled "Public key" and contains a large empty text area. To the right of this section is a green button labeled "Generate key". The bottom section is labeled "Private key" and also contains a large empty text area. To the right of this section, there is a label "Key Size:" followed by a dropdown menu currently showing "2048".

Hình 4: Giao diện tạo khóa bằng ngôn ngữ Java



The screenshot shows a Java application window for encryption. At the top, there are three tabs: "Key", "Encrypt" (selected), and "Decrypt". Below the tabs, the interface is divided into two main sections. The top section is labeled "Plain text" and contains a large empty text area. To the right of this section is a button labeled "Choose file". The bottom section is labeled "Cipher text" and contains a large empty text area. To the right of this section is a button labeled "Encrypt".

Hình 5: Giao diện mã hóa bằng ngôn ngữ Java

Hình 6: Giao diện giải mã bằng ngôn ngữ Java

2.3.2. Giao diện chương trình demo theo ngôn ngữ JavaScript

Hình 7: Giao diện tạo khóa bằng ngôn ngữ JavaScript

Hình 8: Giao diện mã hóa bằng ngôn ngữ JavaScript

Enter ciphertext

Decrypt

Decrypted Text:

Hình 9: Giao diện giải mã bằng ngôn ngữ JavaScript

2.3.3. Giao diện chương trình demo theo ngôn ngữ PHP

CHƯƠNG TRÌNH MÃ HÓA VÀ GIẢI MÃ RSA

Nhập văn bản cần mã hóa hoặc chọn file Không có tệp nào được chọn

PRIVATE KEY

ENCRYPT

PUBLIC KEY

DECRYPT

Hình 10: Giao diện chương trình demo bằng ngôn ngữ PHP

2.3.4. Giao diện chương trình demo theo ngôn ngữ C#

The screenshot shows a Windows application window titled "MÃ HÓA VÀ GIẢI MÃ RSA". The main title "RSA" is centered at the top. The interface is divided into three main sections: "Sinh Khóa" (Generate Key), "Mã Hóa" (Encryption), and "Giải Mã" (Decryption).

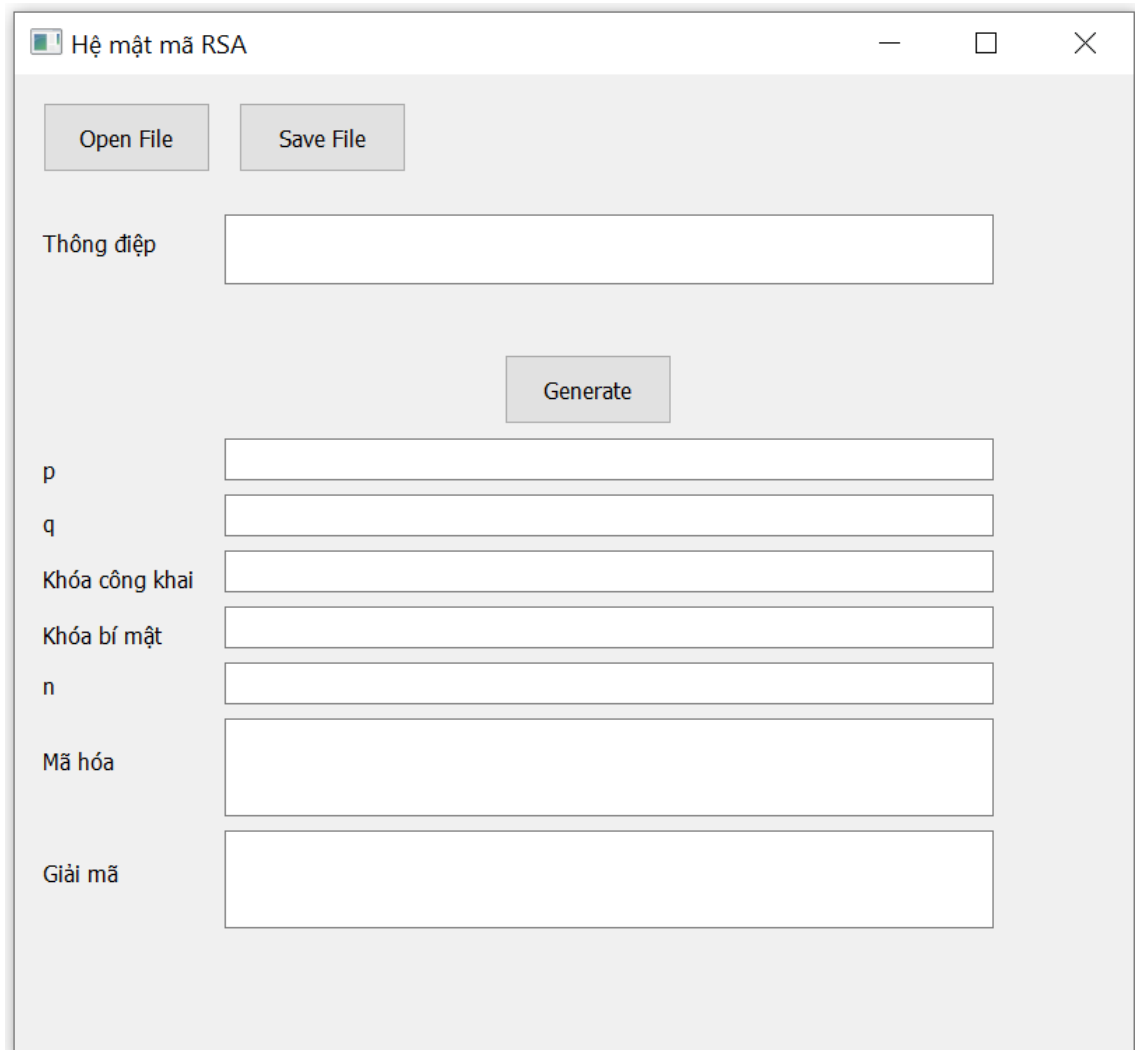
Sinh Khóa (Generate Key): This section contains input fields for "P =", "Q =", "Khóa bí mật :", and "Khóa công khai :". There is a checkbox labeled "sinh khóa tự động" (generate key automatically) and a button labeled "Lấy Khóa" (Get Key).

Mã Hóa (Encryption): This section has a "Bản Rõ :" (Plaintext) label above a text area. To the right is a button "Chọn tài liệu" (Select file). Below the text area is a "Mã Hóa" (Encrypt) button. At the bottom, there is a "Bản Mã :" (Ciphertext) label above another text area, with a "Ghi Bản Mã Hóa" (Save Encrypted Text) button to its right.

Giải Mã (Decryption): This section has a "Bản Mã :" (Ciphertext) label above a text area. To the right is a button "chọn file mã hóa" (select encryption file). Below the text area is a "Giải Mã" (Decrypt) button. At the bottom, there is a "Bản Rõ :" (Plaintext) label above another text area, with a "Ghi Bản Rõ" (Save Plaintext) button to its right.

Hình 11: Giao diện chương trình demo bằng ngôn ngữ C#

2.3.5. Giao diện chương trình demo theo ngôn ngữ Python



The screenshot shows a window titled "Hệ mật mã RSA" (RSA Cryptosystem). It contains the following elements:

- Buttons: "Open File" and "Save File" at the top left.
- Text input field: "Thông điệp" (Message) with a corresponding text box.
- Button: "Generate" centered below the message field.
- Text input fields: "p", "q", "Khóa công khai" (Public Key), "Khóa bí mật" (Private Key), and "n", each with a corresponding text box.
- Text input field: "Mã hóa" (Encryption) with a corresponding text box.
- Text input field: "Giải mã" (Decryption) with a corresponding text box.

Hình 12: Giao diện chương trình demo bằng ngôn ngữ Python

2.4. Cài đặt và triển khai

2.4.1. Giới thiệu công cụ

2.4.1.1. Giới thiệu công cụ Visual Studio Code

Visual Studio Code là một trình soạn thảo mã nguồn được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có chức năng nổi bật cú pháp, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn.

Với giao diện đơn giản, dễ dàng tùy biến theo nhu cầu của ngôn ngữ Python PHP, JavaScript... giúp người dùng lập trình một cách dễ dàng ví dụ như lập trình ứng dụng bằng Python.

2.4.1.2. Giới thiệu công cụ Eclipse

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

Với giao diện đơn giản, dễ sử dụng, tích hợp đầy đủ các thư viện của ngôn ngữ Java giúp người lập trình sử dụng các API một cách dễ dàng. Ngoài 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 , 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

2.4.1.3. Giới thiệu công cụ Visual Studio 2022

Visual Studio 2022 là một môi trường phát triển tích hợp (IDE) được phát triển bởi Microsoft, dành cho việc phát triển các ứng dụng trên Windows, Android, iOS, web và đám mây. Đây là phiên bản Visual Studio đầu tiên được xây dựng như một ứng dụng 64-bit, mang lại hiệu suất cải thiện đáng kể và khả năng xử lý tốt hơn cho các dự án lớn. Visual Studio 2022 cung cấp các tính năng mạnh mẽ như hoàn thành mã thông minh, gỡ lỗi nâng cao, và tích hợp với Git.

Với giao diện hiện đại, dễ dàng tùy biến theo nhu cầu của người dùng, Visual Studio 2022 hỗ trợ lập trình trên nhiều ngôn ngữ như C++, C#, .NET, Python, và JavaScript, giúp người dùng phát triển ứng dụng một cách hiệu quả và nhanh chóng.

- Những điểm mạnh của Visual Studio:
 - ❖ Visual Studio hỗ trợ lập trình trên nhiều ngôn ngữ như: C/C++, C#, F#, Visual Basic, HTML, CSS, JavaScript.
 - ❖ Là một công cụ hỗ trợ việc Debug một cách dễ dàng và mạnh mẽ như: Breakpoint, 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.
 - ❖ 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

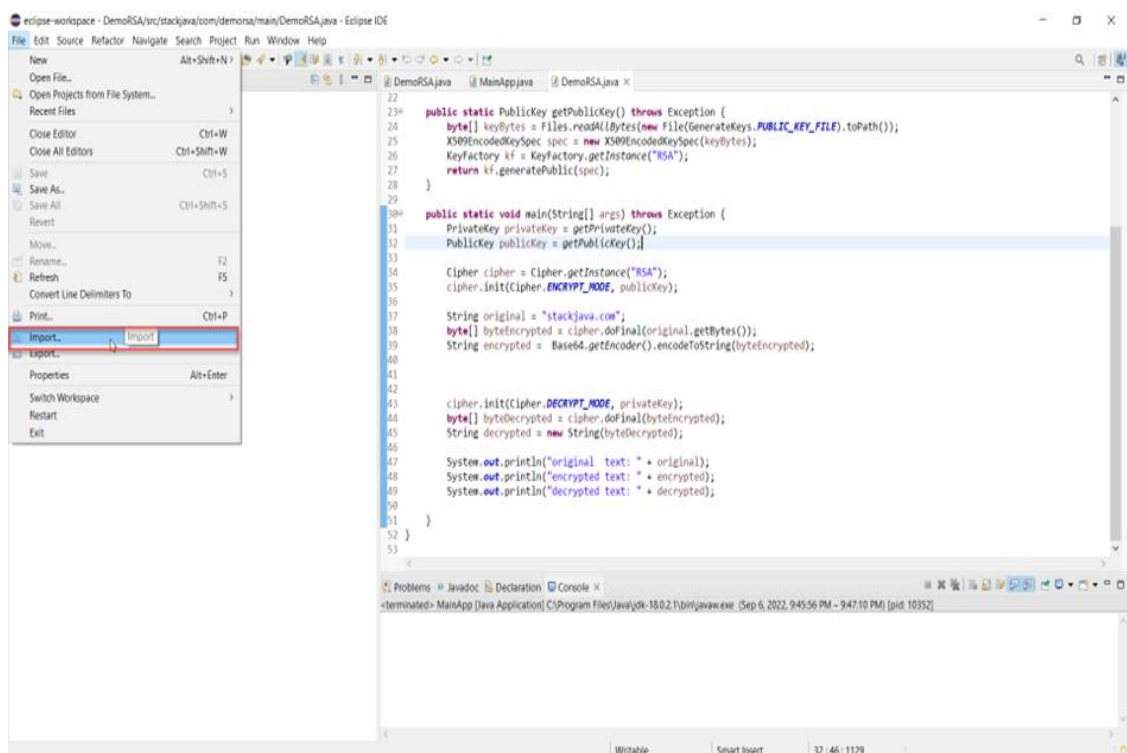
2.4.2.1. Cài đặt Eclipse và chạy chương trình demo

a. Cài đặt Eclipse

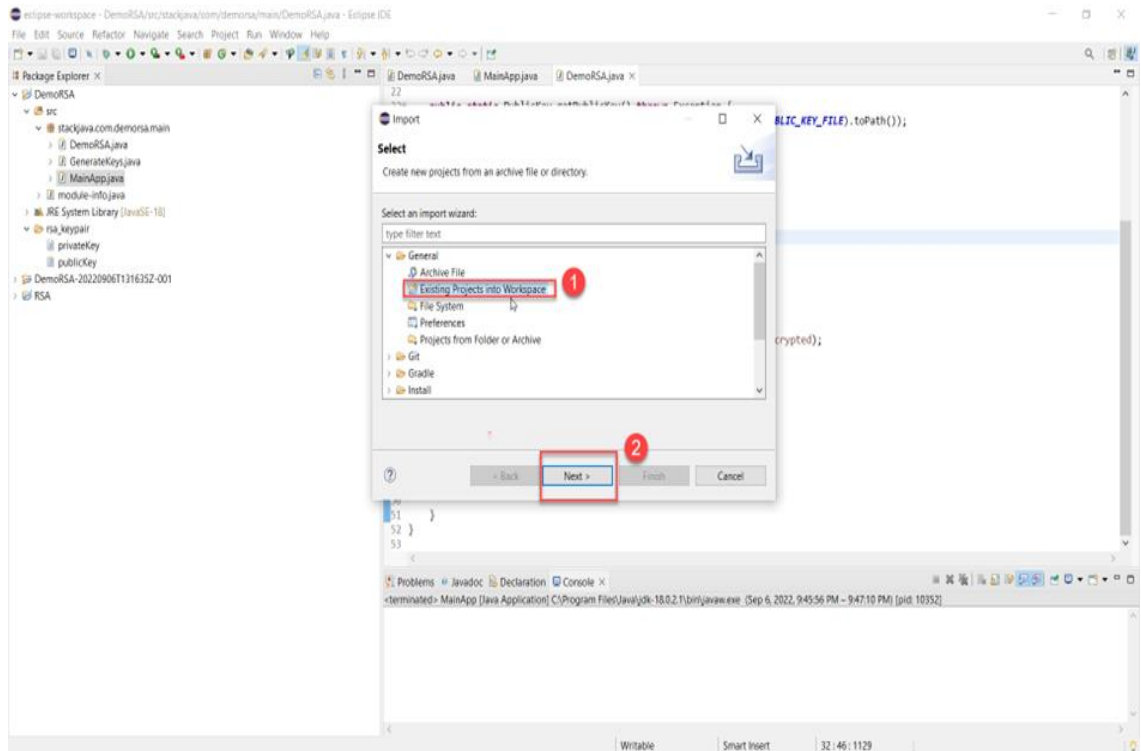
- Bước 1: Truy cập trang <https://www.oracle.com/> để tải JDK và công cụ Eclipse IDE về máy
- Bước 2: Mở file sau khi đã tải thành công
- Bước 3: Ấn theo hướng dẫn cài đặt của phần mềm
- Bước 4: Thiết lập biến môi trường
- Bước 5: Mở công cụ và hoàn tất cài đặt

b. Chạy chương trình demo

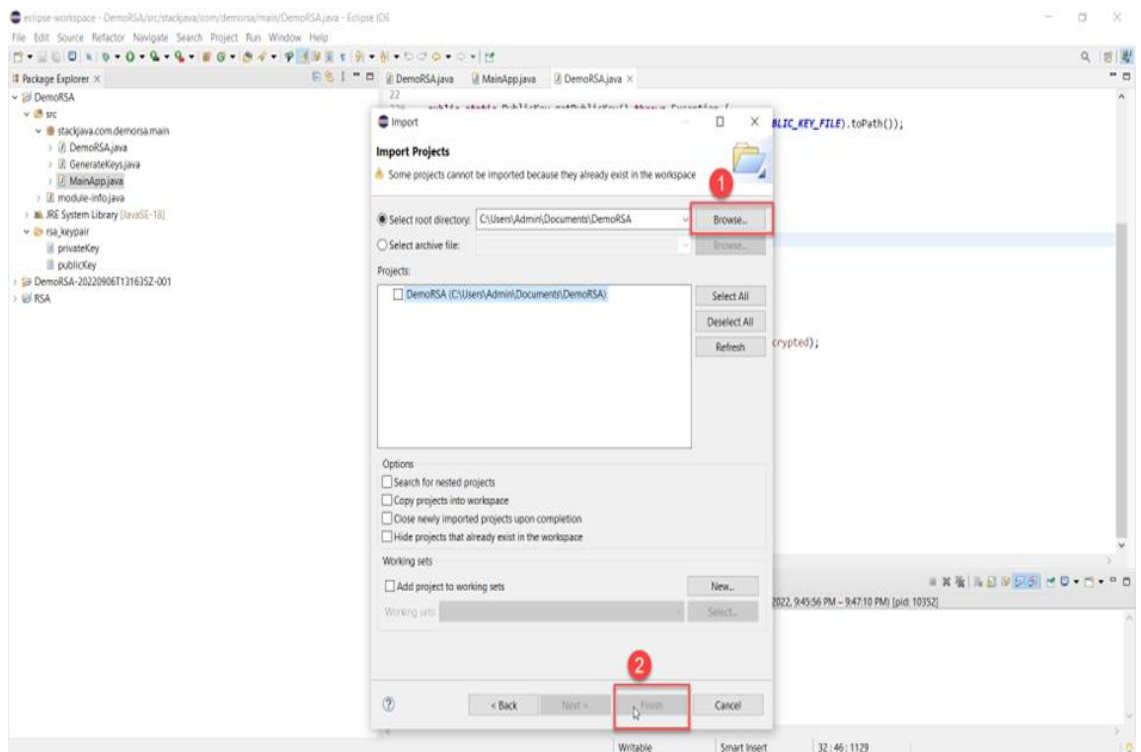
- **Bước 1:** Ở giao diện làm việc của Eclipse IDE, vào **File** chọn **Import**.



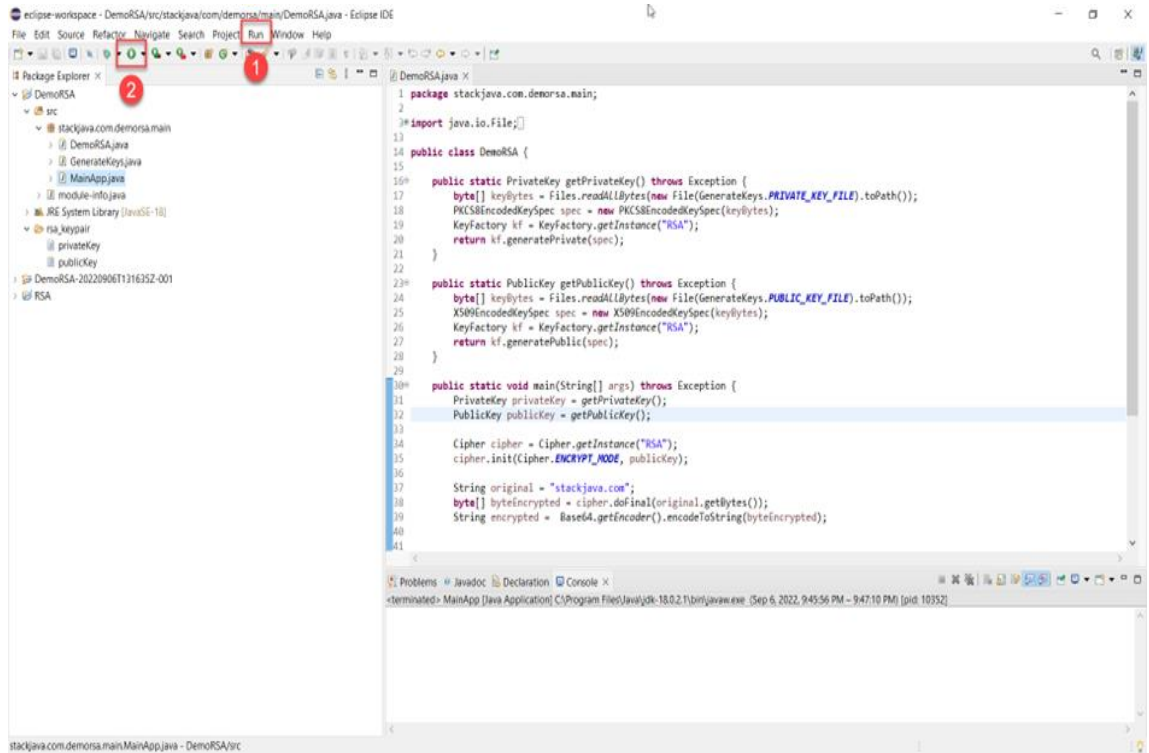
- **Bước 2:** Ở hộp thoại chọn vào **Existing Projects into Workspace** và chọn **Next**. Hộp thoại tiếp theo sẽ mở ra.



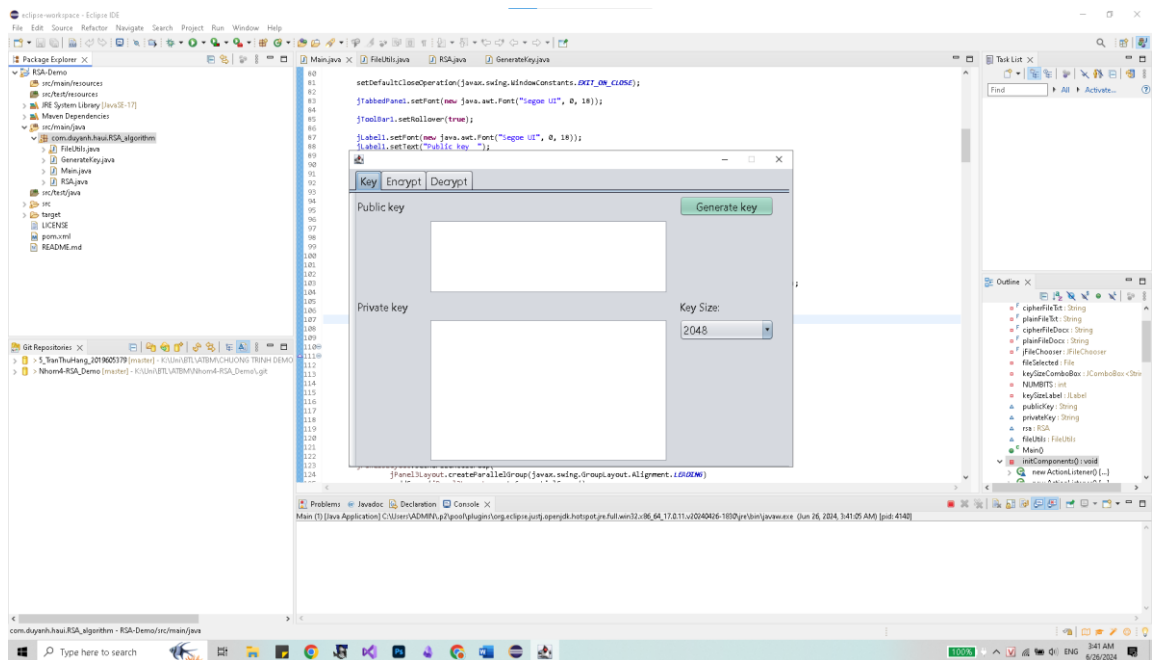
- **Bước 3:** Ở hộp thoại tiếp theo chọn **Select root directory** và trở đến thư mục project. Chọn **Finish**.



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



Giao diện chương trình



2.4.2.2. Cài đặt Visual Studio 2022 và chạy chương trình demo

a. Cài đặt Visual Studio 2022

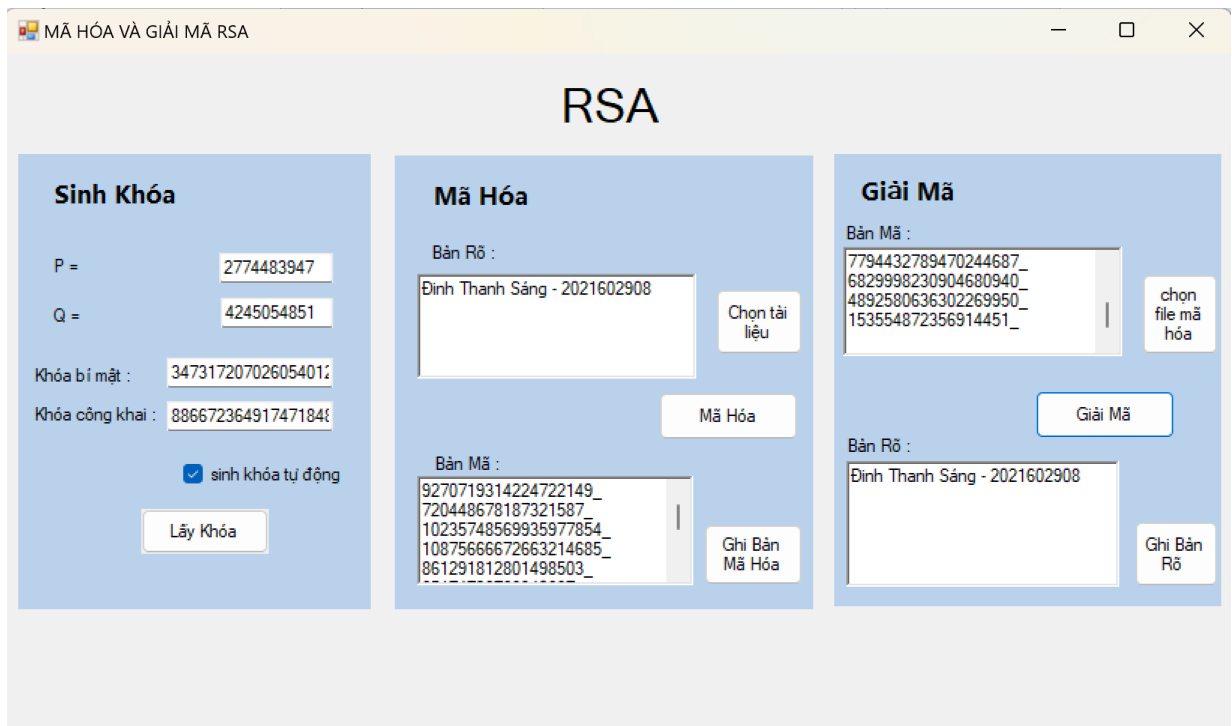
B1: Tải Visual Studio 2022 phiên bản Community

B2: Mở file vừa tải và làm theo hướng dẫn cài đặt của phần mềm

B3: Tích chọn những IDE hỗ trợ ngôn ngữ lập trình C# và hoàn tất cài đặt

B4: Mở công cụ và chạy thử nghiệm một chương trình

b. Chạy chương trình demo



2.4.2.3. Cài đặt Visual Studio Code và chạy chương trình demo

a. Cài đặt Visual studio code

B1: Tải các file cài đặt XAMPP và công cụ Visual studio code về máy

B2: Mở file vừa tải và làm theo hướng dẫn cài đặt của phần mềm

B3: Thiết lập các cổng trong xampp

B4: Mở công cụ và hoàn tất cài đặt

b. Chạy chương trình demo

CHƯƠNG TRÌNH MÃ HÓA VÀ GIẢI MÃ RSA

vũ thị thanh lan Chọn tệp Không có tệp nào được chọn

đọc văn bản

Tạo khóa

PRIVATE KEY

```
-----BEGIN PRIVATE KEY-----
MIICdQIBADANBgkqhkiG9w0BAQEFAASCA18wggJbAgEAAoGBA00W
zFxGutwOjs81
YXZMsjwN6zt5fkwPYvG2n9J9Fzf7crVLLwk414eNCAkTAX1n94Ly
KRFxu1p7BfFM
I/F9hDBUF+tTqmmR4jOfdyxPAvJidf0xi3DqtFLjYFC4R056Lohq
7049SrCmHQgx
QUz5tEF1Ry7ITLi3qUzQkKcFLFuFgMBAAECgYAYKYKve1+Js6LP
gHHQf8WuIxxvE
B56LJ55ACe2isUi6VX8dkHyxLM+dJF+w286S4+5gTDAY0sHzEnhf
-----END PRIVATE KEY-----
```

ENCRYPT

Ma hoa

```
aHP9c8KGZD+U+39SeTwN3OKw3nZv11quFG3yZ74OKt8Etkqh8rmhuU
90HDNSTOZmxp1an38dQm371y4pAdDexNzpeqdMRWw38zINyctHIdAS
tkYKTwcV6P+AmVgYfeeCnPmWFZcH9bMe!V7AzBw8JyYNRZEMR/tnyh
AGLRgOrpI=
```

PUBLIC KEY

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCsQgSIB3DQEBAQUAA4GNADCBiQKBgQDfSxcRrrcDo7PNW
F2TLI8Des7
eX5MD2Lxtp/SfRc3+3K1Sy8JONeHjQgJEwF9Z/eC8ikRcbtaewXxTC
PxYQwVBfr
U6ppkeIzn3csTwLyYnX9MVtw6rXy42BQuEdOei6IauzuPUqwph0IMU
FM+bRBZUcu
yEy4t6lM0JCnHyxVHwIDAQAB
-----END PUBLIC KEY-----
```

DECRYPT

Giai ma

vũ thị thanh lan

2.4.2.4. Cài đặt Visual Studio Code và chạy chương trình demo

a. Cài đặt Visual studio code

B1: Tải xuống và cài đặt VSCode về máy theo hướng dẫn của chương trình.

B2: Cài đặt biến môi trường (Environment Path) Python 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.

b. Chạy chương trình demo

B1: Ở giao diện của VSCode, vào file chọn open folder rồi chọn folder cần chạy.

B2: Nhấn phím F5 để bắt đầu chạy chương trình.

B3: - Nhập thông điệp:

- + Ở giao diện chương trình, nhập thông điệp cần mã hóa và giải mã vào ô “Thông điệp”.

Thông điệp	<input type="text"/>
------------	----------------------

- + Kích nút “Generate” để tiến hành sinh khóa, mã hóa và giải mã.

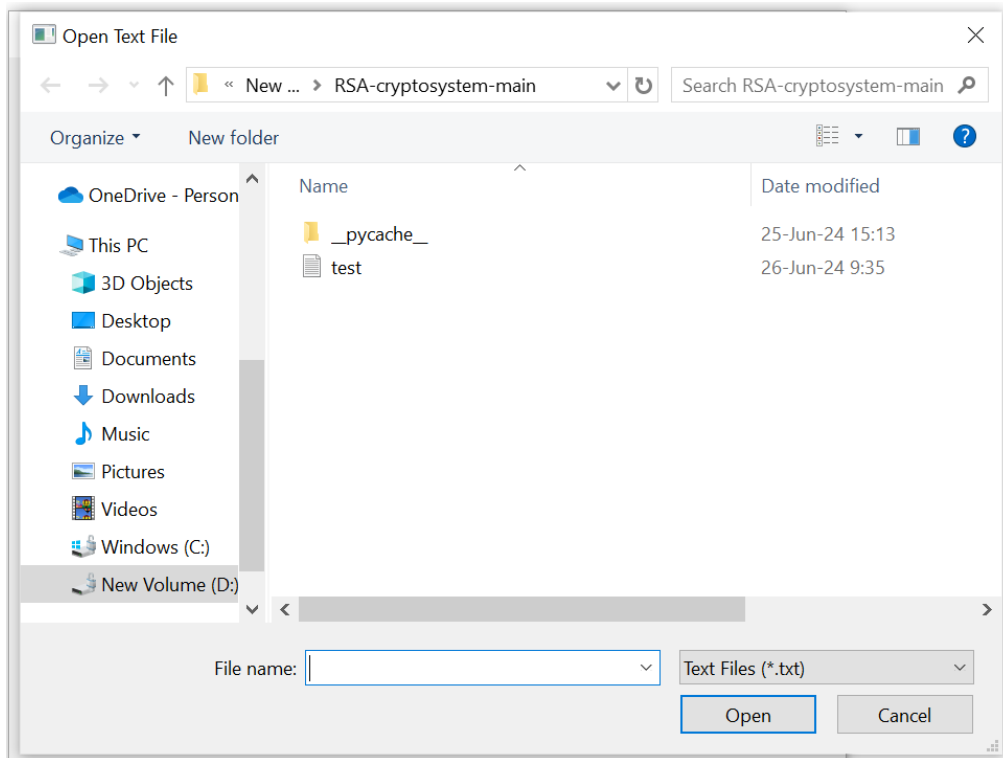
Thông điệp	<input type="text" value="thong diep"/>	
	<input type="button" value="Generate"/>	
p	<input type="text" value="2921783119"/>	
q	<input type="text" value="4204222577"/>	
Khóa công khai	<input type="text" value="(3719263511, 12283826553997277663)"/>	
Khóa bí mật	<input type="text" value="(11438635786766924807, 12283826553997277663)"/>	
n	<input type="text" value="12283826553997277663"/>	
Mã hóa	<input type="text" value="9041518961327307794 9738943428271703892"/> <input type="text" value="8425884606287587311 5260387400631610372"/> <input type="text" value="7327393428931398860 7778258184750451242"/> <input type="text" value="44866464873538363847 4886875264738583888"/>	
Giải mã	<input type="text" value="thong diep"/>	

- Chọn file:

- + Ở giao diện chương trình, kích nút “Open file”.

<input type="button" value="Open File"/>
--

- + Chọn file text cần đọc



+ Kích nút “Generate” để tiến hành sinh khóa, mã hóa và giải mã.

Hệ mật mã RSA

Open File Save File

Thông điệp: Lam Hoang An

Generate

p: 3433100651

q: 2829379913

Khóa công khai: (3679783591, 9713546021246623363)

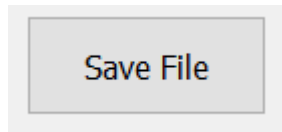
Khóa bí mật: (5396981603528062311, 9713546021246623363)

n: 9713546021246623363

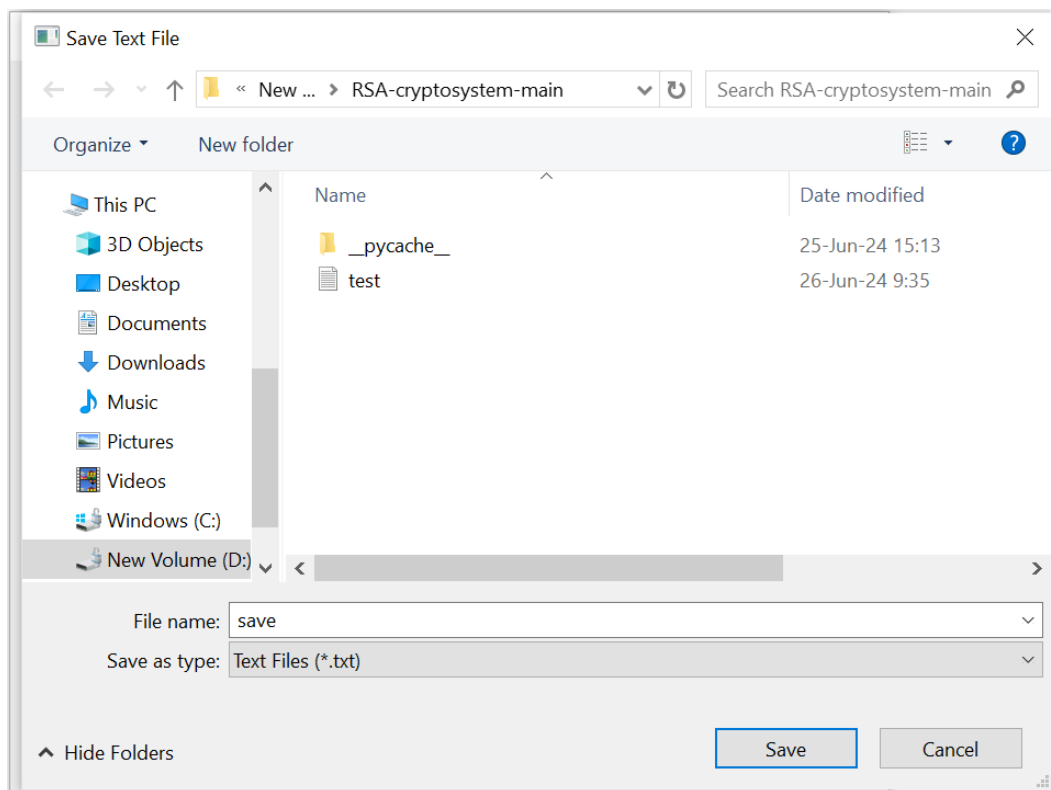
Mã hóa: 2685271012031258838 9132191770046259341
9302942035114500951 488987782288243077 6466655914419124385
3570835084040296113 9132191770046259341
6273848376734886558 4833433886835883544 488887782288243077

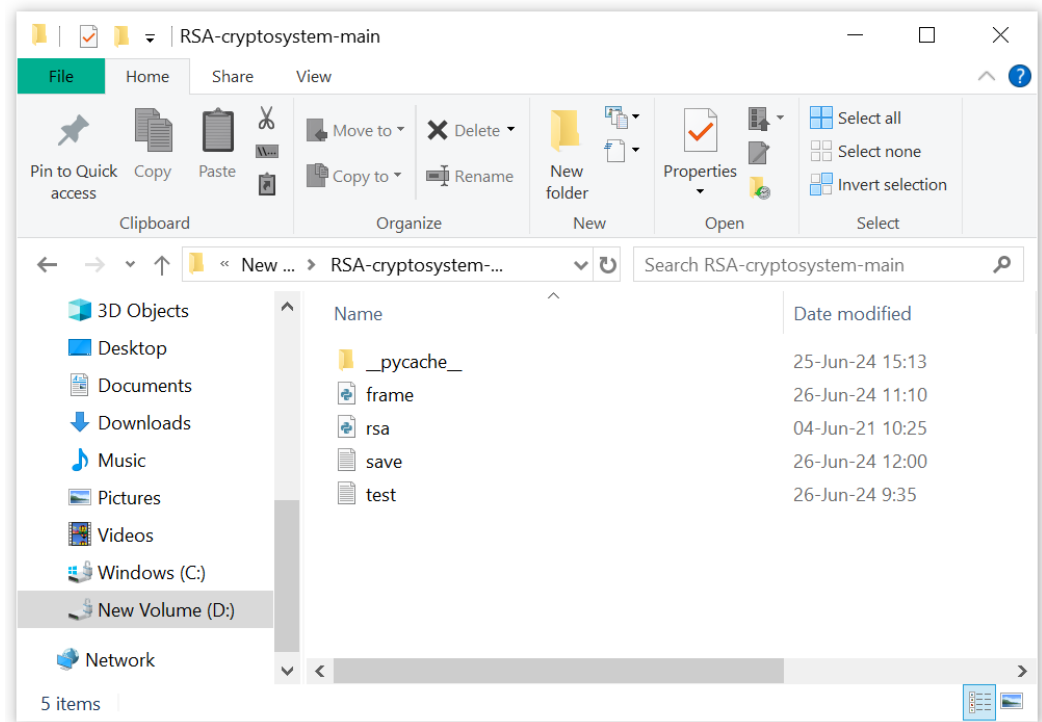
Giải mã: Lam Hoang An

- Lưu file:
 - + Để lưu thông số chương trình đang chạy, kích vào nút “Save file”



- + Đặt tên rồi lưu file





2.5. Thực hiện bài toán

2.5.1. Phân công công việc

Tên sinh viên	Tên công việc
Đinh Đăng Duy Anh	<ul style="list-style-type: none"> - Tìm hiểu về quá trình tạo khóa - Tìm hiểu về độ an toàn của mã hóa RSA - Tìm hiểu thuật toán Miller-Rabin - Cài đặt các chức năng: <ul style="list-style-type: none"> • Tạo khóa • Mã hóa • Giải mã - Viết chương trình demo bằng Java

Nguyễn Phương Long	<ul style="list-style-type: none"> - Tìm hiểu các hệ mật mã và hệ mật mã công khai - Tìm hiểu thuật toán Euclid - Cài đặt các chức năng: <ul style="list-style-type: none"> ● Tạo khóa ● Mã hóa ● Giải mã - Viết chương trình demo bằng JavaScript
Vũ Thanh Lan	<ul style="list-style-type: none"> - Tìm hiểu ưu và nhược điểm hệ mã hóa RSA - Tìm hiểu phần Số học đồng dư (modulo) - Cài đặt các chức năng: <ul style="list-style-type: none"> ● Tạo khóa ● Mã hóa ● Giải mã - Viết chương trình demo bằng PHP
Đinh Thanh Sáng	<ul style="list-style-type: none"> - Tìm hiểu thuật toán tạo khóa hệ mã hóa RSA - Nghiên cứu định lý Fermat - Cài đặt các chức năng: <ul style="list-style-type: none"> ● Tạo khóa ● Mã hóa ● Giải mã - Viết chương trình demo bằng C#

Lâm Hoàng An	<ul style="list-style-type: none"> - Tìm hiểu thuật toán mã hóa và giải mã hệ mã hóa RSA - Tìm hiểu hàm số Euler - Cài đặt các chức năng: <ul style="list-style-type: none"> • Tạo khóa • Mã hóa • Giải mã - Viết chương trình demo bằng Python
--------------	---

2.5.2. Cài đặt chương trình

- Hàm kiểm tra số nguyên tố

```
function isPrime(a) {
  if (a <= 1) return false;
  for (let i = 2; i <= Math.sqrt(a); i++) {
    if (a % i === 0) return false;
  }
  return true;
}
```

- Hàm tính modular multiplicative inverse của a modulo b sử dụng thuật toán Euclid

```
function modularInverse(a, n) {
    let t = 0, newT = 1, r = n, newR = a;
    while (newR !== 0) {
        let quotient = Math.floor(r / newR);
        [t, newT] = [newT, t - quotient * newT];
        [r, newR] = [newR, r - quotient * newR];
    }
    if (r > 1) return -1;
    if (t < 0) t += n;
    return t;
}
```

- Hàm tính $x^n \bmod m$.

```
function modExp(base, exp, mod) {
    let result = 1;
    base = base % mod;
    while (exp > 0) {
        if (exp % 2 === 1) result = (result * base) % mod;
        exp = Math.floor(exp / 2);
        base = (base * base) % mod;
    }
    return result;
}
```

- Thuật toán sinh khóa

```
function generatePrime() {
    let p = Math.floor(Math.random() * 10000);
    while (!isPrime(p)) {
        p = Math.floor(Math.random() * 10000);
    }
    return p;
}
```

```
function generateKeys() {
    p = generatePrime();
    q = generatePrime();
    while (p === q) {
        q = generatePrime();
    }
    n = p * q;
    phi = (p - 1) * (q - 1);

    e = Math.floor(Math.random() * 1000);
    while (e === q || e === p || gcd(e, phi) !== 1 || e > phi) {
        e = Math.floor(Math.random() * 1000);
    }
    d = modularInverse(e, phi);

    document.getElementById("publicKey").innerText = `(${e}, ${n})`;
    document.getElementById("privateKey").innerText = `(${d}, ${n})`;
}
```

- Mã hóa sử dụng khóa bí mật

```
function rsaEncrypt(text, e, n) {
    return text.split('')
        .map(char => modExp(char.charCodeAt(0), e, n))
        .join(' ');
}
```

- Giải mã sử dụng khóa công khai

```
function rsaDecrypt(cipher, d, n) {
    return cipher.split(' ')
        .map(code => String.fromCharCode(modExp(parseInt(code), d, n)))
        .join('');
}
```

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 lĩnh hội

- Thuật toán Euclid
- Định lý Fermat
- Hàm số Euler
- Thuật toán Miller-Rabin
- Hệ mã khóa công khai 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 tiếp thu

- 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 với các thuật toán liên quan như thuật toán nghịch đảo của phép nhân modulo hay thuật toán bình phương 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. Bài học kinh nghiệm

- 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 dồi 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 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 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

- 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 kế 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.

KẾT LUẬN

Sau thời gian tìm hiểu và nghiên cứu, nhóm chúng em đã xây dựng thành công chương trình “Mã hóa và giải mã RSA”. Đây là một trong những hệ mã hóa phổ biến ở Việt Nam cũng như trên thế giới. Vì lý do đó, nhóm đã lựa chọn đề tài với mong muốn góp phần đưa ra các giải pháp hữu ích có thể ứng dụng vào thực tế.

Trong quá trình thực hiện đề tài, chúng em đã gặp không ít khó khăn vì đề tài đòi hỏi nhiều kiến thức chuyên ngành, các thành viên chưa có nhiều kinh nghiệm trong việc xây dựng ứng dụng nên việc tìm hiểu, nghiên cứu và xây dựng còn chậm. Ngoài ra, do lịch học khác nhau nên các thành viên gặp khó khăn khi muốn liên lạc và trao đổi công việc. Tuy nhiên nhờ việc thu thập được nhiều tài liệu để tìm hiểu và nghiên cứu cũng như nhận được sự góp ý, hướng dẫn nhiệt tình của giảng viên, chúng em đã có thể hoàn thành tốt nhất bài báo cáo này.

Qua quá trình học tập và làm việc cùng nhau, chúng em đã trau dồi thêm những kiến thức về thuật toán Euclid, định lý Fermat, hàm số Euler, thuật toán Miller-Rabin, định lý phần dư Trung Hoa... Bên cạnh đó, các thành viên cũng có cơ hội rèn luyện các kỹ năng mềm như kỹ năng làm việc nhóm, kỹ năng giao tiếp, kỹ năng quản lý thời gian hiệu quả, sử dụng thành thạo các công cụ hỗ trợ như word, powerpoint, Visual Studio Code... Đây chắc chắn sẽ là những bài học quý báu, hành trang bổ ích để chúng em có thể tự tin vững bước trên con đường lập nghiệp và thành công trong cuộc sống sau này.

Tuy nhiên, do kiến thức còn hạn hẹp nên chương trình còn đơn giản và chưa thật sự tối ưu.

Trong tương lai, nhóm sẽ tiếp tục tích lũy kinh nghiệm và kiến thức để phát triển hoàn thiện chương trình này hơn nữa và có thể sẽ áp dụng nó trong thực tế cuộc sống.

TÀI LIỆU THAM KHẢO

- [1] V. Phạm, "Bizflycloud," 06 05 2022. [Online]. Available: <https://bizflycloud.vn/tin-tuc/rsa-la-gi-20220506153312797.htm>.
- [2] "Wikipedia," [Online]. Available: [https://vi.wikipedia.org/wiki/RSA_\(m%C3%A3_h%C3%B3a\)](https://vi.wikipedia.org/wiki/RSA_(m%C3%A3_h%C3%B3a)).
- [3] N. V. Q. E, "Viblo," 23 7 2021. [Online]. Available: <https://viblo.asia/p/co-che-hoat-dong-va-ung-dung-cua-rsa-eW65G4XOKDO>.
- [4] T. A. Vu, "Viblo," 6 8 2017. [Online]. Available: <https://viblo.asia/p/java-ma-hoa-va-giai-ma-voi-thuat-toan-rsa-bJzKmW3Xl9N>.
- [5] RSA Security's official guide to cryptography by Steve Burnett
- [6] Cryptanalysis of RSA and Its Variants by M. Jason Hinek
- [7] RSA and Public-Key Cryptography by Richard A. Mollin