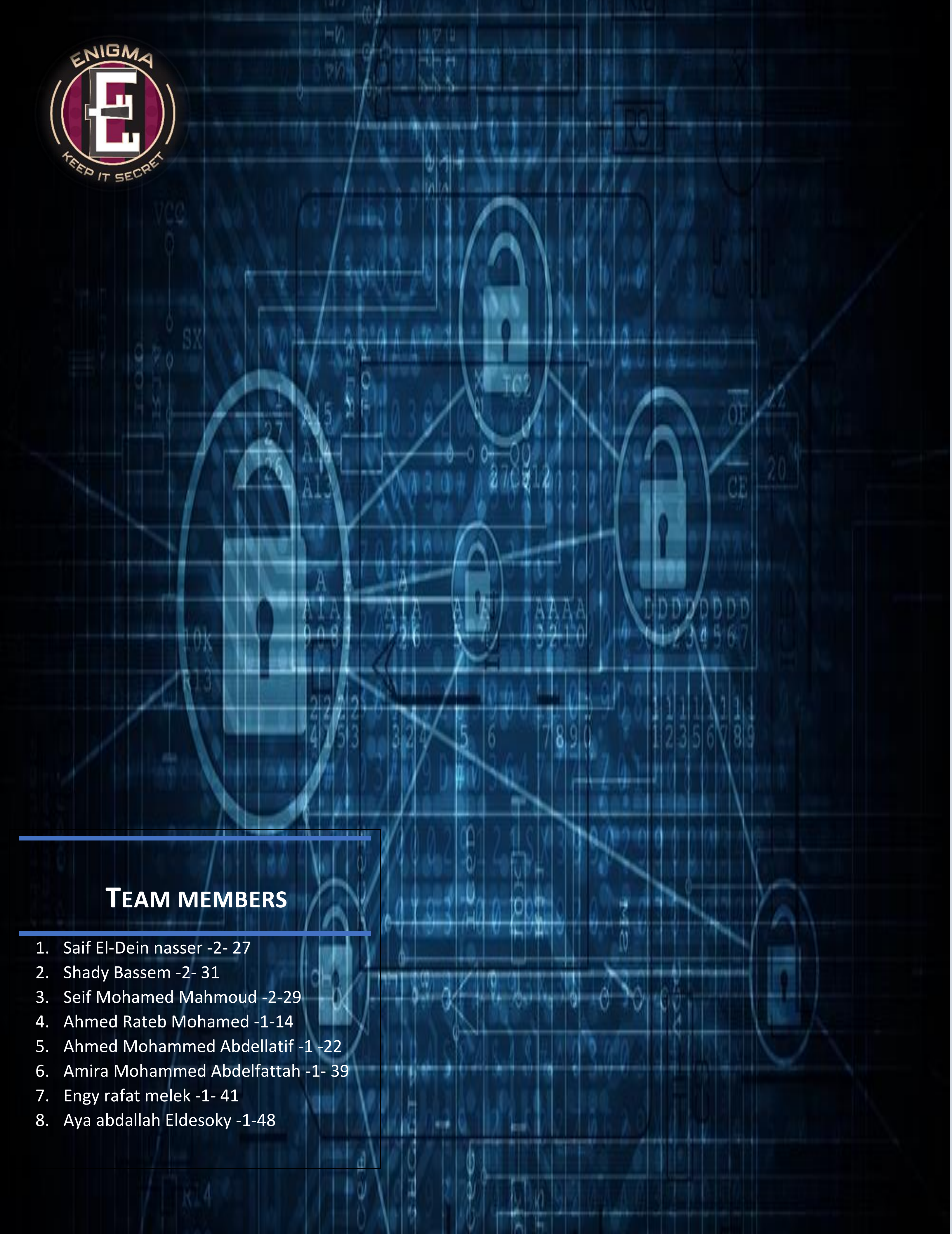
ENIGMA

KEEP IT SECRET



Team members

1. Saif El-Dein nasser -2- 27
2. Shady Bassem -2- 31
3. Seif Mohamed Mahmoud -2-29
4. Ahmed Rateb Mohamed -1-14
5. Ahmed Mohammed Abdellatif -1 -22
6. Amira Mohammed Abdelfattah -1- 39
7. Engy rafat melek -1- 41
8. Aya abdallah Eldesoky -1-48

Table of Contents

Abstract

Encryption is the process of encoding information or data to prevent unauthorized access. These days we need to secure the information that is stored on our computers or is transmitted via the internet against attacks. There are different types of cryptographic methods that can be used. The selecting cryptographic method depends on the application demands such as the response time, bandwidth, confidentiality, and integrity. However, each cryptographic algorithm has its own weak and strong points.

Problem Definition

Securing information is a challenge in today's internet age, with online transactions occurring almost every second and terabytes of data being generated every day on the internet. Cryptography is an essential aspect of information security in today's world. The virtual world is a more secure environment. The method of making information unintelligible to an unauthorized individual is known as cryptography. Performance and implementation costs are, however, critical considerations for all practical applications. Since it is common practice to embed encryption algorithms in other applications such as eCommerce, banking, and online transaction processing applications. In this paper, the four of the popular secret key encryption algorithms, i.e., DES, 3DES, AES (Rijndael), and the Blowfish have been implemented, and their performance is compared by encrypting input files of varying contents and sizes. Based on the experiments, it has been concluded that the Blowfish is the best performing algorithm among the algorithms chosen for implementation.

Encryption Definition

Encryption is a method of disguising data so that it cannot be read by anyone who does not know the key. The key is used to lock and unlock data. To encrypt data, Mathematical functions are to be performed on the data and the result of these functions would produce an output that makes the data appear as meaningless words or symbols to anyone who doesn't know how to reverse the operations. Encryption can be used to encrypt files that the owner feels are too sensitive for anyone else to read. And now, in the age of the Internet, encryption is used to encrypt data, like a credit card number, and then send it across the net. This way no one can read intercept and read the data while it is traveling through the web. The recipient of the data does have to know how to decrypt the information or else the data will look like garbage to the recipient too. Internet, encryption is used to encrypt data, like a credit card number, and then send it across the net. This way no one can read intercept and read the data while it is traveling through the web. The recipient of the data does have to know how to decrypt the information or else the data will look like garbage to the recipient too.[1]

H*istory of encryption*

Cryptography has been used by many people and governments, it began thousands of years ago, and is known as classic cryptography, it depended on pen and paper. Until the beginning of the 20th, when machines started to develop, which helped in providing more sophisticated and efficient ways of encryption.

The Classical Methods:

Cryptography probably started in Egypt around 1900 BC, when a scribe used unexpected hieroglyphic characters instead of the usual ones, carved into the wall of a tomb, although its purpose wasn't exactly to hide security it's considered a way of changing the original text.[2]

The ancient Greeks are also said to have known of ciphers, a secret or disguised way of writing, i.e., the scytale transposition cipher which was used by the Spartan military. Also in 60 BC, Julius Caesar invents a substitution cipher that shifts characters by three places where A becomes D, B becomes E, etc. A simple encoding method at that time can be easily decoded by using letter frequencies in the alphabet .[2]

David Kahn notes in *The Codebreakers* that modern cryptology originated among the Arabs, the first people to systematically document cryptanalytic methods. Al-Khalil (717–786) wrote (The *Book of Cryptographic Messages)*, which contains the first use of permutations and combinations to list all possible Arabic words with and without vowels. [2]

In the 16th century, Vigenère designed a cipher that was supposedly the first cipher that used an encryption key, it depended greatly on the Caesar cipher as it contains many substitutions with varying values that depend on the encryption key, unlike the Caesarcipherr which depends on the secrecy of the text. [2]

In 1854, Charles Wheatstone invents the Playfair Cipher, which encrypts pairs of letters instead of single letters as in the simple substitution cipher.[2]

Although cryptography has a long and complex history, all these methods are still considered classical ones and not efficient enough unlike what was discovered later in the field of cryptography and encryption.[2]

Enigma and world wars:

At the start of the 19th century, the American inventor Edward Hebron invented the electro-mechanical machine in which the key is embedded in a rotating disc. It's the first example of a rotor machine. It encodes a substitution table that is changed every time a new character is typed. This was also broken by using letter frequency. [2]

Later in 1917, the German engineer, Arthur Scherbius, invented the Enigma machine, which uses multiple rotors from the Hebronern machine. They rotate at different rated es output appropriate letters ciphertext . In this case, the key was the initial setting of the rotors. The German military then started using Enigma in sending coded transmissions. However, in 1932, polish cryptographer Marian Rejewski discovered how Enigma works and sent this information to the British intelligence leading the mathematician Alan Turing to figure out how to crack the key every day.[2]

Up to the Second World War, most of the work on cryptography was for military purposes, usually used to hide secret military information. However, cryptography attracted commercial attention post-war, with businesses trying to secure their data from competitors. [3]

Modern Day Encryption:

In the early 1970s: IBM formed a 'crypto group,' which designed a block cipher which was called "Lucifer" to protect its customers' data. In 1973, the US adopted it as a national standard - the Data Encryption Standard, or DES. It remained in use until it cracked in 1997. In 1973, the Nations Bureau of Standards (now called NIST) in the US put out a request for proposals for a block cipher that would become a national standard. They had realized that they were buying a lot of commercial products without any good crypto support. Lucifer was eventually accepted and was called DES or the Data Encryption Standard. In 1997, and the following years, DES was broken by an exhaustive search attack. The main problem with DES was the small size of the encryption key. As computing power increased it became easy to brute force all different combinations of the key to obtaining a possible plain text message. In modern cryptography, the security of encryption depends not on the encryption method (or algorithm) but the secrecy of the keys used for encryption and decryption.[3]

Today's Encryption:

Today, encryption is part of our everyday life, it happens every second without people taking notice of it.

Simply put, it's no longer only humans that communicate. Every time a computer connects to the internet, you visit a webpage (HTTPS), use a messaging or e-mail application on your phone, computers, devices, and software are communicating with each other via the internet, Bluetooth, and Wi-Fi.

HOW ENCRYPTION WORKS?

As we have mentioned Encryption is the process of encoding information or data in order to prevent unauthorized access. There are different types of cryptographic methods that can be used. Each one of them serving different topology and all provide secure transmitted data through network links and ensure authentication and confidentiality. All these end-to-end encryption and decryption algorithms must be applied in the physical layer and security layer of the computer application. At the same time specific IP configurations are needed to be considered as well as the protocol that will be used to transmit the traffics. The diagram below showing us the cipher security classes which are subdivided into 2 models: classical and modern classes.

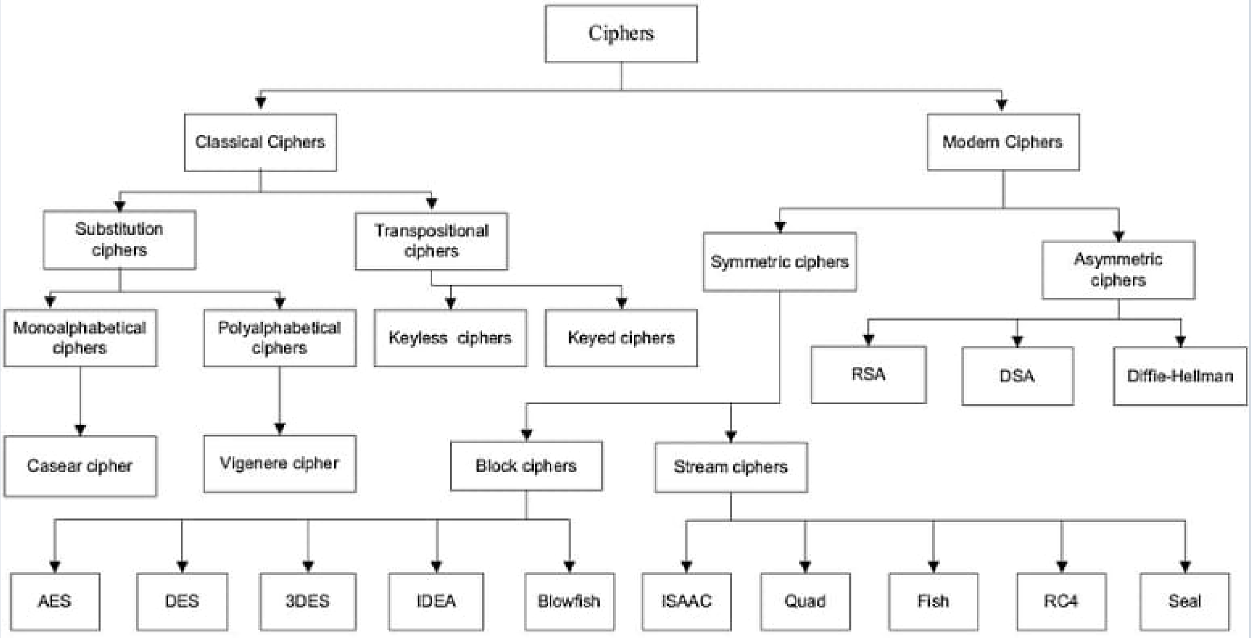


Figure 1:models of cipher security classes

Here are some essential encryption terms you should know:

❖ Algorithm

Also known as a cipher, algorithms are the rules or instructions for the encryption process. The key length, functionality, and features of the encryption system in use determine the effectiveness of the encryption. [3]

❖ Decryption

Decryption is the process of converting unreadable ciphertext to readable information.

❖ Key

An encryption key is a randomized string of bits used to encrypt and decrypt data. Each key is unique, and longer keys are harder to break. Typical key lengths are 128 and 256 bits for private keys and 2048 for public keys. [3]

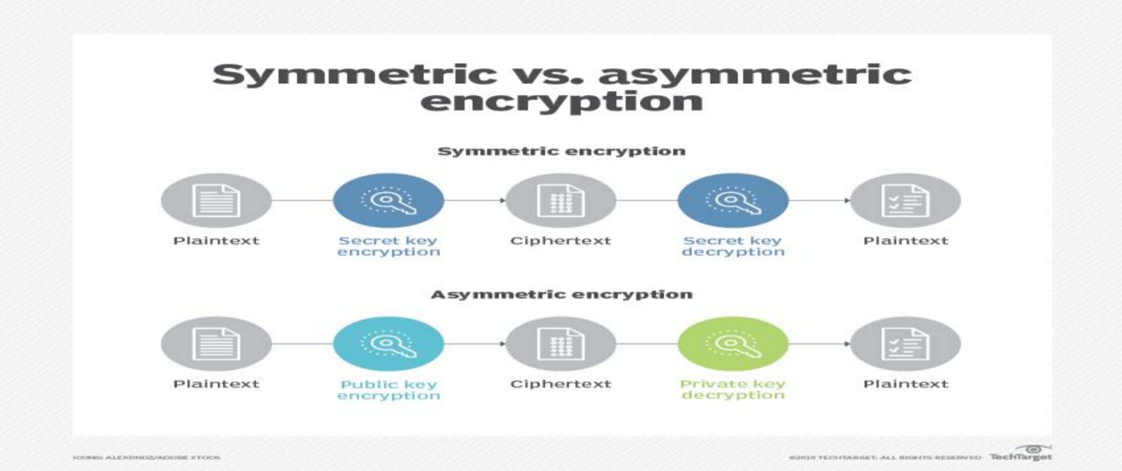
There are two kinds of cryptographic key systems, symmetric, and asymmetric.

1. Symmetric Key Systems

In a symmetric key system, everyone accessing the data has the same key. Keys that encrypt and decrypt messages must also remain secret to ensure privacy. While it’s possible for this to work, securely distributing the keys to ensure proper controls are in place makes symmetric encryption impractical for widespread commercial use. [3]

2. Asymmetric Key Systems

An asymmetric key system, also known as a public/private key system, uses two keys. One key remains secret—the private key—while the other key is made widely available to anyone who needs it. This key is called the public key. The private and public keys are mathematically tied together, so thecorrespondingprivate key can only decrypt that informationencrypted using the public key. [3]



ICONS: ALEXDNDZ/ADOBE STOCK

©2019 TECHTARGET. ALL RIGHTS RESERVED

Figure 2 . difference between symmetric and asymmetric encryption

COMMON ENCRYPTION ALGORITHMS

1) Triple DES:

Triple DES was designed to replace the original Data Encryption Standard (DES) algorithm, which hackers eventually learned to defeat with relative ease. At one time, Triple DES was the recommended standard and the most widely used symmetric algorithm in the industry. Triple DES uses three individual keys with 56 bits each. The total key length adds up to 168 bits, but experts would argue that 112-bits in key strength is more accurate. Despite slowly being phased out, Triple DES has, for the most part, been replaced by the Advanced Encryption Standard (AES). [14]

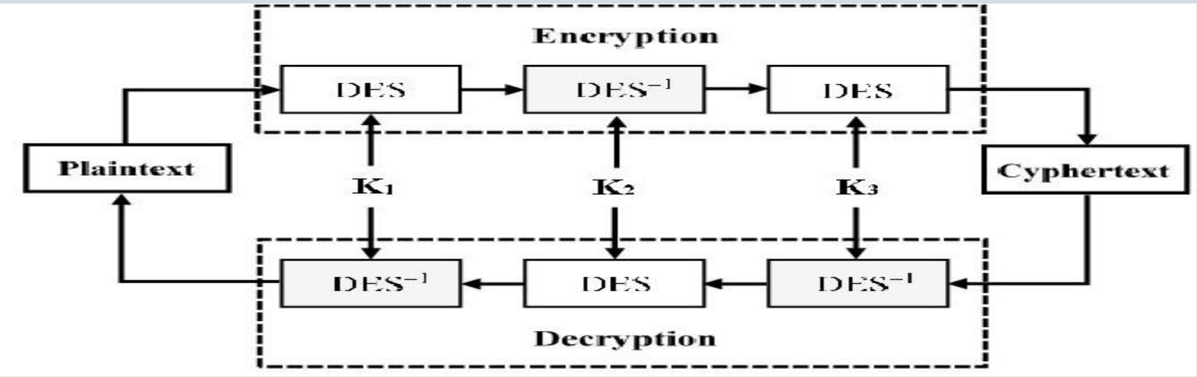
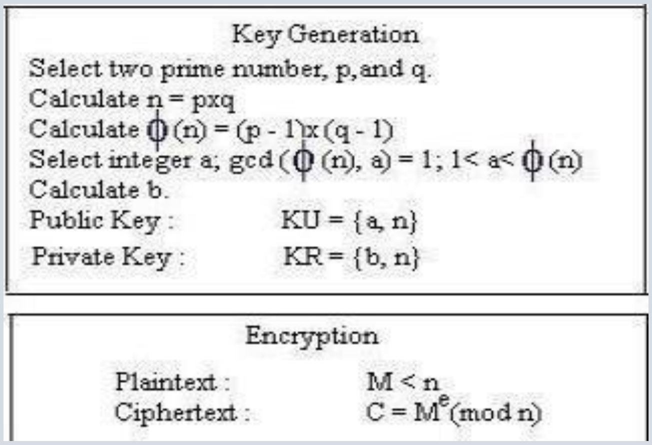


Figure 3: block diagram of 3DES algorithm [4]

2) RSA Security

RSA is a public-key encryption algorithm and the standard for encrypting data sent over the internet. It also happens to be one of the methods used in PGP and GPG programs. Unlike Triple DES, RSA is considered an asymmetric algorithm due to its use of a pair of keys. You've got your public key to encrypt the message and a private key to decrypt it. The result of RSA encryption is a huge batch of mumbo jumbo that takes attackers a lot of time and processing power to break. [14]



Key Generation

Select two prime number, p,and q.
Calculate $n = pxq$
Calculate $\phi(n) = (p - 1)x(q - 1)$
Select integer a; $\gcd(\phi(n), a) = 1; 1< a< \phi(n)$
Calculate b.
Public Key : $KU = \{a, n\}$
Private Key : $KR = \{b, n\}$

Encryption

Plaintext : $M < n$
Ciphertext : $C = M^e (\text{mod } n)$

Figure 4:RSA algorithm

3) AES

The Advanced Encryption Standard (AES) is the algorithm trusted as the standard by the U.S. Government and numerous organizations. Although it is highly efficient in 128-bit form, AES also uses keys of 192 and 256 bits for heavy-duty encryption purposes. AES is largely considered impervious to all attacks, except for brute force, which attempts to decipher messages using all possible combinations in the 128, 192, or 256-bit cipher. [14]

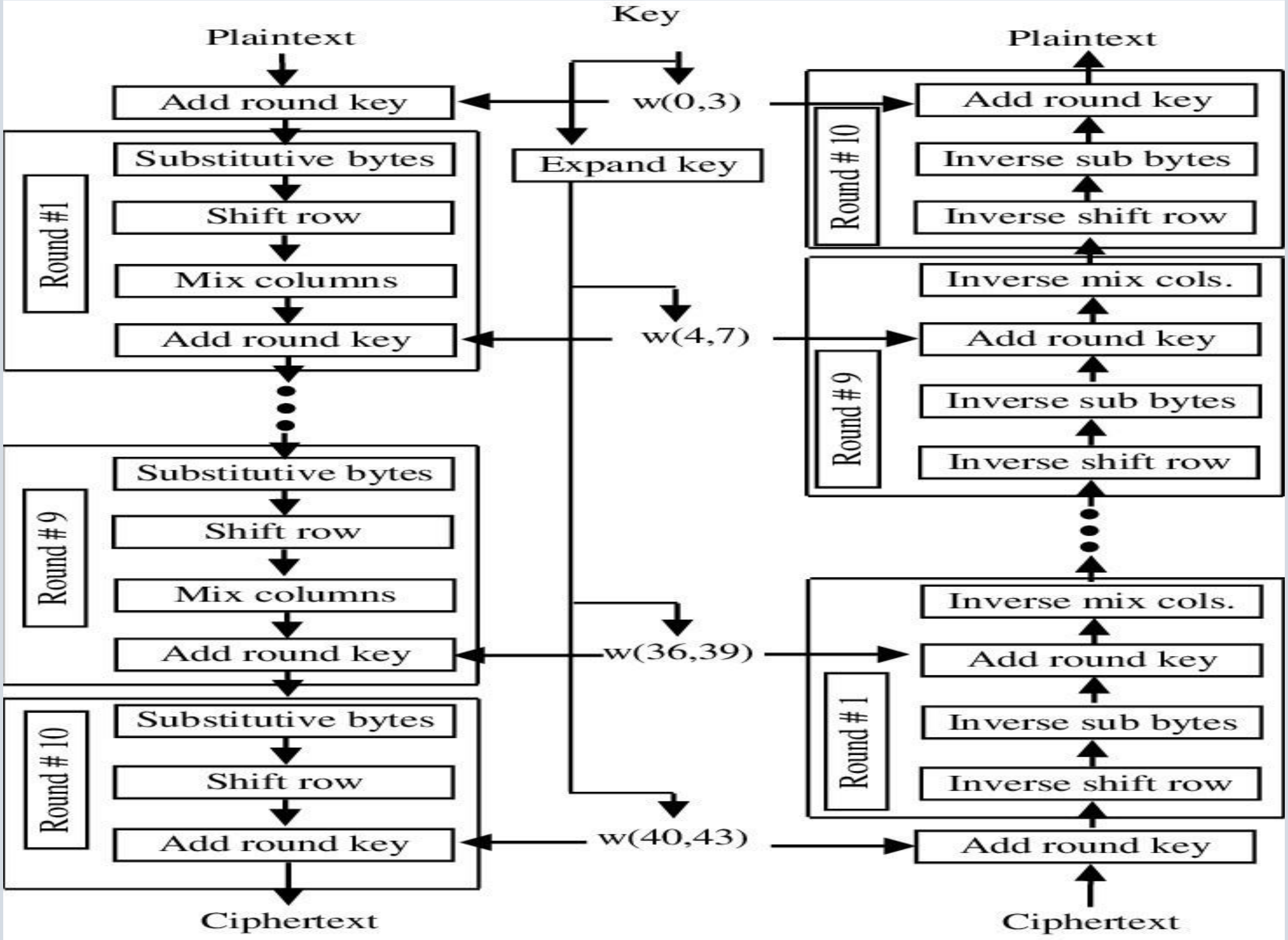


Figure5:Flow chart of AES algorithm [5]

4) Blowfish

Blowfish is yet another algorithm designed to replace DES. This symmetric cipher splits messages into blocks of 64 bits and encrypts them individually. Blowfish is known for its tremendous speed and overall effectiveness. Meanwhile, vendors have taken full advantage of its free availability in the public domain. You’ll find Blowfish in software categories ranging from e-commerce platforms for securing payments to password management tools, where it protects passwords. It’s one of the more flexible encryption methods available. [14]

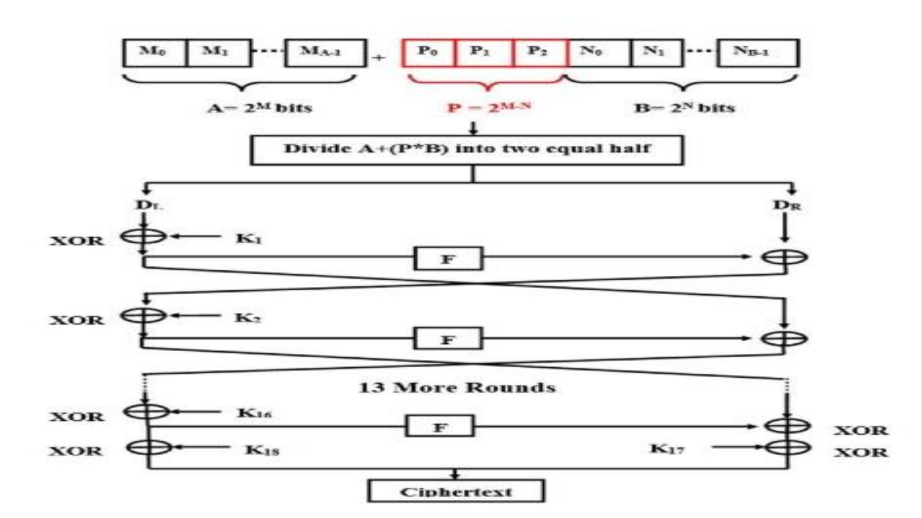


Figure 6 :Diagram of Blowfish encryption algorithm

5) TwoFish

Computer security expert Bruce Schneier is the mastermind behind Blowfish and its successor Twofish. Keys used in this algorithm may be up to 256 bits in length, and as a symmetric technique, you only need one key. Twofish is one of the fastest of its kind and ideal for use in hardware and software environments. Like Blowfish, Twofish is freely available to anyone who wants to use it. [14]

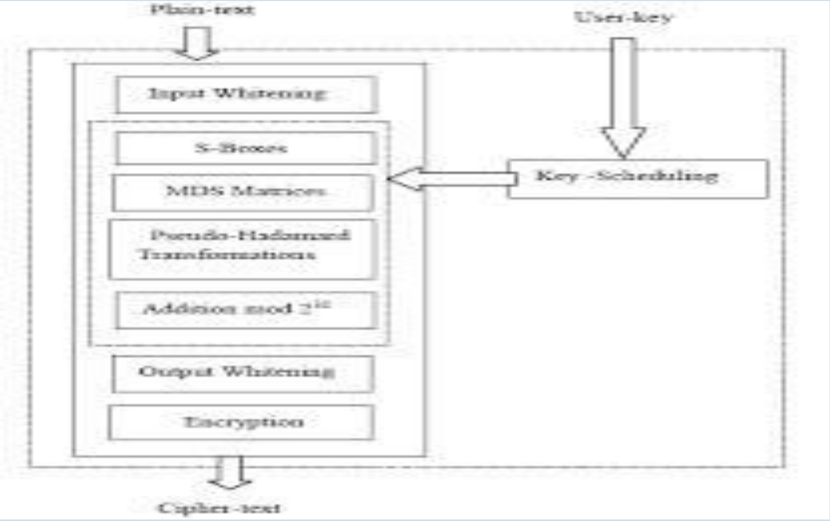


Figure 6 :Diagram for Twofish encryption algorithm

6) Hill cipher

The Hill cipher is a polygraphic substitution cipher built on concepts from Linear Algebra. The Hill cipher makes use of modulo arithmetic, matrix multiplication and matrix inverses; hence, it is a more mathematical cipher than others. The Hill cipher is also a block cipher, so, theoretically, it can work on arbitrary sized blocks. [6]

A Comparison between different Cryptographic Algorithms

Characteristics of algorithms [10]

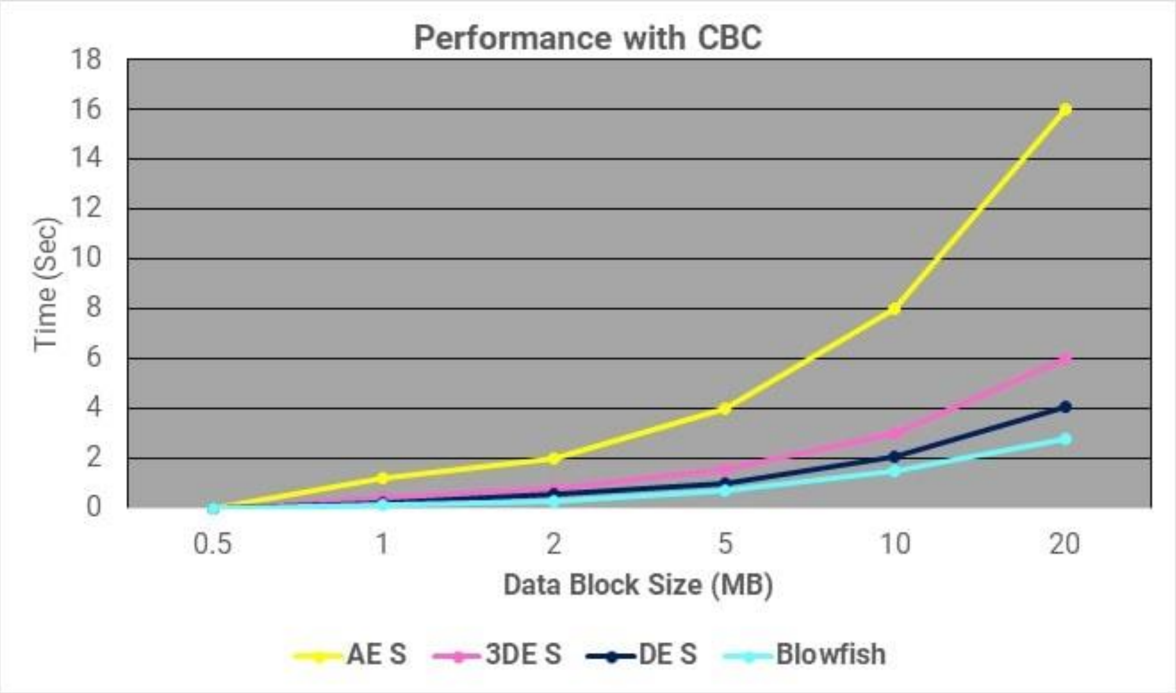


Figure 8.Comparison between common algorithms in performance with CBC

- Comparison in efficiency [10]

The techniques have been compared based on that how much:

- CPU processing speed for encrypting and decrypting data.
- Rate of key generation.
- Key size.
- Security consideration.
- Efficient on the hardware and software in case of implementation.
- The amount of memory required to hold the data in the encryption process.
- Number of users accommodated by the model
- Time required by the model to recover the data in case of key failure.
- Time available to the hacker to produce various types of attacks.
- complexity of algorithm technique. [10]

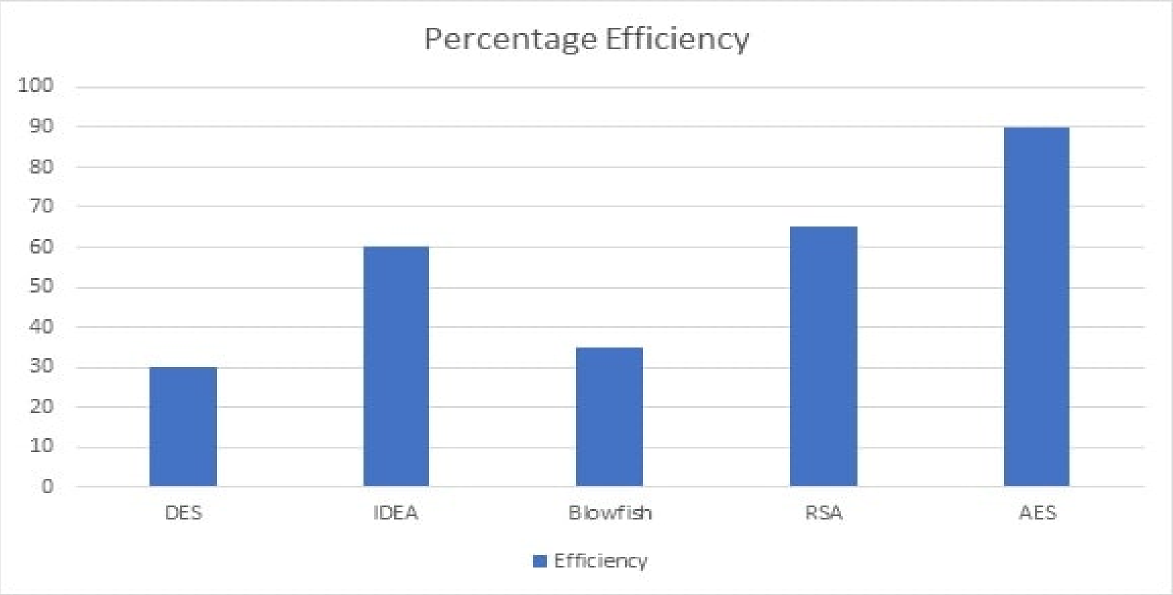


Figure 9.Comparison between common algorithms in efficiency

- Memory used

Algorithm	Memory Used in KB
DES	18.2
3DES	20.7
AES	14.7
Blowfish	9.38
RSA	31.5

Table 1.Memory used in common algorithms

- **Entropy**

is the randomness collected by an application for use in cryptography that requires random data. A lack of entropy can have a negative impact on performance and security. [10]

Algorithm	Average number of bits demanded tooptimally encode a byte of encrypted data
DES	27
3DES	40
Blowfish	128
RSA	44
AES	256

Table 2.Entropy in common algorithms

Methodology

Proposed System Architecture and Algorithm

The proposed system architecture was divided into two sections:

1. Transmitter Side Process

At transmitter side, the algorithm and flowchart were designed and implemented. Then the transmitter sends block of data or files by N matrices. The key matrix is generated depends on selective matrices and data is compressed into hex code for more confusion. Data compressed into hex codes were written to file name (.txt) at sender side. The compressed hex codes data written into file name at sender sides is transmitted to receiver side through the channel. [15]

2. Receiver side Process

At receiver side, algorithm and flowchart were designed and implemented. Then the receiver get data encoded to hex codes through the channel transmitted to it from sender side and receive the key matrix. Using this key matrix decode data encoded into hex codes to equivalent plaintext. Finally, the decoded information's were displayed at receiver side. [15]

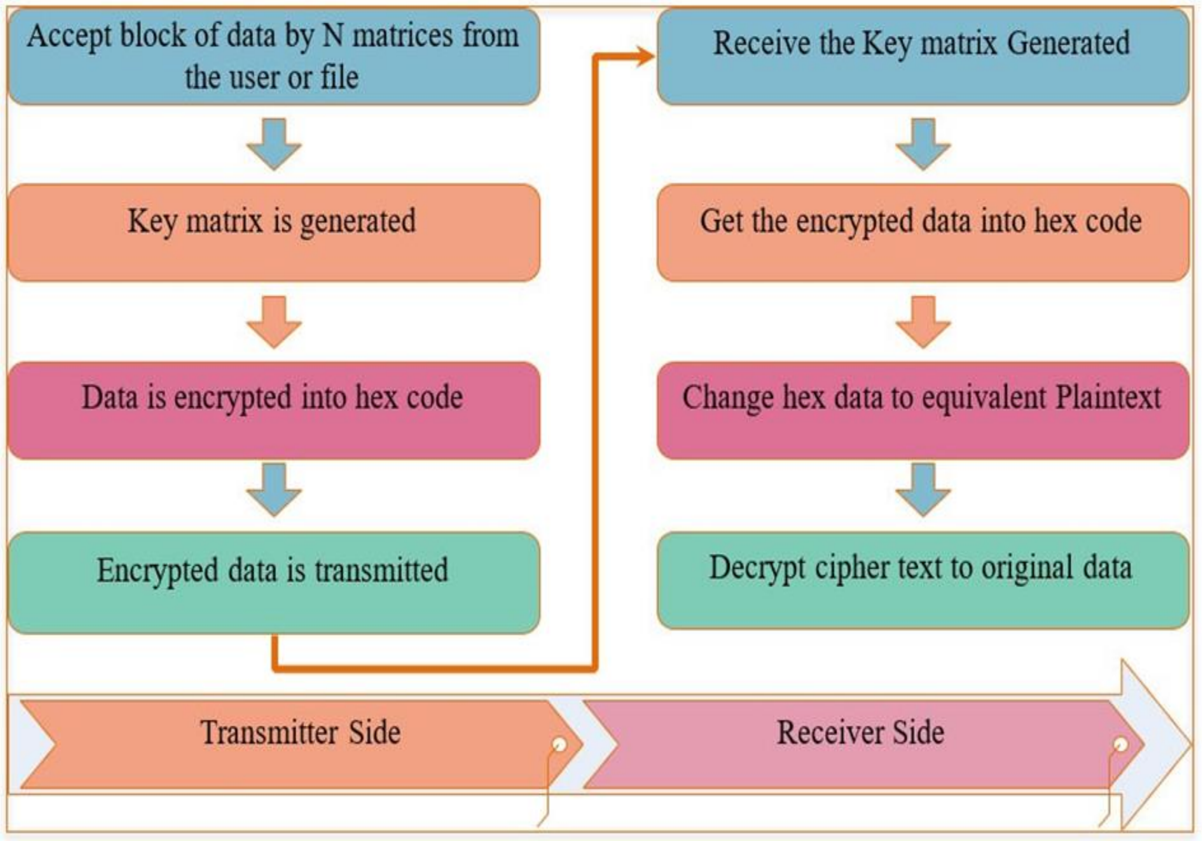

Accept block of data by N matrices from the user or file
Key matrix is generated
Data is encrypted into hex code
Encrypted data is transmitted
Receive the Key matrix Generated
Get the encrypted data into hex code
Change hex data to equivalent Plaintext
Decrypt cipher text to original data
Transmitter Side
Receiver Side

Figure 10: General Process at Transmitter and Receiver Side

The Hill Cipher Algorithm

• What is Hill Cipher?

Hill Cipher, in the pretext of classical cryptography, follows a polygraphic substitution cipher, which means there is uniform substitution across multiple levels of blocks. This polygraphic substitution cipher makes it possible for Hill Cipher to work seamlessly with digraphs (two-letter blocks), trigraphs (three-letter blocks), or any multiple-sized blocks for the purpose of building a uniform cipher.[6]

Hill Cipher is based on linear algebra, the sophisticated use of matrices in general (matrix multiplication and matrix inverses), as well as rules for modulo arithmetic. Evidently, it is a more mathematical cipher compared to others.[6]

The Hill Cipher is also a block cipher. A block cipher is an encryption method that implements a deterministic algorithm with a symmetric key to encrypt a block of text. It doesn't need to encrypt one bit at a time like in stream ciphers. Hill Cipher being a block cipher theoretically, means that it can work on arbitrary-sized blocks.[6]

While Hill Cipher is digraphic in nature, it can expand to multiply any size of letters to add more complexity and reliability for better use. Since most of the problems and solutions for Hill Ciphers are mathematical in nature, it becomes easy to conceal letters with precision.[6]

• Steps for Hill Cipher Encryption and Decryotion:

The stages of the Hill Cipher encryption algorithm are as follows [16]:

1. Organize character alphabetically with numeric A → 1, B → 2, ..., Z → 26 or in ASCII (256 characters)
2. Create a key matrix measuring m x m

$$K_{m*m} = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1m} \\ k_{21} & k_{22} & \dots & k_{2m} \\ \dots & \dots & \dots & \dots \\ k_{m1} & k_{m2} & \dots & k_{mm} \end{bmatrix}$$

3. Matrix K is an invertible matrix that has multiplicative inverse K^{-1} so that K. K^{-1} = 1
4. Plaintext P = p1 p2 ... pn, blocked with the same size as the row or column column K

$$P_{q*m} = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1m} \\ p_{21} & p_{22} & \dots & p_{2m} \\ \dots & \dots & \dots & \dots \\ p_{q1} & p_{q2} & \dots & p_{qm} \end{bmatrix}$$

5. Transpose matrix P and became

$$P_{m*q}^{t} = \begin{bmatrix} p_{11} & p_{21} & \cdots & p_{1q} \\ p_{12} & p_{22} & \cdots & p_{2q} \\ \cdots & \cdots & \cdots & \cdots \\ p_{1m} & p_{2m} & \cdots & p_{qm} \end{bmatrix}$$

6. Multiply matrix K with transposed P in modulo 26 or 256

$$C^{t} = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1m} \\ k_{21} & k_{22} & \cdots & k_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ k_{m1} & k_{m2} & \cdots & k_{mm} \end{bmatrix} \begin{bmatrix} p_{11} & p_{21} & \cdots & p_{1q} \\ p_{12} & p_{22} & \cdots & p_{2q} \\ \cdots & \cdots & \cdots & \cdots \\ p_{1m} & p_{2m} & \cdots & p_{qm} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{21} & \cdots & c_{m1} \\ c_{12} & c_{22} & \cdots & c_{m2} \\ \cdots & \cdots & \cdots & \cdots \\ c_{1q} & c_{2q} & \cdots & c_{mq} \end{bmatrix}$$

7. Then transpose to

$$C=\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1q} \\ c_{21} & c_{22} & \cdots & c_{2q} \\ \cdots & \cdots & \cdots & \cdots \\ c_{m1} & c_{m2} & \cdots & c_{mq} \end{bmatrix}$$

8. Change the result of step 7 into the alphabet using alphabetical correspondence with numeric in step 1 to obtainthe ciphertext.

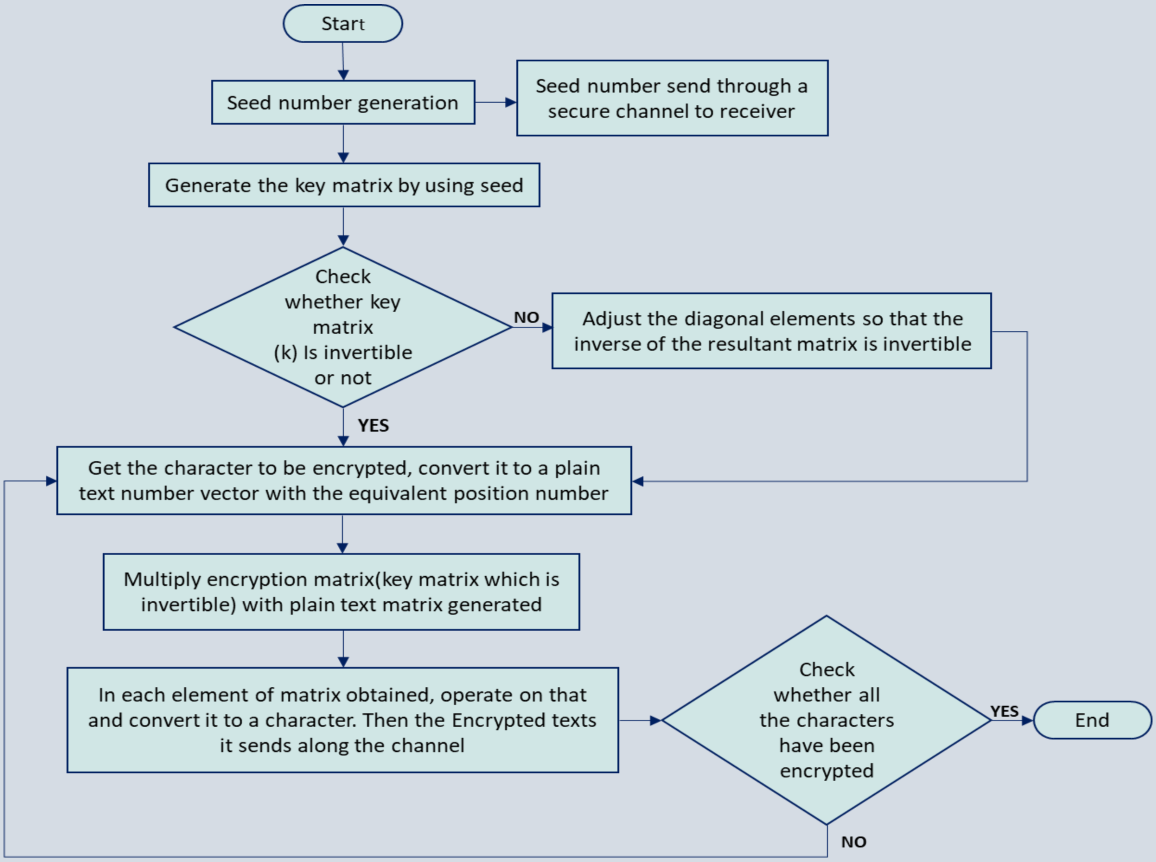

Start
Seed number generation
Seed number send through a secure channel to receiver
Generate the key matrix by using seed
Check whether key matrix (k) Is invertible or not
NO
Adjust the diagonal elements so that the inverse of the resultant matrix is invertible
YES
Get the character to be encrypted, convert it to a plain text number vector with the equivalent position number
Multiply encryption matrix(key matrix which is invertible) with plain text matrix generated
In each element of matrix obtained, operate on that and convert it to a character. Then the Encrypted texts it sends along the channel
Check whether all the characters have been encrypted
YES
End
NO

Figure 11. Flow Chart for Hill Cipher Encryption process

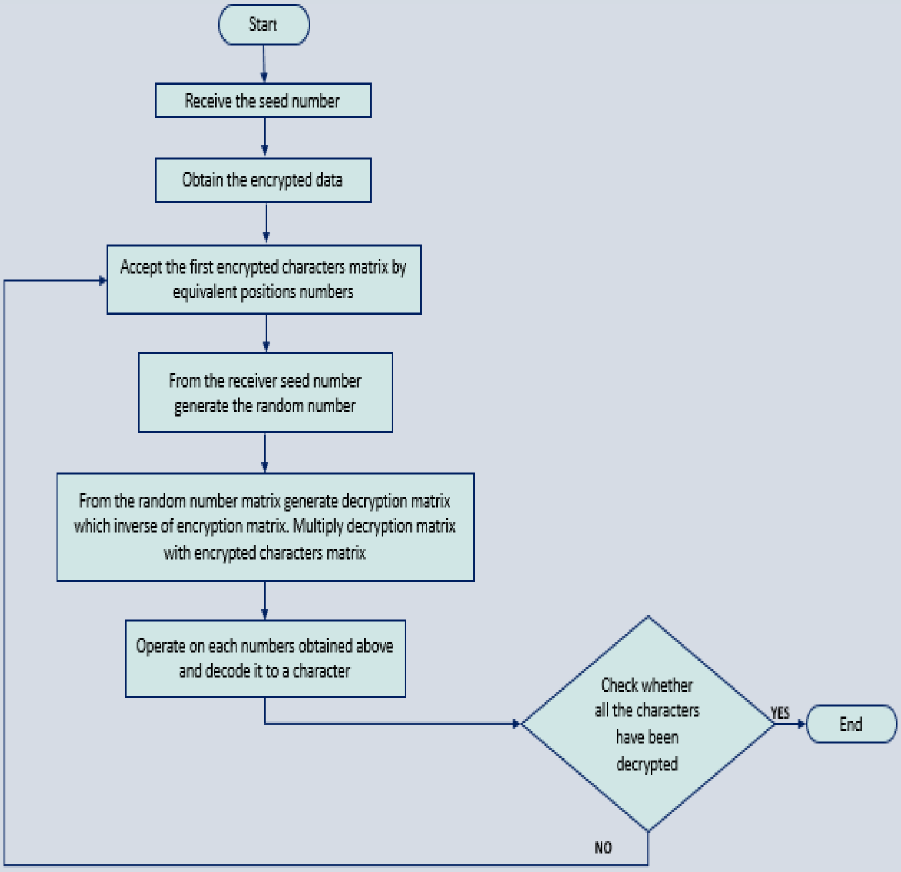

Start
Receive the seed number
Obtain the encrypted data
Accept the first encrypted characters matrix by equivalent positions numbers
From the receiver seed number generate the random number
From the random number matrix generate decryption matrix which inverse of encryption matrix. Multiply decryption matrix with encrypted characters matrix
Operate on each numbers obtained above and decode it to a character
Check whether all the characters have been decrypted
YES
End
NO

Figure 12. Flow Chart for Hill Cipher Decryption process

- ## Hill cipher algorithm example:

as an example of the Hill Cipher technique, let’s encrypt the text, “ACT”, and, later, decrypt the resulting cipher text. This will help us understand how the Hill Cipher works.

To keep the example simple, here is a straightforward substitution scheme with the letter A mapped to 0, B mapped to 1, and so on and so forth.[6]

- Hill Cipher Encryption example:

We have to encrypt the message “ACT” (n=3). The key is “GYBNQKURP”, which in the form of a nxn matrix looks like below:[6]

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}$$

“ACT” is written in the form of the following vector:

$$\begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix}$$

The resulting enciphered vector will be:[6]

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}\begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} = \begin{bmatrix} 67 \\ 222 \\ 319 \end{bmatrix} = \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix}$$

This results in the ciphertext of “POH”.

- Hill Cipher Decryption example:

For the purpose of decryption, the ciphertext will have to be turned back into a vector. Simply, multiply it by the inverse matrix of the key matrix (IFKVIVVMI in letters). The inverse of the matrix will be:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}^{-1} = \begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix}$$

For the previous Ciphertext ‘POH’, multiplying it with the inverse matrix gives:[6]

$$\begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix}\begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} = \begin{bmatrix} 260 \\ 574 \\ 539 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix}$$

The output vector gives back “ACT”.

How is hill cipher secure?

Hill Cipher is a classic cryptographic algorithm that is very strong regarding its security. The Hill Cipher key matrix must be an invertible matrix. The bigger a key matrix, the stronger the security aspect. This algorithm is strong in dealing with ciphertext-only attacks but is weak if attacked with known-plaintext attacks. The thing that needs to be guarded against this algorithm is the key must be confidential. If the key falls to an irresponsible person, then inverse modulo can determine the key with a series of mathematical calculations.

Advantages of the hill cipher:

We can say that the hill cipher has several advantages; it hides the frequent letters in the plaintext, it is also simple because it uses multiplication of matrices in both the encryption and the decryption.[6]

Disadvantages of the hill cipher:

The largest flaw in the hill cipher is that it is a symmetric Encryption, that has only one key in both the encryption and the decryption. This key must be known by both the sender and the receiver, and the key is sufficient to break the secrecy of the message.[6]

RSA algorithm (Rivest-Shamir-Adleman)

What is the RSA algorithm (Rivest-Shamir-Adleman)?

The RSA algorithm (Rivest-Shamir-Adleman) is the basis of a cryptosystem -- a suite of cryptographic algorithms that are used for specific security services or purposes -- which enables public key encryption and is widely used to secure sensitive data, particularly when it is being sent over an insecure network such as the internet.

RSA was first publicly described in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman of the Massachusetts Institute of Technology, though the 1973 creation of a public key algorithm by British mathematician Clifford Cocks was kept classified by the U.K.'s GCHQ until 1997.

Public key cryptography, also known as asymmetric cryptography, uses two different but mathematically linked keys-- one public and one private. The public key can be shared with everyone, whereas the private key must be kept secret.

RSA is a type of asymmetric encryption, which uses two different but linked keys.

In RSA cryptography, both the public and the private keys can encrypt a message. The opposite key from the one used to encrypt a message is used to decrypt it. This attribute is one reason why RSA has become the most widely used asymmetric algorithm: It provides a method to assure the confidentiality, integrity, authenticity, and non-repudiation of electronic communications and data storage.Many protocols, including Secure Shell (SSH), OpenPGP, S/MIME, and SSL/TLS, rely on RSA for encryption and digital signature functions. It is also used in software programs -- browses are an obvious example, as they need to establish a secure connection over an insecure network, like the internet, or validate a digital signature. RSA signature verification is one of the most commonly performed operations in network-connected systems. [11]

• Why is the RSA algorithm used?

RSA derives its security from the difficulty of factoring large integers that are the product of two large prime numbers. Multiplying these two numbers is easy, but determining the original prime numbers from the total -- or *factoring* -- is considered infeasible due to the time it would take using even today's supercomputers.

The public and private key generation algorithm is the most complex part of RSA cryptography. Two large prime numbers, p and q, are generated using the Rabin-Miller primality test algorithm. A modulus, n, is calculated by multiplying p and q. This number is used by both the public and private keys and provides the link between them. Its length, usually expressed in bits, is called the key length.

The public key consists of the modulus n and a public exponent, e, which is normally set at 65537, as it's a prime number that is not too large. The e figure doesn't have to be a secretly selected prime number, as the public key is shared with everyone.

The private key consists of the modulus n and the private exponent d, which is calculated using the Extended Euclidean algorithm to find the multiplicative inverse with respect to the totient of n. [11]

• Steps for RSA:

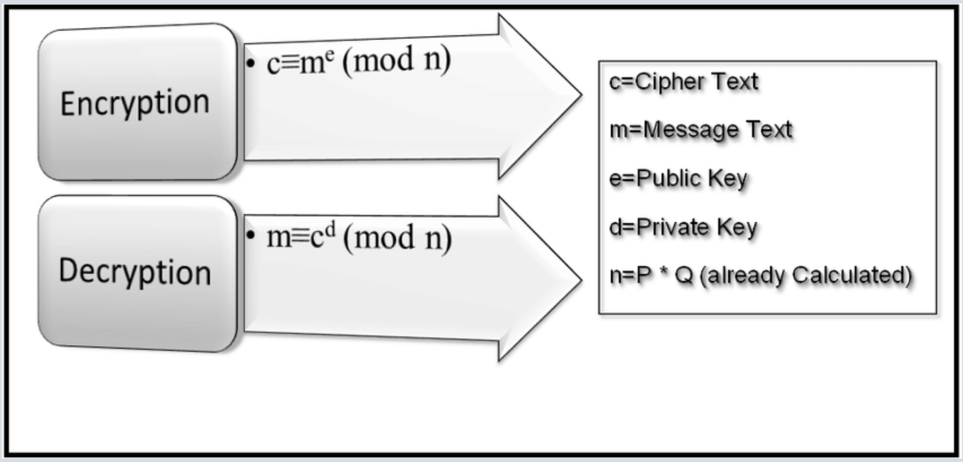


Figure13. main RSA encryption and decryption process

• Key generation

We get two different prime p q

The product of them is n where n is the public key

(n=p*q Φ(n)=(p-1) (q-1))

Choose an e such that 1 < e < φ(n), and such that e Must be coprime with φ(n)

Key is (n,e) [11]

- Message encryption

As we have generated the key, we can change plain text to cipher text by the following formula

$C=p^e \bmod n$

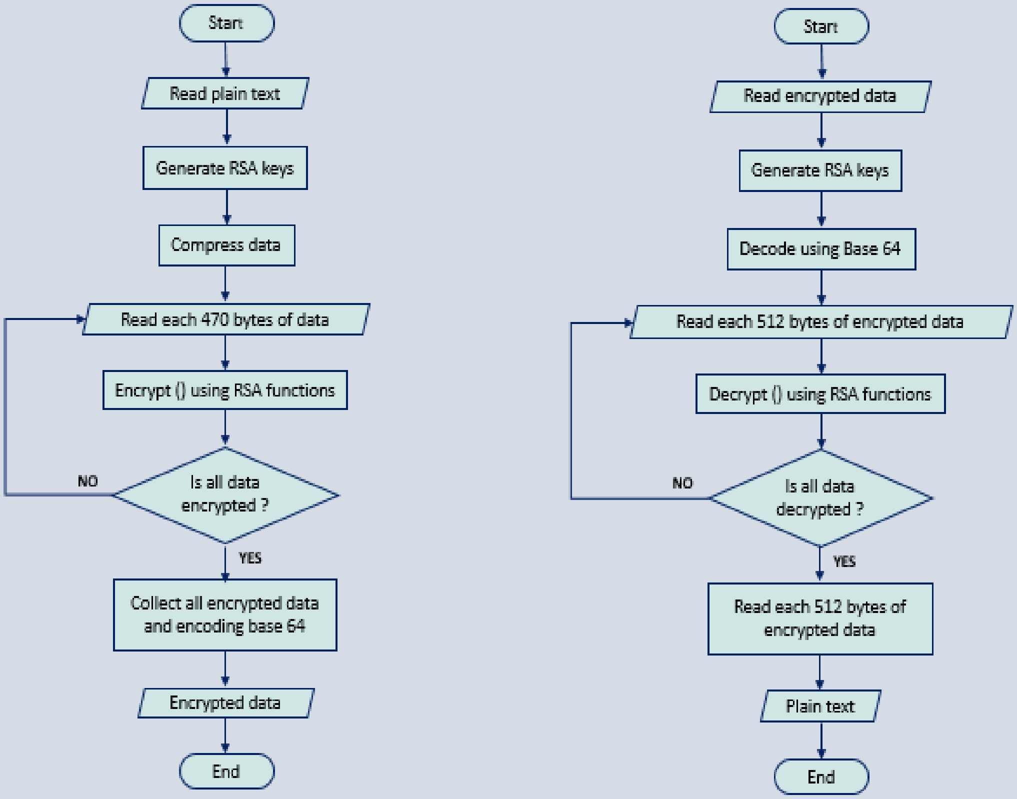
- Message decryption

First, we need to get the private key using the following equation

$d=e-1\bmod\varphi$, d is called modular inverse

After we get the d we can easily decrypt the cipher text to plain text

$p=c^d \bmod n$ [11]



Start
Read plain text
Generate RSA keys
Compress data
Read each 470 bytes of data
Encrypt () using RSA functions
Is all data encrypted ?
NO
YES
Collect all encrypted data and encoding base 64
Encrypted data
End
Start
Read encrypted data
Generate RSA keys
Decode using Base 64
Read each 512 bytes of encrypted data
Decrypt () using RSA functions
Is all data decrypted ?
NO
YES
Read each 512 bytes of encrypted data
Plain text
End

Figure 14. Flow Chart for RSA Encryption and Decryption process

- ## RSA algorithm example:

Alice generates her RSA keys by selecting two primes: p=11 and q=13. The modulus is n=p×q=143. The totient is n φ(n)=(p−1)x(q−1)120. She chooses 7 for her RSA public key e and calculates her RSA private key using the Extended Euclidean algorithm, which gives her 103.
Bob wants to send Alice an encrypted message, M, so he obtains her RSA public key (n, e) which, in this example, is (143, 7). His plaintext message is just the number 9 and is encrypted into ciphertext, C, as follows:

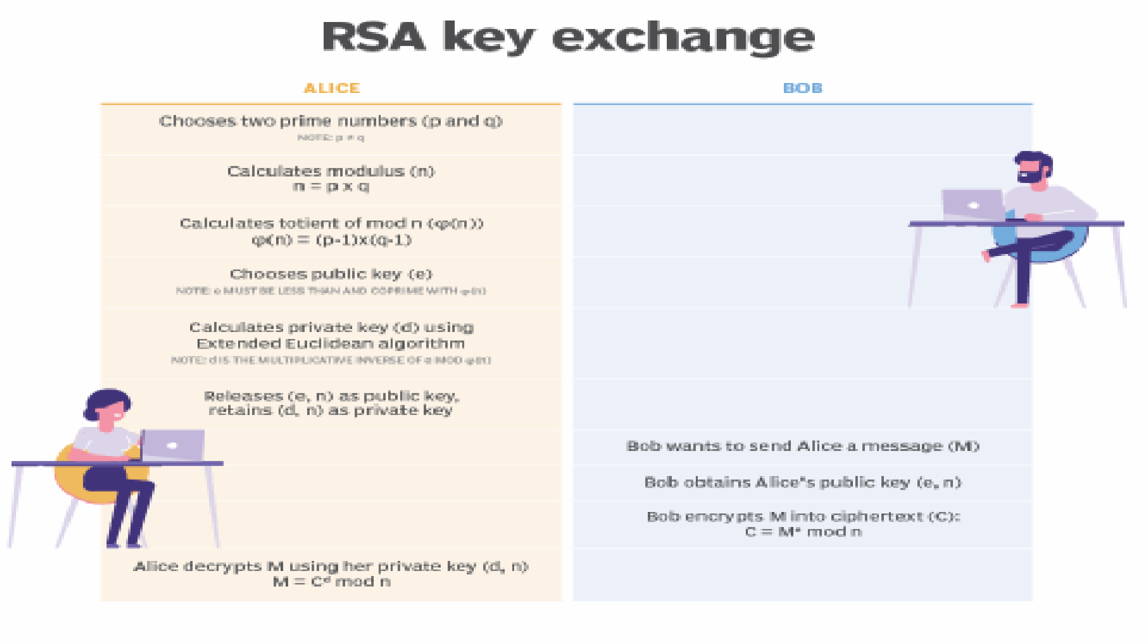
M^e mod n = 9^7 mod 143 = 48 = C

When Alice receives Bob's message, she decrypts it by using her RSA private key (d, n) as follows:

C^d mod n = 48^{103} mod 143 = 9 = M

To use RSA keys to digitally sign a message, Alice would need to create a hash -- a message digest of her message to Bob -- encrypt the hash value with her RSA private key, and add the key to the message. Bob can then verify that the message has been sent by Alice and has not been altered by decrypting the hash value with her public key. If this value matches the hash of the original message, then only Alice could have sent it -- authentication and non-repudiation -- and the message is exactly as she wrote it -- integrity.

Alice could, of course, encrypt her message with Bob's RSA public key -- confidentiality -- before sending it to Bob. A digital certificate contains information that identifies the certificate's owner and also contains the owner's public key. Certificates are signed by the certificate authority that issues them, and they can simplify the process of obtaining public keys and verifying the owner. [11]



RSA key exchange

ALICE

Chooses two prime numbers (p and q)
NOTE: p ≠ q

Calculates modulus (n)
n = p x q

Calculates totient of mod n (φ(n))
φ(n) = (p-1)x(q-1)

Chooses public key (e)
NOTE: e MUST BE LESS THAN AND COPRIME WITH φ(n)

Calculates private key (d) using
Extended Euclidean algorithm
NOTE: d IS THE MULTIPLICATIVE INVERSE OF e MOD φ(n)

Releases (e, n) as public key,
retains (d, n) as private key

Alice decrypts M using her private key (d, n)
M = C^d mod n

BOB

Bob wants to send Alice a message (M)

Bob obtains Alice's public key (e, n)

Bob encrypts M into ciphertext (C):
C = M^e mod n

How is RSA secure?

RSA security relies on the computational difficulty of factoring large integers. As computing power increases and more efficient factoring algorithms are discovered, the ability to factor larger and larger numbers also increases.

Encryption strength is directly tied to key size. Doubling key length can deliver an exponential increase in strength, although it does impair performance. RSA keys are typically 1024- or 2048-bits long, but experts believe that 1024-bit keys are no longer fully secure against all attacks. Therefore, the government, and some industries are moving to a minimum key length of 2048-bits. Barring an unforeseen breakthrough in quantum computing it will be many years before longer keys are required, but elliptic curve cryptography (ECC) is gaining favour with many security experts as an alternative to RSA to implement public key cryptography. It can create faster, smaller and more efficient cryptographic keys. Modern hardware and software are ECC-ready, and its popularity is likely to grow. It can deliver equivalent security with lower computing power and battery resource usage, making it more suitable for mobile apps than RSA.

A team of researchers, which included Adi Shamir, a co-inventor of RSA, successfully created a 4096-bit RSA key using acoustic cryptanalysis. However, note that any encryption algorithm is vulnerable to attack. [11]

Advantages of RSA:

1. It is not hard compared to the other ways of encryption.
2. RSA is so difficult to be attacked as it uses numbers and complex math instead of characters.
3. Sharing public key to users is easy.

Disadvantages of RSA:

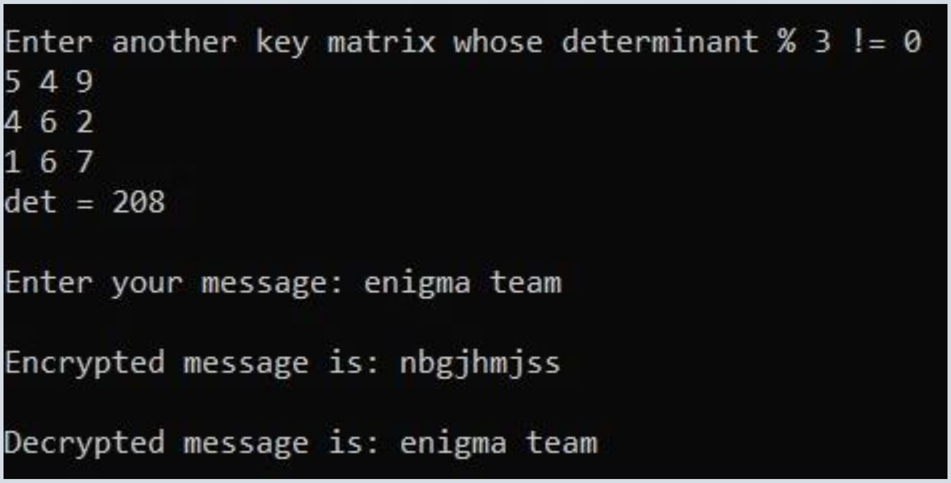
1. It needs high processing as it deals with large numbers.
2. It requires third party to verify the reliability of public keys sometimes.
3. it has slow rate of performance.

Hill cipher experimentation and analysis

First experiment

procedures

we will use two matrices 3x3 with different numbers and different determined, first one a matrix of determined 40 and the second is 208

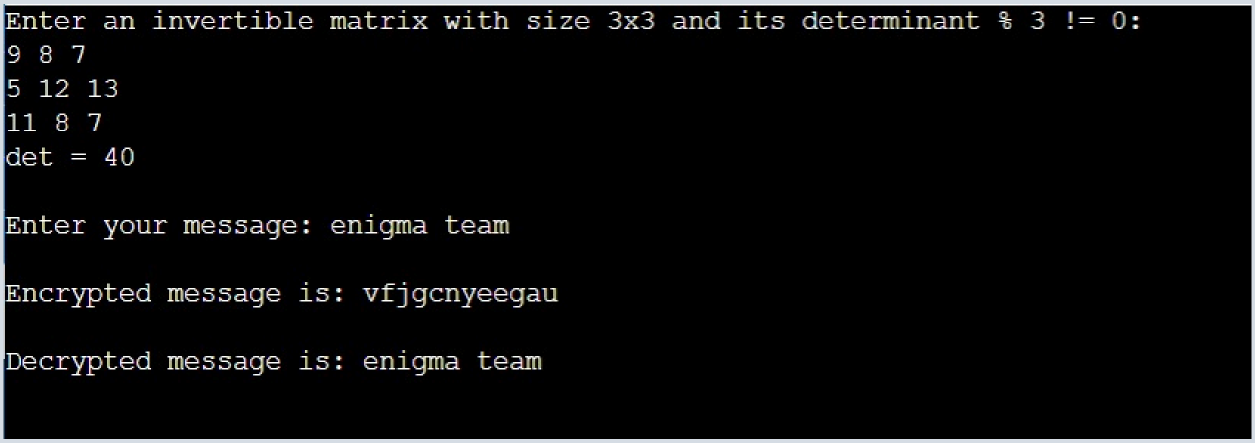


```
Enter another key matrix whose determinant % 3 != 0
5 4 9
4 6 2
1 6 7
det = 208

Enter your message: enigma team

Encrypted message is: nbgjhmjss

Decrypted message is: enigma team
```



```
Enter an invertible matrix with size 3x3 and its determinant % 3 != 0:
9 8 7
5 12 13
11 8 7
det = 40

Enter your message: enigma team

Encrypted message is: vfjgcnyeegau

Decrypted message is: enigma team
```

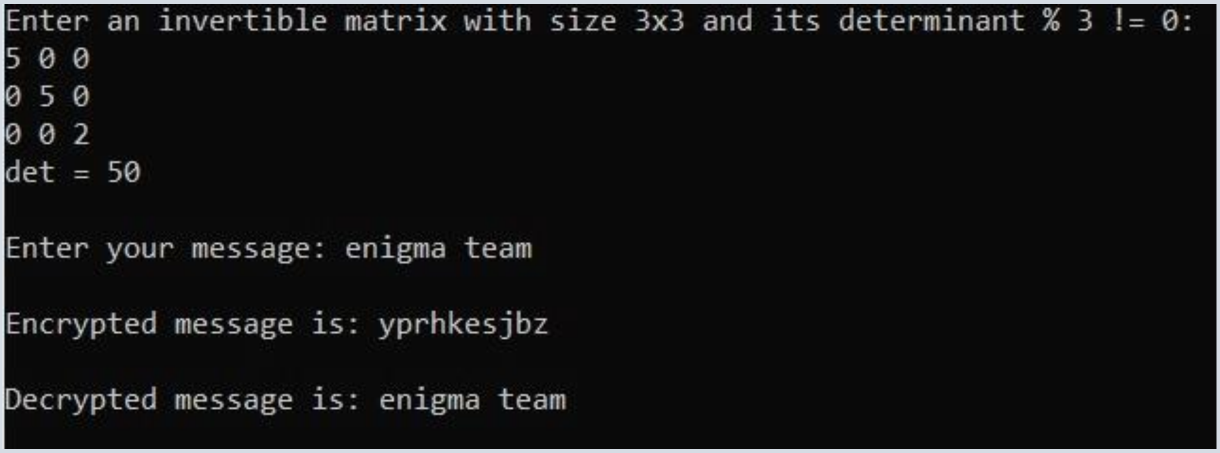
Results

The outputs are not the same, same text is encrypted by different text, case one enigma team change to “nbgjhmjss” while in the second case change into “vfjgcnyeegau”.

second experiment

procedures

we will use two matrices 3x3 with different numbers and same determined, determined=50

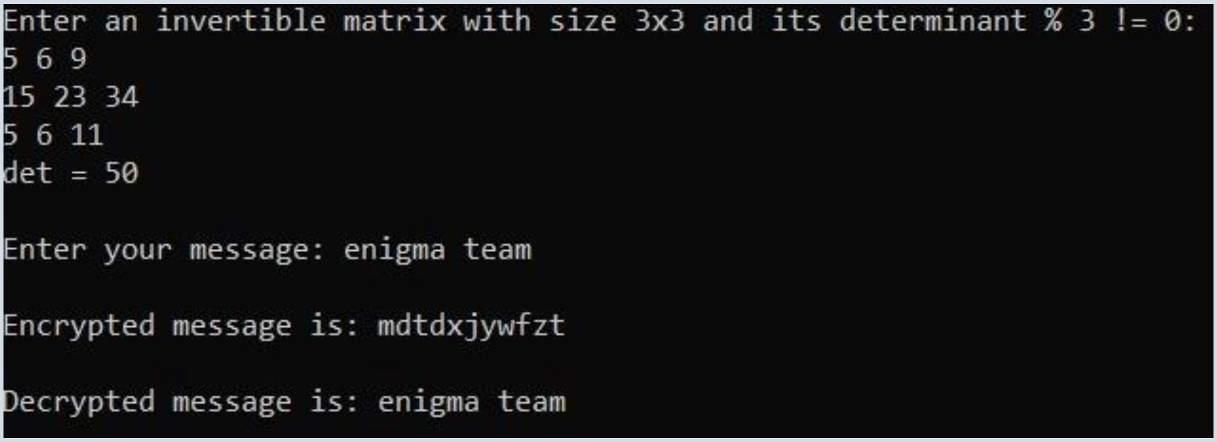


```
Enter an invertible matrix with size 3x3 and its determinant % 3 != 0:
5 0 0
0 5 0
0 0 2
det = 50

Enter your message: enigma team

Encrypted message is: yprhkesjbz

Decrypted message is: enigma team
```



```
Enter an invertible matrix with size 3x3 and its determinant % 3 != 0:
5 6 9
15 23 34
5 6 11
det = 50

Enter your message: enigma team

Encrypted message is: mdtdxjywfzt

Decrypted message is: enigma team
```

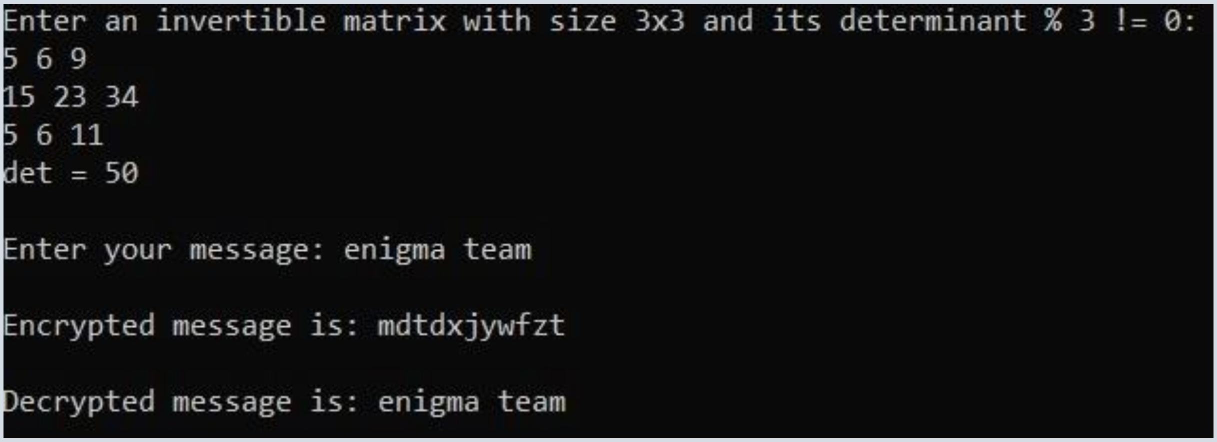
Results

Although the determined are the same the outputs are completely different, same text is encrypted by different text, case one enigma team change to “yprkesjbz” while in the second case change into “mdtdxjywfzt”.

Third experiment

procedures

we will use the same matrix 3x3 before and after Gauss elimination and definitely have the same determined, determined=50

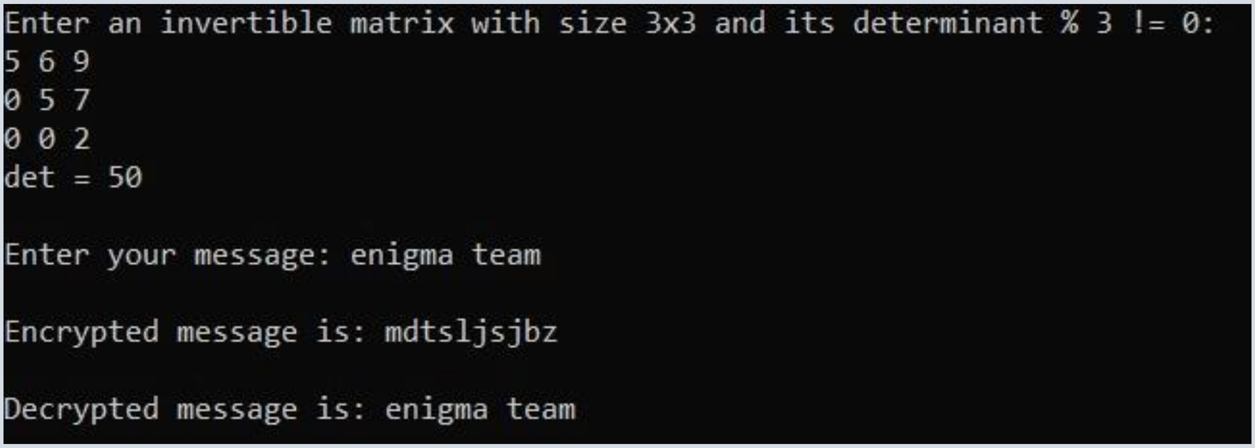


```
Enter an invertible matrix with size 3x3 and its determinant % 3 != 0:
5 6 9
15 23 34
5 6 11
det = 50

Enter your message: enigma team

Encrypted message is: mdtdxjywfzt

Decrypted message is: enigma team
```



```
Enter an invertible matrix with size 3x3 and its determinant % 3 != 0:
5 6 9
0 5 7
0 0 2
det = 50

Enter your message: enigma team

Encrypted message is: mdtsljsjbz

Decrypted message is: enigma team
```

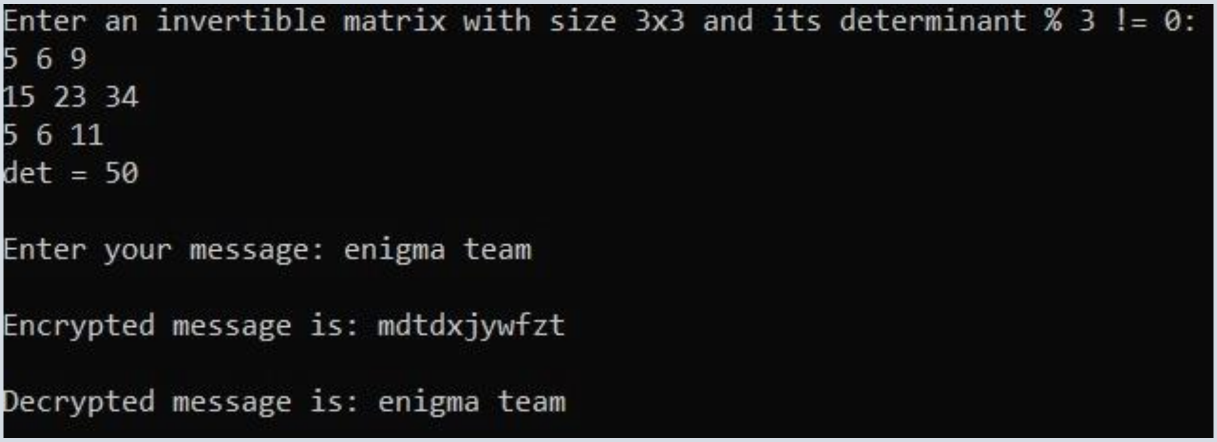
Results

Although they are the same matrix the outputs are different but not completely different, , case one enigma team change to “mdtdxjywfzt” while in the second case change into “mdtsljsjbz”, the first 3 letters are the same ”mdt”.

Fourth experiment

procedures

we will use the same matrix 3x3 and definitely have the same determined, determined=50, but change the arrange of words of the text

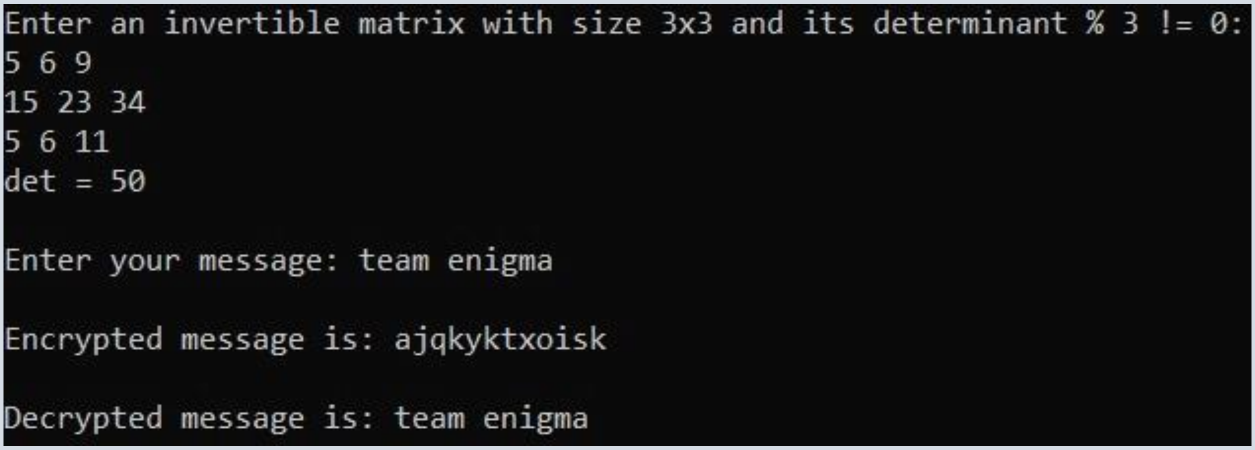


```
Enter an invertible matrix with size 3x3 and its determinant % 3 != 0:
5 6 9
15 23 34
5 6 11
det = 50

Enter your message: enigma team

Encrypted message is: mdtdxjywfzt

Decrypted message is: enigma team
```



```
Enter an invertible matrix with size 3x3 and its determinant % 3 != 0:
5 6 9
15 23 34
5 6 11
det = 50

Enter your message: team enigma

Encrypted message is: ajqkyktxoisk

Decrypted message is: team enigma
```

Results

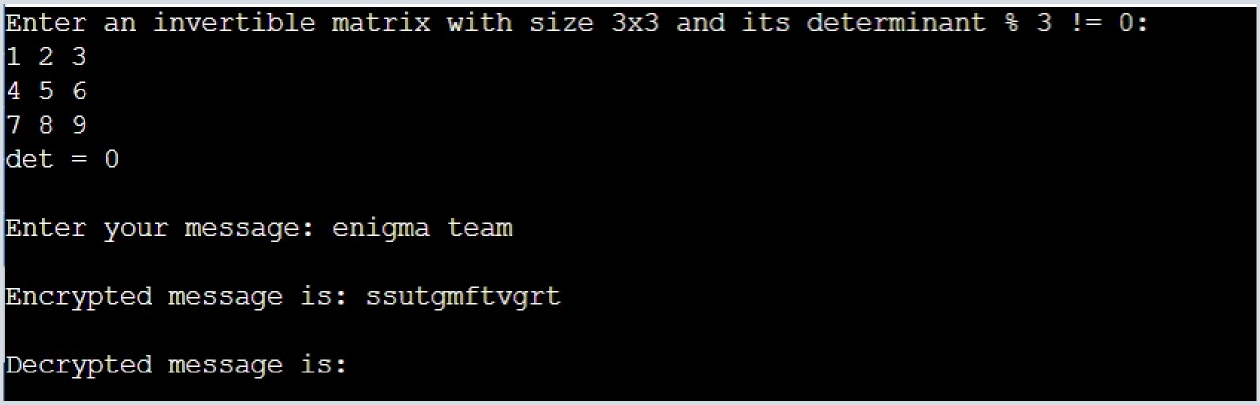
Although they are the same matrix the outputs are completely different, , case one enigma team change to “mdtdxjywfzt” while in the second case team enigma change

into “ajqkyktxoisk”.

Fifth experiment

procedures

we will use a matrix 3x3 of determined equal zero, determined=0,



```
Enter an invertible matrix with size 3x3 and its determinant % 3 != 0:
1 2 3
4 5 6
7 8 9
det = 0

Enter your message: enigma team

Encrypted message is: ssutgmftvgrt

Decrypted message is:
```

Results

There will be no decryption as its not invertible

RSA encryption analysis

Key generation

Key is the most important parameter that has a great effect on the encryption process. The key itself depends on various parameters.

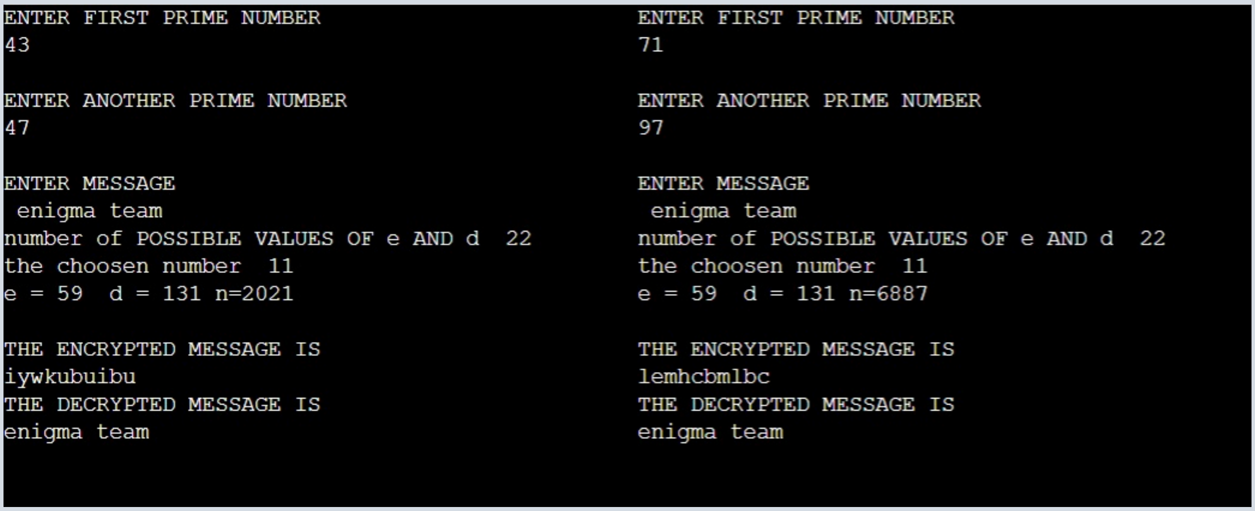
1) p and q the two prime number that we choose to generate the key

First experiment

procedures

First try we will take p=43 q=47 and all other parameters are constant (same text, same e, same d)

Second try we will take p=71 q=97 and all other parameters are constant (same text, same e, same d)



```
ENTER FIRST PRIME NUMBER                        ENTER FIRST PRIME NUMBER
43                                              71

ENTER ANOTHER PRIME NUMBER                      ENTER ANOTHER PRIME NUMBER
47                                              97

ENTER MESSAGE                                   ENTER MESSAGE
 enigma team                                     enigma team
number of POSSIBLE VALUES OF e AND d  22        number of POSSIBLE VALUES OF e AND d  22
the choosen number  11                          the choosen number  11
e = 59  d = 131 n=2021                          e = 59  d = 131 n=6887

THE ENCRYPTED MESSAGE IS                        THE ENCRYPTED MESSAGE IS
iywkubuibu                                      lemhcbmlbc
THE DECRYPTED MESSAGE IS                        THE DECRYPTED MESSAGE IS
enigma team                                     enigma team
```

Results

The outputs are not the same although e and d are the same

2) e and d (modular inverse)

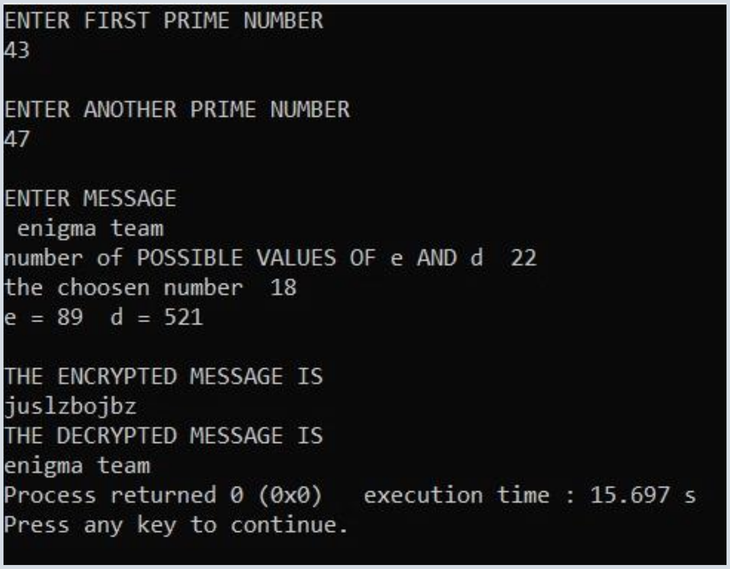
second experiment

Procedures

Choose an e such that 1 < e < φ(n), and such that e Must be coprime with φ(n) and the modular inverse is calculated by d=e^−1 modφ

First try we will take e=89 d=521 and all other parameters are constant (same text, same p, same q)

Second try we will take e=13 d=1189 and all other parameters are constant (same text, same p, same q)

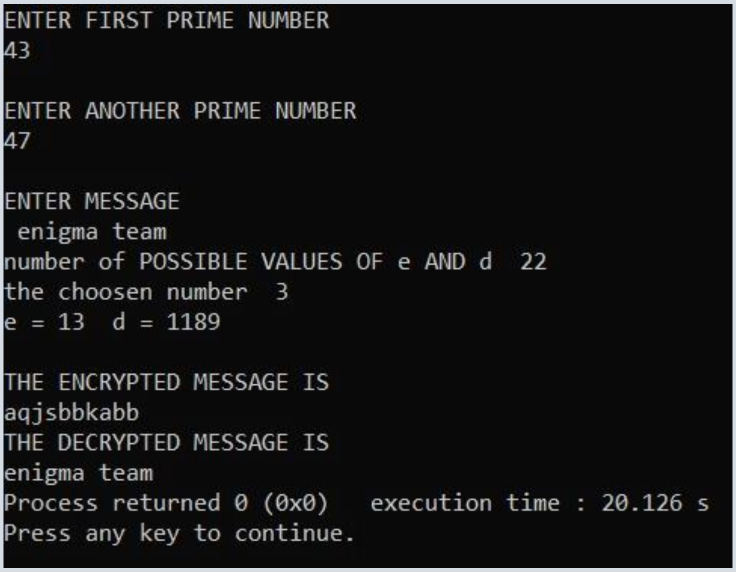


```
ENTER FIRST PRIME NUMBER
43

ENTER ANOTHER PRIME NUMBER
47

ENTER MESSAGE
 enigma team
number of POSSIBLE VALUES OF e AND d  22
the choosen number  18
e = 89  d = 521

THE ENCRYPTED MESSAGE IS
juslzbojbz
THE DECRYPTED MESSAGE IS
enigma team
Process returned 0 (0x0)   execution time : 15.697 s
Press any key to continue.
```



```
ENTER FIRST PRIME NUMBER
43

ENTER ANOTHER PRIME NUMBER
47

ENTER MESSAGE
 enigma team
number of POSSIBLE VALUES OF e AND d  22
the choosen number  3
e = 13  d = 1189

THE ENCRYPTED MESSAGE IS
aqjsbbkabb
THE DECRYPTED MESSAGE IS
enigma team
Process returned 0 (0x0)   execution time : 20.126 s
Press any key to continue.
```

Results

The outputs are not the same although p and q are the same

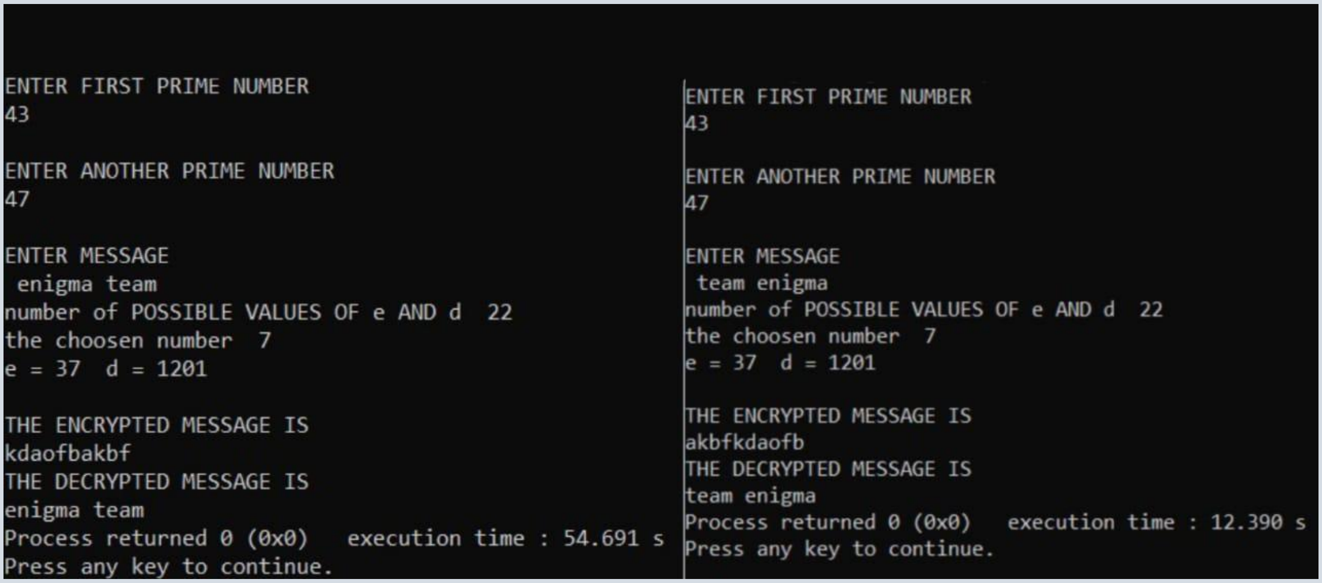
3) plain text

third experiment

procedures

first try the plain text is enigma team (same p, same q, same e, same d)

second try the plain text is team enigma (same p, same q, same e, same d)



```
ENTER FIRST PRIME NUMBER
43

ENTER ANOTHER PRIME NUMBER
47

ENTER MESSAGE
 enigma team
number of POSSIBLE VALUES OF e AND d  22
the choosen number  7
e = 37  d = 1201

THE ENCRYPTED MESSAGE IS
kdaofbakbf
THE DECRYPTED MESSAGE IS
enigma team
Process returned 0 (0x0)   execution time : 54.691 s
Press any key to continue.
```

```
ENTER FIRST PRIME NUMBER
43

ENTER ANOTHER PRIME NUMBER
47

ENTER MESSAGE
 team enigma
number of POSSIBLE VALUES OF e AND d  22
the choosen number  7
e = 37  d = 1201

THE ENCRYPTED MESSAGE IS
akbfkdaofb
THE DECRYPTED MESSAGE IS
team enigma
Process returned 0 (0x0)   execution time : 12.390 s
Press any key to continue.
```

Results

(enigma) change into (kdaofb), (team) change into (akbf) in the two times

The letter a change to b, the letter e change to k

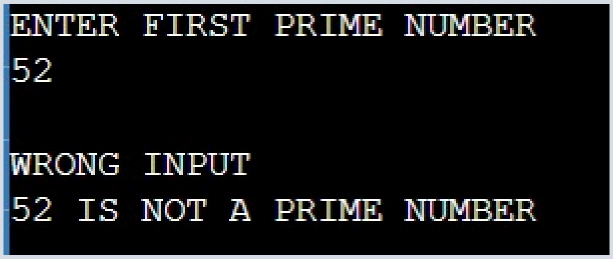
Fourth experiment

Procedures

Entering non-prime number

Let p=52

Results



```
ENTER FIRST PRIME NUMBER
52

WRONG INPUT
52 IS NOT A PRIME NUMBER
```

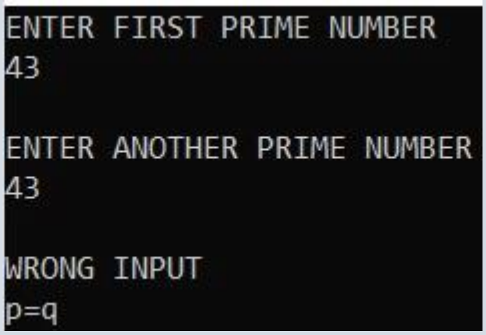
Fifth experiment

Procedures

Entering p and q by the same value p=q

Let p=43 and p=43 (43 is a prime)

Results



```
ENTER FIRST PRIME NUMBER
43

ENTER ANOTHER PRIME NUMBER
43

WRONG INPUT
p=q
```

Codes Link: https://github.com/ahmedmo221/Text-Encryption/tree/main

Conclusion of comparison and future work

- **Conclusion of Hill Cipher experiment:**
    1. When the determined is different the key will be completely different and the inverse also will change .
    2. The elements of the matrix play an important role in the text encryption. If the hacker knew the determined of the used matrix, he still can't decrypt the text.
    3. The elements of the matrix play an important role in the text encryption, but changing one or two element will not completely change the encrypted text.
    4. The plain text is turn into a matrix then be multiplied by the key so change the text or change the arrange of the words will change the encrypted text.
    5. The determined of the matrix should be invertible
- **Conclusion of RSA experiment:**

1. The encrypted text depends on the value of n (product number of p and q)
   So when we change the two prime numbers the value of n also change
2. As we have generated the key, we can change plain text to cipher text by the following formula C=M^e mod n
3. The encrypted text depends on the value of e (modular invers)
4. Encryption formula depends on e C=M^e mod n
5. d is called the private key which based on the value of modular inverse that we choose
6. as the value of φ increase number of suitable e increase ,it could be 22 or more option
7. As p, q, n, e and d are the same, each letter will change to certain letter as this method work on each letter individually
8. p and q shouldn't be equal

General Conclusion:

1. The study of various algorithms shows that the model's strength depends upon the key management type of cryptography, number of keys, number of bits used in a key. All the keys are based on mathematical properties. The keys having a greater number of bits requires more computation time, indicating that the system takes more time to encrypt the data.
2. Each of cryptographic algorithms has weakness points and strength points. We select the cryptographic algorithm based on the demands of the application that will be used.

Future work:

Increasing computing power will soon make existing encryption algorithms ineffective. Here’s how the industry is responding and how your agency can benefit from new encryption innovations today.

the encryption arms race is about to become much more challenging. That’s because quantum computing will soon make processors vastly more powerful. That eventuality will require significant advances in encryption. [13]

1) Expect Double Exponential Growth via Quantum Computing

Classical computing uses electrical signals to encode data in bits. Each bit can have a value of 0 or 1. Quantum computing can use other physical systems, such as electrons and protons, to encode data in qubits. A qubit can have a state of 0, 1 or some combination of those digits. Because the quantum state of a qubit can be almost infinite, a qubit can encode exponentially more data than a bit.

Classical computing performance has experienced exponential growth, increasing by powers of 2 (2, 4, 8, 16, etc.). Quantum computing is expected to involve double exponential growth, increasing by powers of powers of 2 (4, 16, 256, 65,536, etc.). You can see how quickly the performance of quantum computers will leap ahead.

What’s more, quantum computers are ideally suited to the integer factorization needed to crack today’s asymmetric encryption algorithms. Common public key cryptosystems such as RSA could become trivially easy to break. [13]

2) Making Encryption Harder, Better, Faster and Stronger

In response, the industry is advancing encryption on several fronts. Some efforts are focused on increasing key sizes to protect against brute-force decryption. Other efforts are looking at new cryptographic algorithms. For example, the National Institute of Standards and Technology is evaluating a next-generation public key algorithm intended to be quantum safe.

The trouble is that most quantum-safe algorithms aren’t efficient in classical computer architectures. To address this problem, the industry is focused on developing accelerators to speed up algorithms on x86 platforms.

A third area of research is homomorphic encryption, an amazing concept that allows users to perform calculations on encrypted data without first decrypting it. So, an analyst who needs to can query a database containing classified information without having to ask an analyst with higher clearance to access the data or request that the data be declassified.

A big advantage of homomorphic encryption is that it protects data in all its states — at rest (stored on a hard drive), in motion (transmitted across a network) or in use (while in computer memory). Another boon is that it’s quantum safe, because it’s based on some of the same math as quantum computing.

A downside is that homomorphic encryption performs very poorly on traditional computers, because it’s not designed to work with them. The industry is collaborating to develop x86-style instructions to make these new cryptosystems operate at cloud speeds. Practical applications are still a few years away, but we’re confident we’ll get there. [13]

Applications of text encryption

For most of recorded history, encryption has been used to protect the secrecy of communications between a sender and a receiver. Governments have historically been heavy users of encryption. The Caesar cipher goes back to the Roman Empire. Ciphers were used by both sides in the American Revolutionary War. Histories of World War II dwell at length on the contribution of defeating German and Japanese encryption systems to the Allied victory. At the same time, the Allies also relied on encryption systems, some of which were defeated by Axis codebreakers. Governments’ reliance on encrypted communications continues to the present day.[8]

In recent years, encryption has become far more widely available on a wide range of consumer and business products and services. Increasingly, encryption is available by default—often without the user even being aware of it—and the keys for decrypting data are held by individual users. As a result, more data is routinely encrypted today than ever before.[8]

Today, encryption protects the communications of individuals and organizations from unsophisticated and sophisticated criminals and repressive governments. It assures the security of electronic commerce transactions over the Internet—for example making it possible to transmit credit card numbers. It protects information stored on smartphones, laptops, and other devices. Encrypted communication capabilities are built into major computing platforms and in an array of messaging applications that are used by hundreds of millions of users.[8]

Computer and communications systems use cryptography for three broad purposes—to protect the confidentiality of information National Academies of Sciences, Engineering, and Medicine. [12]

References :

[1] Encryption. (n.d.). Retrieved May 13, 2022, from https://mathweb.ucsd.edu/~crypto/students/crypt.html

[2] *A brief history of encryption*. Thales Group. (2021, October 1). Retrieved May 13, 2022, from https://www.thalesgroup.com/en/markets/digital-identity-and-security/magazine/brief-history-encryption

[3] Jun7 5 Common Encryption Algorithms and the Unbreakables of the FutureIT News. (2021, June 22). *5 common encryption algorithms and the unbreakables of the future*. StorageCraft Technology, LLC. Retrieved May 13, 2022, from https://blog.storagecraft.com/5-common-encryption-algorithms/

[4] *Figure : Block Diagram for AES encryption and decryption*. (n.d.). Retrieved May 13, 2022, from https://www.researchgate.net/figure/Block-diagram-for-AES-encryption-and-decryption_fig1_324796235

[5] *Flow chart diagram for the encryption and decryption process*. (n.d.). Retrieved May 13, 2022, from https://www.researchgate.net/figure/Flow-chart-diagram-for-the-encryption-and-decryption-process_fig11_235760962

[6] *What is Hill cipher? explained with step-by-step example*. Intellipaat Blog. (2022, February 10). Retrieved May 13, 2022, from https://intellipaat.com/blog/what-is-hill-cipher/#no3

[7] Cryptography. (n.d.). Retrieved May 13, 2022, from https://sites.math.washington.edu/~king/coursedir/m308a01/Projects/Cryptography.htm

[8] Mishra, N. (2018, September 13). *Hill cipher in C and C++ (encryption and decryption)*. The Crazy Programmer. Retrieved May 13, 2022, from https://www.thecrazyprogrammer.com/2017/02/hill-cipher-c.html

[9] *C++ program to implement the RSA algorithm*. Sanfoundry. (2021, September 9). Retrieved May 13, 2022, from https://www.sanfoundry.com/cpp-program-implement-rsa-algorithm/

[10] Bhattacharya, P. B. A. (2022, February 11). *Comparing encryption algorithms: Encryption Consulting*. Encryption Consulting | Encryption Consulting. Retrieved May 13, 2022, from https://www.encryptionconsulting.com/comparison-of-various-encryption-algorithms-and-techniques-for-securing-data/

[11] Cobb, M. (2021, November 4). *What is the RSA algorithm? definition from searchsecurity*. SearchSecurity. Retrieved May 13, 2022, from https://www.techtarget.com/searchsecurity/definition/RSA

[12] *Read "decrypting the encryption debate: A Framework for Decision Makers" at nap.edu*. 2 Encryption and Its Applications | Decrypting the Encryption Debate: A Framework for Decision

Makers |The National Academies Press. (n.d.). Retrieved May 13, 2022, from https://www.nap.edu/read/25010/chapter/4#16

[13]Steve Orrin Steve Orrin is the federal CTO for Intel Corporation. He has held architectural and leadership positions at Intel, driving strategy and projects on identity. (2021, July 29). *The Future of Data Encryption: What You Need To Know Now*. Technology Solutions That Drive Government. Retrieved May 13, 2022, from https://fedtechmagazine.com/article/2021/07/future-data-encryption-what-you-need-know-now

[14]Simplilearn. (2022, March 4). *What is Data Encryption: Algorithms, methods and techniques [2022 edition]: Simplilearn*. Simplilearn.com. Retrieved May 13, 2022, from https://www.simplilearn.com/data-encryption-methods-article

[15]Data Encryption and decryption by using Hill cipher algorithm. (2020). *Control Theory and Informatics*. https://doi.org/10.7176/cti/10-01

[16] M. Khoerudin, “Algoritma Hill Cipher (Sandi Hill),” 2015. [Daring]. Tersedia pada:

https://muamalkhoerudin.wordpress.com/2015/03/22/algoritma-hill-cipher-sandi-hill/

. [Diakses: 01-Okt-2018].

THE END

Thanks to Dr.samah Elshafiey