Georgia State University

14th Annual Georgia State Undergraduate Research Conference

Cryptography: From Caesar Cipher to RSA Cipher

Name: Maisa Basher

To Quyen Pham

Advisor: Professor Muiny Somaya

Date: March 20, 20

Contents

Abstract

Cryptography is an essential technique used in securing information over networks. Although it had a tremendous impact on the outcome of World War II, cryptography is not limited to government and military intelligence. With the revolution of electronic communications nowadays, an extensive use of cryptography to maintain confidentiality is required. To achieve that purpose, rigorous math is needed. In this paper, we focus on the mathematical foundations of Caesar Cipher and RSA Cipher, and demonstrate implementation of encrypted/decrypted messages using both ciphers.

I. Introduction

Cryptography is the science of using advanced mathematical algorithms to transform data into a certain set of symbols which is known as ciphertext so that only the intended recipient can process and interpret the message. Symmetric encryption is the oldest and simplest kind of encryption which involves only one secret key, which is known to both the sender and the receiver. The biggest concern of the symmetric encryption is that all the involved parties have to exchange their secret key so that the beneficiary can use it to decrypt the message; nevertheless, there is always a risk of their key being stolen when transferring it over the internet. Asymmetric encryption, also known as public key cryptography, was invented to replace the adversary of the former encryption. This encryption involves two distinct keys, a public key and private key. The big breakthrough of the latter method is that the knowledge of public key provides virtually no clue how to find the private key.

Modern cryptography has four main objectives:

- Confidentiality is the guarantee that information cannot be disclosed to unauthorized individuals
- Integrity is the guarantee that the information has not been altered when transmitted over networks
- Non-repudiation is the guarantee that the sender of the information cannot deny at a later stage his/her intention of the message
- Authentication is the guarantee that both sender and receiver can confirm each other's identity.

II. The Caesar Cipher

A. Brief Introduction of Caesar Cipher and Symmetry Cryptography

The Caesar Cipher is one of the earliest and simplest ciphers used for sending secret messages. The Caesar Cypher uses a symmetric method meaning that both the sender and the receiver are aware of the encryption and decryption procedures. This means that the

knowledge of how to encrypt simultaneously gives answered keys for decrypting messages. Notably, in the Caesar Cipher technique, each letter in plain text is shifted a certain number of places down the alphabet.

B. The Process of the Caesar Cipher.

1.1. The Caesar Cipher algorithm

For instance, each letter of the alphabet is encoded by its positions relatively to the others, so that A= 01, B= 02, …, Z= 26, and the Caesar Cipher encrypts messages by changing each letter of the alphabet to the one three places farther along. The Caesar Cipher operation is as follows:

Let M represents the numerical version of the plaintext and C represents the numeric version of the ciphertext, and the encryption/ decryption functions are:

Encryption function: $C = (M + 3)\ mod\ 26$

Decryption function : $M = (C - 3)\ mod\ 26$

1.2.A Worked example:

Suppose Bod sets up a Caesar Cipher, he encodes the letter of the alphabet such as A= 01, B= 02,…, Z= 26. Alice wants to send Bob the message 'HI', and she divides her message into blocks, one for letter 'H' and another for letter 'I'. Since 08 and 09 represent for 'HI' respectively, the encryption procedures are as following

- For letter H, $C = (08 + 3)\ mod\ 26 = 11$
- For letter I, $C = (09 + 3)\ mod\ 26 = 12$

Here 11 and 12 represent 'KL' in the English alphabet, and 'KL' is actually the message sent over the internet. When Bob receives the message, he uses following algorithm to get the original message:

- For number 11, $M = (11 - 3)\ mod\ 26 = 8$
- For number 12, $M = (12 - 3)\ mod\ 26 = 9$

Hence, Bobs retrieves message 8 and 9 which correspond to the letter 'HI'.

C. Demonstration of the Caesar Cipher

The code below is generated using Java language and NetBeans software.

Figure 1: The Caesar Cipher algorithm is implemented with Java language

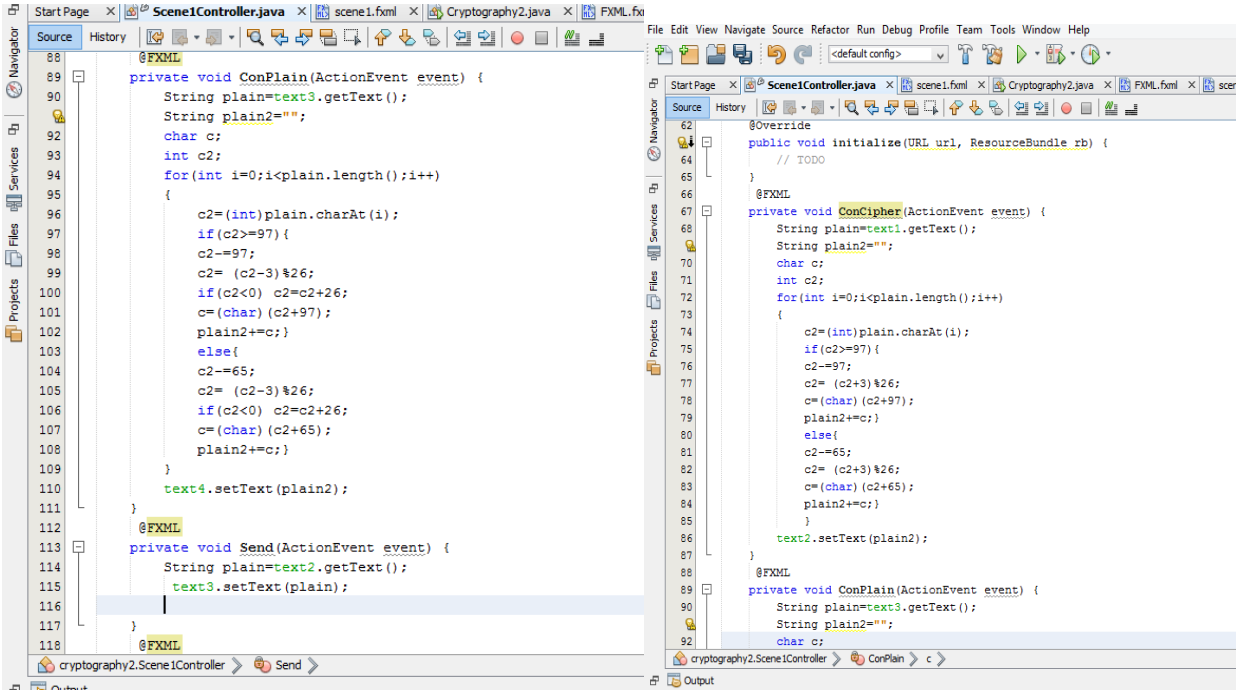
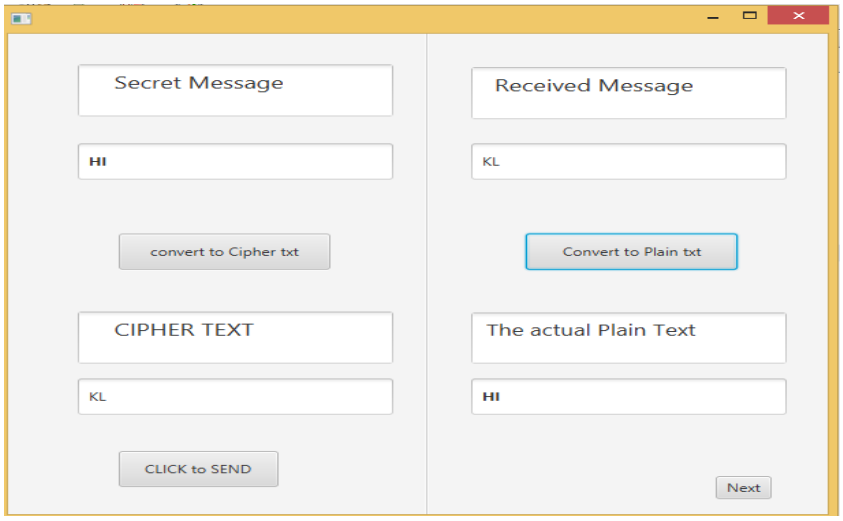


Figure 2: Caesar Cipher Encryption and Decryption Application of Graphical User Interface

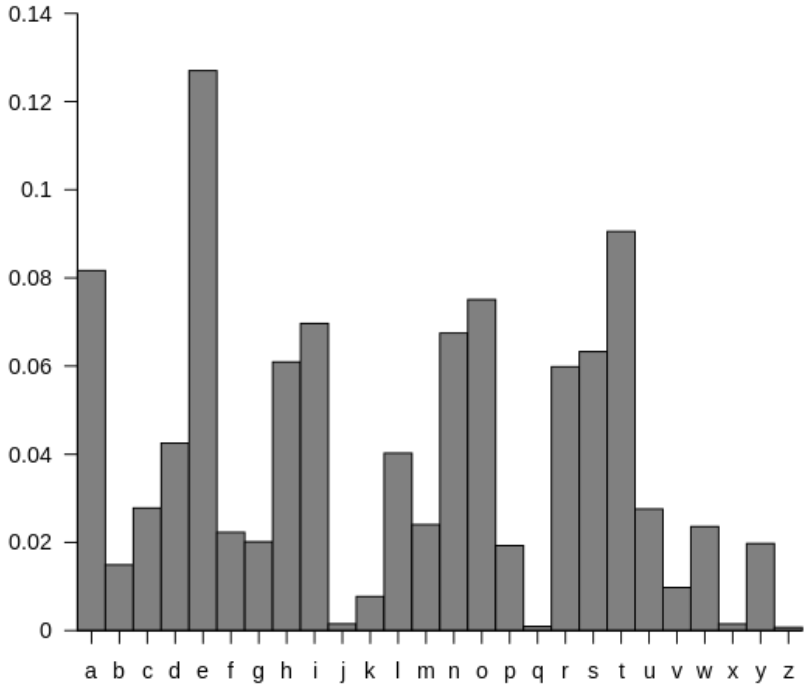


D. Caesar Cipher Security

One of the main problems of the Caesar Cipher is its inheritance symmetric properties. Firstly, the secret key must be sent to the recipient in order to decrypt the message; however, there is no assurance that the secret key is hacked when transmitted over the internet. Besides, in the Caesar Cipher the knowledge of how to encrypt can automatically give the method of how messages are decrypted since the formula of encryption and

decryption are symmetric. For instance, in the above encryption algorithm to encrypt a message, each plaintext is shifted to the one three places farther along the alphabet and modulo 26. Conversely, an attacker can easily recover the original message by subtracting the ciphertext to 3 and modulo 26. Also, the possibility of getting a right shift is very limited which ranges from 1 to 25; therefore, with certain attempts making a piece of readable text, an attacker can easily find the decryption key. That makes the Caesar Cipher the most breakable encryption method ever. Furthermore, the alphabet letters are mainly used in encryption procedures, each letter has a certain frequency when used in English text. Hence, a person with proper knowledge of word distribution can easily break the Caesar Cipher codes. The table below shows frequencies of each letter used in the English alphabet:

Table 1: Letter distribution in the English alphabet



Source:("Letter Frequency")

As shown in table 1, vowels are used more frequently than consonants in the text, so an attacker can make reasonable guesses based on the ciphertext with the knowledge of letter distribution.

III. The RSA Cipher

A. Brief Introduction of RSA and Symmetry Cryptography

The RSA Cipher was invented in 1976-1977. It is one of the first and widely used techniques for public key encryption. The algorithm was named after its inventor Ron Rivest, Adi Shamir and Leonard Adleman who were mathematicians and computer scientists working at M.I.T. The mathematical idea of RSA encryption is based on a pair of numbers, known as the public key, which is published, and people use those two numbers to encrypt their message. However, only the beneficiary, who knows the secret key, can decrypt the message.

B. The Process of RSA Cipher

1. Primary definitions:

1.1.Relatively Prime Numbers

Two integers a and b are relatively prime if and only if their greatest common devisor is i.e. $gcd(a, b) = 1$

1.2.Modular Arithmetic

Let a, b, c, d and n be integers with n > 1, and suppose

$$a \equiv c \ (mod\ n) \quad and \quad b \equiv d \ (mod\ n)$$

Then, the modular addition, subtraction and multiplication is consistent with its original modular arithmetic:

$$(a + b) \equiv (c + d) \ (mod\ n)$$

$$(a - b) \equiv (c - d) \ (mod\ n)$$

$(ab) \equiv (cd) \ (mod\ n)$

$a^m \equiv c^m \ (mod\ n)$ for all integers m

1.3.Extended Euclidean Algorithm

For all coprime integers a and n, there exits an integer s such that $(as \equiv 1 \ (mod\ n)$. And, the integer s is called the inverse of a modulo n

1.4.Fermat's Little Theorem

If p is any prime number and a is any integer such that p ł a, then $a^{p-1} \equiv 1 \ (mod\ p)$

1.5.Euclid's lemma

For all integer a, b and c, if the greatest common divisor of a and c is 1 and $a \mid bc$, then $a \mid b$.

2. RSA Cipher Operation

RSA cipher operation involves three steps: key generation, encryption and decryption

2.1. Key generation

There two distinguish keys used in the process: public key (e, n) and private key (p, q). The public key is used for encrypting messages and is known to everyone. However, the

private key is known to only the messages' recipient and is used for decrypting messages. Although those two keys are mathematically related, knowledge of the public key provides virtually no clue on how to obtain the private key. The keys for the RSA algorithm are generated as follows:

- Step 1: Choose two prime numbers p and p and each number is at least 40 digits long
- Step 2: Compute $n = pxq$, n is usually expressed in bit and will be used as the modulus for both encryption and decryption functions
- Step 3: Compute Euler's totient function $(n) = (p-1)(q-1)$, and (n) indicates exactly how many numbers are coprime with number n
- Step 4: Choose an integer e such that $gcd(e,(N)) = 1$, and $1 < e < (n)$ i.e. e and □(n) are coprime integers. This number e will be used as encryption exponent when a message is encrypted
- Step 5: Determine number d such that $ed \equiv 1\ mod\ ((n))$, and 1 < d < □(n) i.e. number d is the multiplicative inverse of e (mod (□(n)). That number d is computed using the Extended Euclidean Algorithm and is used as decryption exponent to recover the original message.

2.2.Encryption function

Suppose Bob sets up an RSA cipher, he publishes his public key (n, e) and keeps his private key (p, q) as a secret. The numerical version of the plaintext for a letter is denoted by M and the numerical version of the ciphertext is denoted by C. Alice wishes to send Bod a message, so she computes the cipher text C corresponding to

$$C = M^e\ mod\ n \qquad (1)$$

2.3.Decryption function

Bob retrieves the original messages from Alice by using his private key (p, q) to compute the decryption key d. Then he recovers the plaint text M corresponding to

$$M = C^d\ mod\ n \qquad (2)$$

2.4.A worked example

Bod uses following steps for setting up the cipher:

- Step 1: Choose two distinct numbers, such as $p = 2$ and $q = 7$
- Step 2: Compute $n = pxq$, giving $n = 2x7 = 14$
- Step 3: Compute Euler's totient function $(n) = (p-1)(q-1)$, giving $(n) = (2-1)(7-1) = 6$
- Step 4: Choose encryption exponent e such that $1 < e < 6$, and the number e also need to be coprime with 6. He takes $e = 5$ as it satisfies all requirements.

- Step 5: Compute decryption exponent d, the inverse of e modulo n such that $1 < d < 6$, and $5d \equiv 1 \; mod \; 6$. Hence, $d = 11$

Then, Bod will pubic the public key $(e = 5, n = 14)$, but keeps the private key $(p = 2, q = 7)$ as a secret. He also encodes each letter of the alphabet by its positions relatively to the others, so that A= 01, B= 02,…, Z= 26. For the sake of simplicity, parameters used are artificially small; however, in reality those parameters are enormous, and it is extremely hard to factor them. Say Alice wishes to send Bob the message 'HI'. The encryption process is as follows:

- For letter 'H', $C = 8^5 \; mod \; 14 = 32768 \; mod \; 14 = 8$
- For letter 'I', $C = 9^5 \; mod \; 14 = 59049 \; mod \; 14 = 11$

Hence, 10 and 1 are messages Bob will receive. And, he applies the decryption function to retrieve the original messages as following:

- For C=8, $M = 8^{11} \; mod \; 14 = (8^8 x 8^2 x 8^1) mod \; 14 = [(8^8 \; mod \; 14)(8^2 \; mod \; 14)(8^1 \; mod \; 14)] \; (mod \; 14) = (8x8x8)(mod \; 14) = 512(mod \; 14) = 8 \; (mod \; 14)$
- For C=11, $M = 11^{11} \; mod \; 14 = (11^8 x 11^2 x 11^1) \; mod \; 14 = [(11^8 \; mod \; 14)(11^2 \; mod \; 14)(11^1 \; mod \; 14)] \; (mod \; 14) = (9x9x11)(mod \; 14) = 891 \; (mod \; 14) = 9 \; (mod \; 14)$

Hence, Bobs retrieves message 8 and 9 which correspond to the letter 'HI'.

2.5.Proofs of Correctness

For the RSA cryptography method, from encryption function (1) and decryption function (2), and by substitution:

$M = C^d \; mod \; n = \; (M^e \; mod \; n)^d \; mod \; pq$

And, by modular arithmetic exponential law:

M≡ $M^{ed} mod \; pq$, and we will prove this function is true.

- Case 1: Using Fermat's Little Theorem, when M is relatively prime to pq

Recall that, d was chosen to be a positive inverse for e modulo (p-1)(q-1)

i.e. $ed \equiv 1\big(mod \; (p-1)(q-1)\big)$

Equivalently,

$ed = 1 + h(p-1)(q-1)$ for some positive integer h

Hence,

$$M^{ed} = \; M^{1+h(p-1)(q-1)} = M(M^{p-1})^{h(q-1)} = M(M^{q-1})^{h(p-1)}$$

If p ∤ M and q ∤ M, then by Fermat's little theorem (1.4),

$$M^{ed} = M(M^{p-1})^{h(q-1)} \equiv M \; (mod \; p) \; \equiv M \; (mod \; q)$$

- Case 2: Using Euclid's lemma, when M is not relatively prime to pq

In this case, then either p | M or q | M

$$\begin{cases} M^{ed} \equiv 0 \equiv M \pmod{p} \\ M^{ed} \equiv 0 \equiv M \pmod{q} \end{cases}$$ By modular equivalences, $$\begin{cases} p \mid (M^{ed} - M) \\ q \mid (M^{ed} - M) \end{cases}$$

By definition of divisibility,

$$M^{ed} - M = pt \text{ ,for some integer } p \qquad (3)$$

By substitution,

$$q \mid pt \quad (4)$$

As p and q are two prime numbers and (4), then by Euclid's lemma theorem (1.5)

$$q \mid t$$

Thus, $qu = t$, for some integer u

Substitute to (3), then $M^{ed} - M = pt = p(qu) = (pq)u$

Therefore, $pq \mid (M^{ed} - M)$

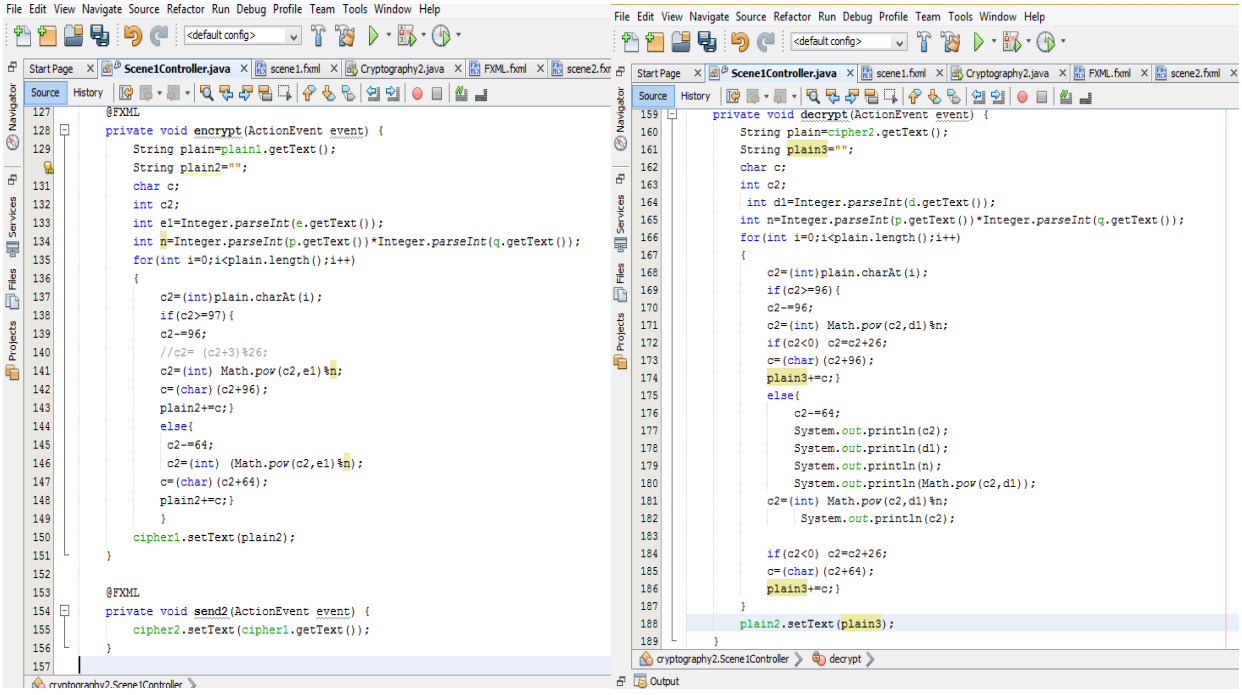
Hence, $(M^{ed} - M) \equiv 0 \pmod{pq}$

i.e. $M \equiv M^{ed} \bmod pq$

C. Demonstration of the RSA

The code below is generated using Java language and NetBeans software.

Figure 3: The RSA Cipher algorithm is implemented with Java



```
@FXML
private void encrypt(ActionEvent event) {
    String plain=plain1.getText();
    String plain2="";
    char c;
    int c2;
    int e1=Integer.parseInt(e.getText());
    int n=Integer.parseInt(p.getText())*Integer.parseInt(q.getText());
    for(int i=0;i<plain.length();i++)
    {
        c2=(int)plain.charAt(i);
        if(c2>=97){
        c2-=96;
        //c2= (c2+3)%26;
        c2=(int) Math.pow(c2,e1)%n;
        c=(char)(c2+96);
        plain2+=c;}
        else{
         c2-=64;
         c2=(int) (Math.pow(c2,e1)%n);
        c=(char)(c2+64);
        plain2+=c;}
         }
    cipher1.setText(plain2);
}

@FXML
private void send2(ActionEvent event) {
    cipher2.setText(cipher1.getText());
}
```

```
private void decrypt(ActionEvent event) {
    String plain=cipher2.getText();
    String plain3="";
    char c;
    int c2;
     int d1=Integer.parseInt(d.getText());
    int n=Integer.parseInt(p.getText())*Integer.parseInt(q.getText());
    for(int i=0;i<plain.length();i++)
    {
        c2=(int)plain.charAt(i);
        if(c2>=96){
        c2-=96;
        c2=(int) Math.pow(c2,d1)%n;
        if(c2<0) c2=c2+26;
        c=(char)(c2+96);
        plain3+=c;}
        else{
            c2-=64;
            System.out.println(c2);
            System.out.println(d1);
            System.out.println(n);
            System.out.println(Math.pow(c2,d1));
        c2=(int) Math.pow(c2,d1)%n;
             System.out.println(c2);

        if(c2<0) c2=c2+26;
        c=(char)(c2+64);
        plain3+=c;}
    }
    plain2.setText(plain3);
}
```

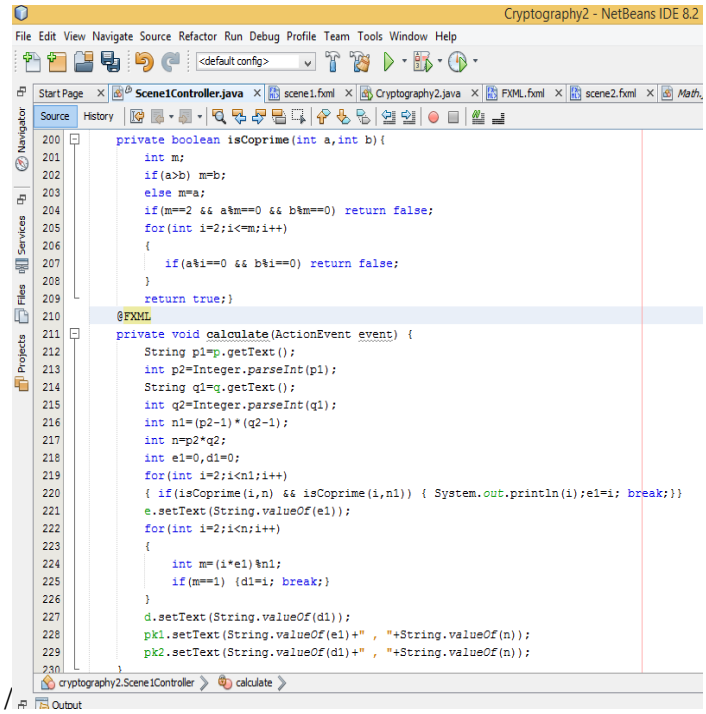
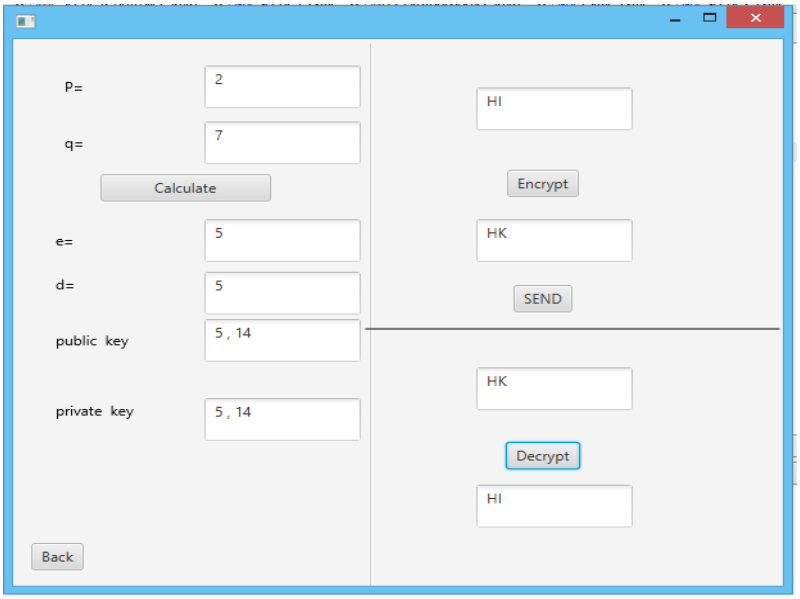


Figure 4: The RSA Encryption and Decryption Application of Graphical User Interface



C. RSA Security

RSA security relies on two mathematical problems: the RSA problem and the difficulty of factoring large composite numbers. First, as the RSA encryption algorithm states $C = M^e \ mod \ n$, then so as to retrieve the original message M from the ciphertext C an attacker can take the e^{th} roots of an arbitrary number modulo a composite n, where (n, e) is the public key. However, there

is still no efficient method for solving the RSA problem as given large RSA key sizes which in fact exceed 1024 bits. Subsequently, the alternative method known to solve the RSA problem is by factoring the modulus n into initial two prime number p and q. With the knowledge of those two numbers, an attacker can compute Euler's totient function $(n) = (p-1)(q-1)$ to find encryption exponent e, and define decryption exponent d from e. Then, he can decrypt messages using standard process $M = C^d \, mod \, n$. However, there is currently no fast way to identify the prime factor of very large numbers even with the best computers. The following table, was presented in 1978 by one of the authors of RSA, summarizes polynomial time of factoring a semiprimes n for various length of n:

Table 2: Time of factoring composite number n-digits length

Digits	Number of operations	Time
50	$1.4x10^{10}$	3.9 hours
75	$9.0x10^{12}$	104 days
100	$2.9x10^{15}$	7.4 years
200	$1.2x10^{23}$	$3.8x10^{9}$years
300	$1.5x10^{29}$	$4.9x10^{15}$years
500	$1.3x10^{29}$	$4.2x10^{25}$years

Source: (*Yevgeny.Pdf*)

In practice, for the sake of security and speed, the most common length recommended by the RSA authors is approximately 200 digits long (approximately 1024 to 2048 bit-length). As of 2020, RSA-250 decimal digits (829 bits), which is the largest RSA factored number, was factored by Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thomé, and Paul Zimmermann. Therefore, the suggested length 1204-bit keys are forecasted to be breakable, and some research scientists also believe that 4096-bit keys could even be factored in foreseeable future

IV. Conclusion

Cryptography plays a crucial role to help our daily activities such as online purchases, internet banking transactions, distanced conversations etc. function smoothly. The mathematical ideas of RSA Cipher deal with common definitions such as prime numbers, modular arithmetic and factorization; however, the combination of the whole fulfills all main four objectives of modern cryptology. The rigorous encryption and decryption algorithms guarantee confidentiality, integrity, non-repudiation as well as authentication as other unauthorized parties except the intended recipient cannot interpret the information.

Acknowledgements

We gratefully thank Professor Muiny Somaya for her continuous support as well as her guidance and encouragement from our very initial steps of doing this research paper. Also, we would like to give a special thanks to Professor Srilatha Bingi for giving us useful advice to improve our codes.

References

1. “Letter Frequency.” Wikipedia, 7 Mar. 2020. Wikipedia, https://en.wikipedia.org/w/index.php?title=Letter_frequency&oldid=944334192.
2. “RSA Numbers.” Wikipedia, 18 Mar. 2020. Wikipedia, https://en.wikipedia.org/w/index.php?title=RSA_numbers&oldid=946209912.
3. Yevgeny.Pdf. https://sites.math.washington.edu/~morrow/336_09/papers/Yevgeny.pdf. Accessed 20 Mar. 2020.
4. Cryptography Defined/Brief History. https://www.laits.utexas.edu/~anorman/BUS.FOR/course.mat/SSim/history.html. Accessed 31 Jan. 2020.
5. Susanna S. Epp. Discrete Mathematics with Applications 4th Edition. Cengage Learning; 004 edition (August 4, 2010)