LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC5009NI Cyber Security in Computing

Assessment Weightage & Type

40% Individual Coursework 01

Year and Semester

2024 -25 Autumn Semester

Student Name: Prabesh Sundar Taksari

London Met ID: 23047464

College ID: NP01NT4A230138

Assignment Due Date: November 24, 2024

Assignment Submission Date: January 20, 2025

Word Count (Where required): 13216

I confirm that I understand my coursework needs to be submitted online via My Second Teacher under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Year and Semester

2024 -25 Autumn Semester

Student Name: Prabesh Sundar Taksari

London Met ID: 23047464

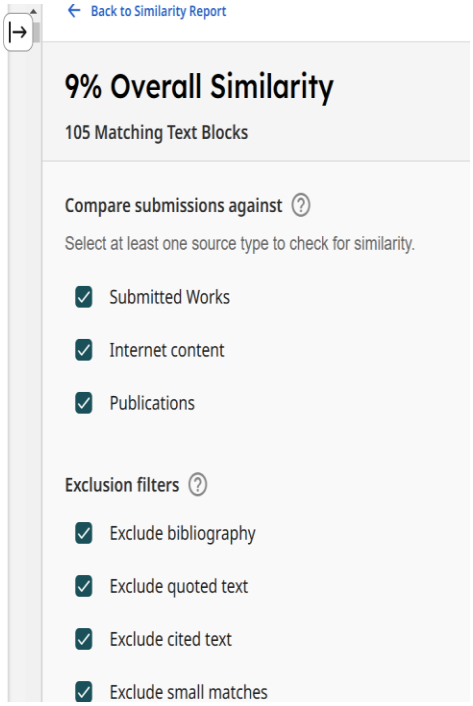
College ID: NP01NT4A230138

Assignment Due Date: November 24, 2024

Assignment Submission Date: January 20, 2025

Word Count (Where required): 13216

I confirm that I understand my coursework needs to be submitted online via My Second Teacher under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.



Abstract

This research presents a new cryptographic algorithm called Quadracrypt Cipher which is a multiple level cryptographic technique for improving security in information transfer. This algorithm comprises both the Playfair and Caesar Ciphers from the classical encryption type, together with left-shifting, diagonal-shifting, binary XOR operations, and custom mapping, enabling one to counter contemporary threats of cryptographic security. Quadracrypt Alphabet uses eight consecutive conversion forms or phases for encryption and decryption to meet modern security challenges, including direct assault, statistical analysis and other approaches that make use of statistical analysis, including frequency and pattern analysis.

The versatile approach of the algorithm allows the application of key focuses and configurations in different ways which makes it versatile. The experiments on plaintext data proved its capability to produce random ciphertext with fast decryption procedures. Uses of the Quadracrypt Cipher involve sectors that should not afford to have leakages in matters concerning data security such as the finance sector, government, military, healthcare, commerce and cloud computing.

All the same, the algorithm has its strengths that include increasing security and protection against recognizable attacks and some drawbacks that include high computation and susceptibility to repetitive patterns in specific settings. The following areas are given below containing scope for future research and optimization. All in all, the Quadracrypt Cipher highlights a further enhancement of the encryption method in the existing world by providing useful techniques for protecting sensitive information from interoperability.

Table of Contents

Table of Figures

Table of Table

1. Introduction

1.1. Security

Securing refers to the deeds and methods made to secure the people, organizations and systems against dissimilar dangers such as physical damage, the cyberattacks, and other attack. In the subject of information technology, security refers to a range of strategies meant to secure digital assets against disturbances or exploitation by malicious actors. What this includes is both physical security like surveillance systems to physical controls as well as information security methods including the encryption of the data and firewalls. With the continuing complexity of threats in today's environment coming at us from all sides, every effort, including several layers of defence, need to be brought to bear to mitigate risks. In the end though, it's not just about stopped the attacks, security is also about resilience: make sure that systems can get back up and running after a challenge. (Bacon, 2024)

1.2. CIA Triad

The CIA Triad is a foundational model in information security that consists of three core principles: These are Confidentiality, Integrity and Availability. This framework is used to follow a security policy and procedures for organizations to protect sensitive information from numerous threats.

Figure 1 CIA Triad

- Confidentiality

  Only legally permitted parties are given accessibility to the secured information to maintain confidentiality. It includes strategies like authentication, limitations on access, and encryption to block access. This has been designed to protect confidentiality of information and enables organizations to prevent data breaches. That is crucial for preserving confidence and maintaining operating by rules.

- Integrity

  Integrity makes sure that during its entire existence, data remains accurate and unharmed. It involves identifying illegal modifications using techniques like electronic signatures and hashing. To ensure the accuracy of information, the integrity of data must be maintained. Based on accurate facts, this helps organizations in making intelligent choices.

- Availability

  Availability provides accurate availability of information and facilities for approved individuals when they're needed. Techniques such as redundant protection, backup systems, and regular service are utilized for achieving this. Maintaining availability minimizes interruptions and ensures the operation of essential services. Both user happiness and the continuity of business are dependent on it.

The CIA Triad provides a framework to develop comprehensive security policies and procedures inside organizations. Organizations may establish strong security strategies that successfully defend against a range of threats while maintaining continuous information access by establishing a balance between the following three elements: confidentiality, integrity, and availability. (Fasulo, 2021)

1.3. Cryptography

Cryptography, the art and science of confidential communication, conducted for thousands of years, its earliest document implementations date to ancient Greece and Egypt. Cryptography has progressed over the years, from basic substitution ciphers to complex methods of encryption, and it has been essential for protecting private data and sensitive information. With the development of computers and the rise of electrical communications, cryptography has also evolved, and it is still developing. (Krichen, 2023)

1.4. Terminologies of Cryptography

The following elements represent the basis for cryptography, which enables encrypted communication. The original accessible information, or plaintext, represents what an individual wants to consistently provide to a destination without it getting intercepted by other parties. This plaintext is transformed all through the process of encryption to ciphertext, which is a meaningless or incomprehensible form of communication that is challenging for unidentified individuals to decode. An algorithm for encryption and a key at the end of the message are utilized to transform plaintext to ciphertext, which is called encryption. The opposite method, called decryption, involves transforming ciphertext to its original plaintext at the point of receipt applying a decryption algorithm and the exact same or the same key. The key, which could be symbolic, alphanumeric, or numeric, is crucial as it performs an essential part in the cryptographic process's reliability. Additionally, Hashing is utilized to ensure data integrity by generating a unique electronic fingerprint of the data, which is often used in document authentication and password storage processes. (CISSPPREP, 2024)

1.5. The Origins and Evolution of Cryptography

Cryptography originates in ancient Egypt (around 1900 BCE) when Khnumhotep ll's tomb illustrates that how hieroglyphs were modified to hide communications. By 1500 BCE, business secrets were being protected by the Babylonians through ancient encryption. In 500 BCE, the the Spartans used a letter-transposition system called the

scytale cipher. One of the first replacement ciphers, the Caesar cipher, was designed by Julius Caesar in 100 BCE. (Cryptool, 2024)

The history of cryptography has been separated into three different phases. During World War I, Cryptography is a key component of communications and surveillance tactics. Traditional encryption systems were susceptible to monitoring at the time due to the increasing use of radio communications. Cryptography was constrained by the skills of a code clerk, who was assisted only by basic mnemonic devices. This led advancements in encryption to safeguards Confidentiality. Governments used manually early mechanical cryptography tools created by Parker Hitt, an officer in the U.S. Army. To lower the possibility of compromising, codes were changed frequently; however, the quick distribution of new codebooks caused logistical difficulties. (Britannica, 2024)

After World War I, Cryptography evolved into the mechanization phase, using technologies like punch tapes, telephone switches, and Marchants and Brunsvigas calculators. During World War II, rotor machines (like Enigma) were invented which helps to enable faster and more complex encryption and decryption. Innovations including the Advanced Encryption Standard (AES) and Data Encryption Standard (DES) reached processing rates of billions of bits per second by the end of the 20th century. This represents a billionfold growth in encryption. (Krichen, 2023)

In the last two decades of the 20th century, the third phase of cryptography was developed, introducing with it an important shift toward distributed cryptography, digital signatures and verification. The requirement for technological advances to complete responsibilities formerly performed by paper documents propelled this era. The time was about modifying cryptography to the needs of the information age, permitting safe digital communications and operations although public-key cryptography was still highly significant. These advancements have made it possible to control electronic information more effectively and securely.

1.6. Symmetric and Asymmetric

Symmetric: The method of changing a message's structure to avoid unauthorized access is known as encryption. Symmetric encryption is not secure as a key is used to encrypt the information, and the same key is used to decrypt it. Also, there is a requirement of a secure way to transfer the key between such parties (Authority, 2024).

Asymmetric: The most common in all cryptography techniques is the asymmetric key encryption, which essentially uses a single key, and its opposite encrypt the data and decrypt the encrypted data. One key, called a private key, keeps its contents completely confidential, while another, which is known as a public key, can be exchanged freely amongst service users. (Authority, 2024)

Difference Between Symmetric and Asymmetric Key Encryption

Symmetric Key Encryption	Asymmetric Key Encryption
Only a single key will be required for encryption and decryption.	Public key and private key are needed one for encrypt other for decrypt.
The ciphertext's size is the same or lower than the original plaintext’s.	The ciphertext’s size is the same or greater than the original plaintext’s.

The encryption operation is quite fast.	The encryption operation is slow.
This key length has several bits that could be either 128 or 256.	This key length is used 2048 or higher.

Table 1 Difference Between Symmetric and Asymmetric Key Encryption

This are the some of the difference between the Symmetric and Asymmetric Key encryption. (GeeksForGeeks, 2024)

1.7. Aim And Objectives

The aim is to develop a safe and reliable cryptographic method that makes use of innovative encryption and decryption algorithms to protect sensitive data. The method promises to offer protection against contemporary cryptographic threats and be performance-optimized for many kinds of devices. It aims to provide confidentiality, strong information integrity, and authenticity for trustworthy communication.

Objectives

- To develop a technique that uses an exclusive pattern of transposition and swapping procedures to ensure secure encryption and decryption.
- To develop a reliable algorithm that functions on several kinds of devices, including systems with exceptional performance and low-power devices.
- To promote secure interaction through providing excellent protection against cryptographic vulnerabilities like differential evaluation and brute force attacks.

- To keep implementation simple and to ensure smooth connection with current safety systems.
- To carefully assess the algorithm's reliability, safety, and accuracy in a variety of settings.

2. Background

2.1. Playfair Cipher

In 1854, the British cryptographer Sir Charles Wheatstone developed the Playfair cipher, a manual encryption algorithm which encryption digraphs pairs of letters instead of individual letters. The name was inspired after the Lyon Playfair promoted it for its use. The Playfair cipher, in contrast with simpler ciphers, improves security by making the analysis of frequencies, an effective cryptographic attack, more challenging.

This cipher uses a 5x5 matrix of characters to encrypt information depending on keyword. Because it provided a secure yet realistic method of encryption for field utilization, it was used in military communications, especially by British soldiers during Second Boer War (1899-1902) and the World War I (1914-1918). Although it olds historical importance, Because of weaknesses like digraph frequency analysis, the Playfair encryption cannot be considered as secure by current criteria. After this in 1940s, Australia, Germany, and New Zealand implement the Playfair cipher. Because of its complex and difficult to throw off cryptographer's but didn't require any specific tools or equipment to solve.

Example of Playfair Cipher

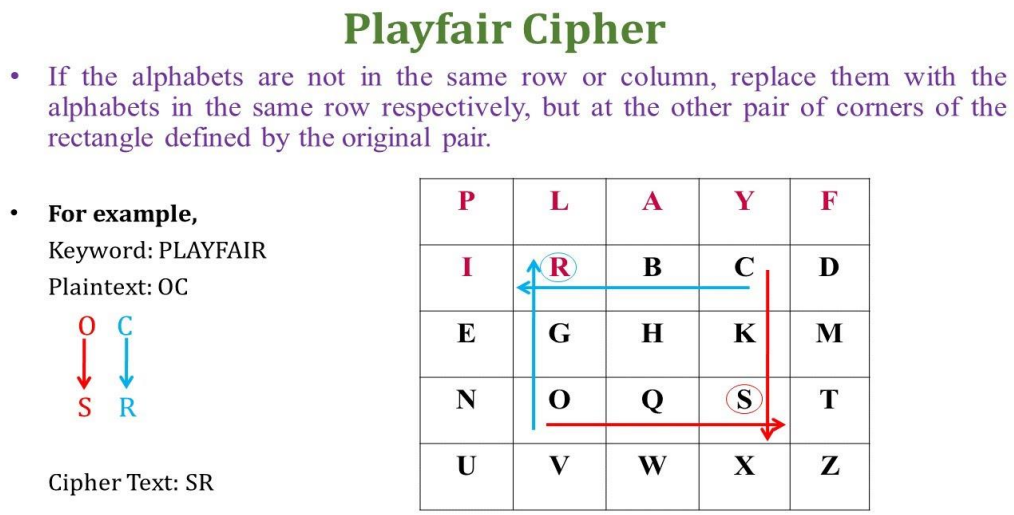


Figure 2 Example of Playfair Cipher

The picture explains how the Playfair cipher maps two different letters but in the same row or in the same column with another pair of letters, a rectangle in the cipher grid with the two letters forming opposite corners. (Youtube, 2019)

2.2. Advantages and Disadvantages of Playfair Cipher

Aspect	Advantage (Pros)	Disadvantage (Cons)
Simplicity	It is easy to comprehend and apply with nothing more than a key matrix.	Necessitates the use of manual rules for elements such as the treatment of the letter "J" and repeated letters, which may lead to difficulties.
Security	Substitutes digraphs (the encryption of two letters at one trial) instead of the single letters providing more security than monoalphabetic cipher.	Yet susceptible to digraph probability analysis and other present brand techniques.
Speed	Quicker than the use of transposition or other polyalphabetic forms of cipher to manually encrypt and decrypt.	Time consuming and involving several mistakes in case the decryption is done manually due to several procedures.
Key Characteristics	The key matrix removes repeated letters, which is a further complication of the approach.	This restricts the flexibility, as it occupies a 5 x 5 fixed space, and is not easily expandable to a much larger character set or superior encrypted security.
Known Plaintext Attacks	Is based on the idea to enhance monoalphabetic ciphers and encrypt two letters while are harder to decipher by frequency analysis.	When an attacker has plaintext and ciphertext pairs, this approach allows him to learn portions of the key matrix degrading security.

Table 2 Advantages and Disadvantages of Playfair Cipher

These are the Advantages and Disadvantages of Playfair Cipher. (Unext, 2022)

2.3. Workings of Playfair Cipher

Playfair Cipher Example

Let's take the following as an example:

- **Keyword**: "EXAMPLE"
- **Plaintext**: "HELLO WORLD"

Step 1: Create the 5x5 Matrix (Key Table)

1. First, we form the 5x5 key matrix. We eliminate duplicate letters and combine "I" and "J" to fit the 25-letter grid.

Keyword: EXAMPLE

Matrix:

E	X	A	M	P
L	B	C	D	F
G	H	I/J	K	N
O	Q	R	S	T
U	V	W	Y	Z

Step 2: Prepare the Plaintext

- The plaintext "HELLO WORLD" needs to be split into digraphs (pairs of two letters).
- If there are two identical letters in a digraph (like "LL" in "HELLO"), we add an extra letter (commonly an "X") between them.
- Also, if the length of the plain text is odd, we add an extra letter at the end (e.g., "X").

Plaintext: "HELLO WORLD"

Modified Plaintext: "HE LX LO WO RL DX"

Step 3: Apply the Playfair Cipher Encryption Rules

- **Rule 1**: If the letters are in the same row, replace them with the letters to their immediate right (wrap around to the left side of the row if necessary).

- **Rule 2**: If the letters are in the same column, replace them with the letters immediately below them (wrap around to the top if necessary).
- **Rule 3**: If the letters form a rectangle, replace them with the letters on the same row but at the opposite corners of the rectangle.

Let's now go through the encryption of the digraphs one by one:

1. **HE**:
    - H is at (3,2) and E is at (1,1).
    - They form a rectangle. The letters at the opposite corners are **G** (row 3, column 1) and **X** (row 1, column 2).
    - **HE** = **GX**
2. **LX**:
    - L is at (2,1) and X is at (1,2).
    - They form a rectangle. The letters at the opposite corners are **E** (row 1, column 1) and **C** (row 2, column 2).
    - **LX** = **EC**
3. **LO**:
    - L is at (2,1) and O is at (4,1).
    - They are in the same column. The letter below L is **G**, and the letter below O is **Q**.
    - **LO** = **GQ**
4. **WO**:
    - W is at (5,3) and O is at (4,1).
    - They form a rectangle. The letters at the opposite corners are **V** (row 5, column 1) and **Q** (row 4, column 3).
    - **WO** = **VQ**

5. **RL**:
    - R is at (4,3) and L is at (2,1).
    - They form a rectangle. The letters at the opposite corners are **D** (row 2, column 3) and **F** (row 4, column 1).
    - **RL** = **DF**
6. **DX**:
    - D is at (2,4) and X is at (1,2).
    - They form a rectangle. The letters at the opposite corners are **A** (row 1, column 4) and **B** (row 2, column 2).
    - **DX** = **AB**

Ciphertext: GX EC GQ VQ DF AB

Decryption Process

To decrypt, we simply reverse the encryption process using the same key table and follow the reverse rules:

1. **GX**:
    - G is at (3,1) and X is at (1,2).
    - They form a rectangle. The letters at the opposite corners are **H** (row 3, column 2) and **E** (row 1, column 1).
    - **GX** = **HE**
2. **EC**:
    - E is at (1,1) and C is at (2,3).
    - They form a rectangle. The letters at the opposite corners are **L** (row 1, column 3) and **X** (row 2, column 1).

- **EC** = **LX**

3. **GQ**:
    - G is at (3,1) and Q is at (4,2).
    - They are in the same column. The letter above G is **L**, and the letter above Q is **O**.
    - **GQ** = **LO**
4. **VQ**:
    - V is at (5,1) and Q is at (4,2).
    - They form a rectangle. The letters at the opposite corners are **W** (row 5, column 2) and **O** (row 4, column 1).
    - **VQ** = **WO**
5. **DF**:
    - D is at (2,4) and F is at (2,5).
    - They are in the same row. The letter to the left of D is **R**, and the letter to the left of F is **L**.
    - **DF** = **RL**
6. **AB**:
    - A is at (1,4) and B is at (2,2).
    - They form a rectangle. The letters at the opposite corners are **D** (row 1, column 2) and **X** (row 2, column 4).
    - **AB** = **DX**

Decrypted Plaintext: HELLO WORLD

3. Development

3.1. Quadracrypt Cipher

Quadracrypt Cipher is the next generation encryption technique which can handle current cipher issues such as Playfair cipher. With the multi-layer security systems approach. Substitution, transposition, and binary, operations can together form a powerful security mechanism that uses the best features of each.

3.1.1. Research and Background Required for modifications

The Playfair Cipher converts digraphs, (pairs of letters), and substitutes them using a 6x5 matrix of letters. Digraph frequency patterns remain immediately apparent even when individual character frequency patterns are hidden. Furthermore, it needs to remove letters (mainly J) because of its 26-character limitation; accomplishing so might almost increase the difficulty of decryption. These problems are easily addressed when more information analyzing is included, such as random matrix mixing, columnar transposition, and binary encoding.

3.1.2. Encryption Process Using Mathematical Calculations

- Dynamic Key Matrix Generation

  Dynamically developed, more complex rules such as those that are based on hashing approaches or pre-shared secrets are applied to take place of a basic static 6x5 matrix that includes a term.

  The Way it Works:

  The plaintext is split (on a couple of letters, known as digraphs) into pairs of letters. A filler (e.g. B) is added if a letter repeats in a pair.

  A 6x5 key matrix is constructed, and the positions of the digraph letters in the matrix determine the encryption process:

  Rules

  - Letters in the same row are replaced by letters to their right.
  - Letters in the same column are replaced by the letters below them.
  - If the letters form a rectangle, in encryption replaced by anticlockwise corner and in decryption replaced by clockwise corner.
  - If the corner of the rectangle is the number, then swap with its upward alphabet.

1. **Key Preparation:**

Developing a matrix with dimensions 6x5 by choosing a keyword. The character 'J' is combined with 'I', and any unnecessary letters are removed.

Keyword: MUSIC

Key Matrix

M	U	S	I/J	C	A
B	D	E	F	G	H
K	L	N	O	P	Q
R	T	V	W	X	Y
Z	1	2	3	4	5

2. **Pairing the Plaintext:**

Divide the plain text into digraphs, that are composed of two-letter pairs. If the same word appears more than once, use a filler letter X.

Plaintext: **GUITAR**

3. **Divide into digraphs: GU, IT, AR**

Applying Playfair rules for encryption

- GU = CD
- IT = UW
- AR = MY

Ciphertext after applying Playfair cipher process is CDUWMY.

- Caesar Cipher (Shift by 5)

  Combining substitution and shifting techniques can be integrated as a modification to the Caesar Cipher to improve its security margin. Here I use a hybrid approach where just the Playfair Cipher is used to encrypt digraphs, and a Caesar Cipher shift can be used as a secondary layer of encryption. This modification complicates the work and gives more resistance to cryptanalysis.

The Way it Works

In a "Shift by 5" cipher, each alphabet letter is shifted five places forward, from letter 'Z' to letter 'E' (if one moves forward five places), the new position is updated (A = F, B =G, so on till Z=E).

The Way It Works:

Start with plain text or ciphertext.

- Encrypt: Shift each letter 5 positions forward for encryption.
- Decrypt: Shift each letter 5 positions backward for decryption.
- Non-Alphabetic Characters: Keep spaces, numbers, and symbols unchanged.

Alphabet	Shifted Alphabet	Alphabet Position
A	**F**	**0**
B	**G**	**1**
C	**H**	**2**
D	**I**	**3**
E	**J**	**4**
F	**K**	**5**
G	**L**	**6**
H	**M**	**7**
I	**N**	**8**
J	**O**	**9**
K	**P**	**10**
L	**Q**	**11**
M	**R**	**12**
N	**S**	**13**
O	**T**	**14**
P	**U**	**15**
Q	**V**	**16**
R	**W**	**17**

S	**X**	**18**
T	**Y**	**19**
U	**Z**	**20**
V	**A**	**21**
W	**B**	**22**
X	**C**	**23**
Y	**D**	**24**
Z	**E**	**25**

Table 3 Shift by 5 Cipher Mapping

Ciphertext for Caesar Cipher: CDUWMY

Now,

Shift each letter with the gap of 5.

- C = H
- D = I
- U = Z
- W = B
- M = R
- Y = D

Ciphertext after applying Caesar Cipher is HIZBRD.

- Left Shift

  It is a variation that can improve by modifying a new layer of complexity to reorganize it in the shape of the ciphertext. This method shifts each letter of the resulting message as in the Playfair Cipher and left in its string by a specific number of positions.

  The Way It Works:

  - Convert into a Grid: Organize the ciphertext into a systematic grid layout.
  - Determine Shift Direction: Execute a left shift, relocating each letter one position to the left in the grid.

- Manage Wrap Around: Characters at the beginning of the row circle back to the end of that row.
- Modified Ciphertext: The moved grid transforms into the new ciphertext.

HIZBRD = $\begin{matrix} H & I & Z \\ B & R & D \end{matrix}$

Now Shift the letter from left to right. Then the letters will shift left side by one step.

$\begin{matrix} H & I & Z \\ B & R & D \end{matrix} = \begin{matrix} I & Z & H \\ R & D & B \end{matrix}$

Ciphertext after applying the process of left shift is $\begin{matrix} I & Z & H \\ R & D & B \end{matrix}$.

- Diagonal Shift

  Another modification is diagonal shift method, and by inclusion of leading and trail spaces before ciphering and afterwards, the ciphertext security has been improved. Here it, improves security by shifting the text diagonally within a matrix or grid.

  In this process letters will change diagonally from top left side to button right side. If there will 3 rows, then it will also shift other letters from the second top letter to bottom right of third row.

  The Way It Works:

  - Start with the ciphertext organized in a grid layout.
  - Diagonal Transformation: Shift the letters diagonally, beginning from the upper-left corner to the lower-right corner of the grid.
  - Handling Multiple Rows: For grids that contain several rows, keep shifting letters diagonally, moving to the next open diagonal positions.
  - Revised Ciphertext: The restructured grid transforms into the revised ciphertext.

  Ciphertext for Diagonal Shift = $\begin{matrix} I & Z & H \\ R & D & B \end{matrix}$

  $\begin{matrix} I & Z & H \\ R & D & B \end{matrix} = \begin{matrix} D & B & H \\ R & I & Z \end{matrix}$

  Ciphertext after applying Diagonal Shift approaches the output is $\begin{matrix} D & B & H \\ R & I & Z \end{matrix}$.

The Output from the process is DBHRIZ.

3.1.3. Encryption Process Using Logical Calculation

- Convert Cipher Value into Binary

But how to get the Cipher Values that can be, in turn, converted into binary numbers? This step involves conversion from cipher values to binary where data must be encrypted. Dynamic mapping using a shared key or hash guarantee binary values changes after every session making the system more secure. If one makes the binary length 5-bit or 6-bit, it becomes possible to encode more characters. They improve flexibility and thus improve the encryption process as well. In this the value of both binary and Alphabet position will be same for each other.

The Way it Works:

- Assign each letter in the cipher the number of the position it holds in the alphabet.
- Succeed these positions when transformed into 5-bit binary numbers.
- XOR operations should also be compatible with the previous versions.
- Alphabet position and Decimal will be same after converting cipher value into Alphabet Position and insert Binary with the help of Binary.

Alphabet	Alphabet Position	Decimal
A	0	1
B	1	2
C	2	3
D	3	4
E	4	5
F	5	6
G	6	7
H	7	8
I	8	9
J	9	10
K	10	11

L	11	12
M	12	13
N	13	14
O	14	15
P	15	16
Q	16	17
R	17	18
S	18	19
T	19	20
U	20	21
V	21	22
W	22	23
X	23	24
Y	24	25
Z	25	26

Table 4 Letter, Decimal and Alphabetic Positions

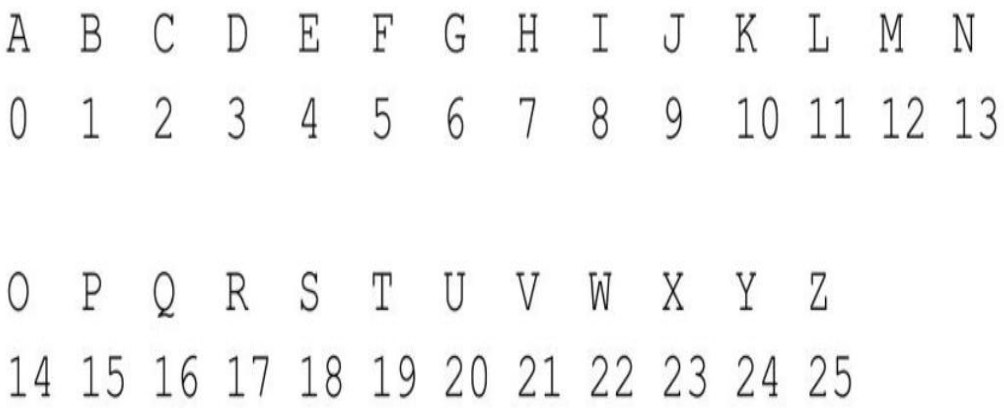


Figure 3 Cipher Value Position

Cipher Value: DBHRIZ

Cipher Value	Alphabet Position	Decimal	Binary
D	3	3	00011
B	1	1	00001
H	7	7	00111
R	17	17	10001
I	8	8	01000
Z	25	25	11001

Table 5 Cipher Value into Binary

- Binary XOR Calculation

  Binary XOR Calculation Binary XOR computational operations can be made more secure by using session keys of variable length derived from whatever is currently available during a particular session. One main idea in implementing the concept is that multiple keys, when used one after the other, pose difficulty in decryption. Other bit-level transformations such as shuffling before XOR introduce another layer of security into our system. These changes make for the XOR step to again be extremely unpredictable and efficient as explained earlier.

  This table shows how the XOR result is calculated using the cipher value, its binary equivalent, and the key. Each cipher value is converted to a 5-bit binary number, and the XOR operation is performed between this binary value and the 5-bit key. The XOR result is obtained by comparing corresponding bits, following the rule:

- 1 + 1 = 0
- 0 + 0 = 0
- 1 + 0 = 1
- 0 + 1 = 1

The Way it Works:

- The binary number values are calculating and the XOR operation to be done with the obtained key.
- Like special bitwise OR and AND, one needs to compare each bit and apply XOR rules.
- The second two-bit sequence should be produced by means of the new binary sequence as the result of the XOR operation.

XOR with Key

Key = 10011.

The following table shows the calculation of XOR Operations.

Cipher Value	Binary	Key	XOR Result
D	00011	10011	10000
B	00001	10011	10010
H	00111	10011	10100
R	10001	10011	00010
I	01000	10011	11011
Z	11001	10011	01010

Table 6 XOR Calculation Table

- Converting Binary to Decimal

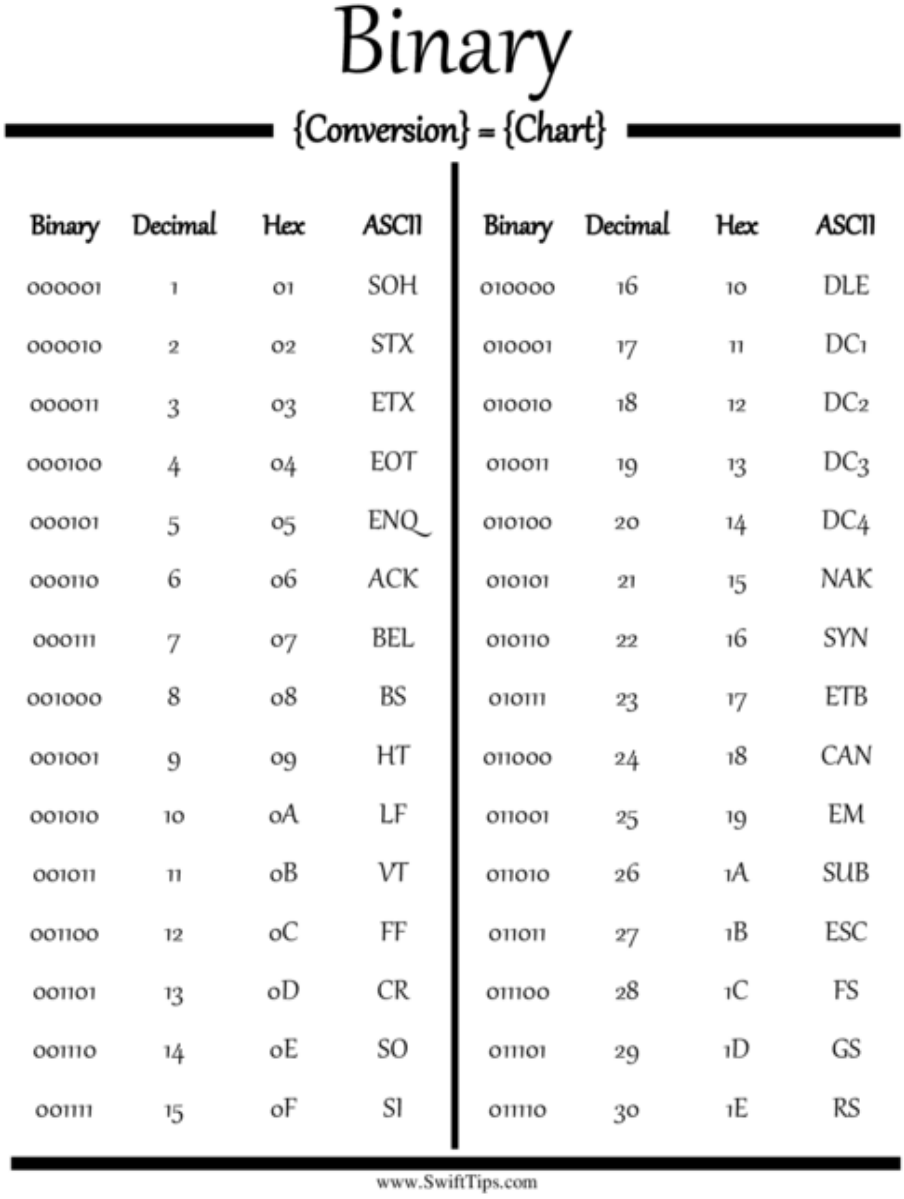
  Converting Binary to Decimal Base conversion of binary to decimal is best practiced by diversifying using number base other than 10 such as base 7 or 12. When transforming values before converting binaries, it becomes difficult to reverse-engineer when transformation rules are applied to binary values. These enhancements afford the attacker with no way of ascertaining the original sequence. This step becomes so crucial in enhancing the general process of encryption.

  The XOR result must be converted from binary to decimal as part of the entire process. Beginning with the rightmost bit, each binary value is multiplied by the matching power of 2. The decimal equivalent is then obtained by adding the results. The following figure will visually demonstrate the conversion process from binary to decimal for the XOR result.

  The Way it Works:

  - The XOR results are now changed into a decimal format.
  - On the other hand, each binary digit is multiplied by its respective power of two because of multiplication.
  - The products are added to give the ten's place referred to as the decimal equivalent of the binary number.

- Decimal values are in the middle between values and characters; hence they are used for mapping into character alphabets.
- For this, this process makes a connection between the final XOR result and the last mapping of characters.



Binary

{Conversion} = {Chart}

Binary	Decimal	Hex	ASCII	Binary	Decimal	Hex	ASCII
000001	1	01	SOH	010000	16	10	DLE
000010	2	02	STX	010001	17	11	DC1
000011	3	03	ETX	010010	18	12	DC2
000100	4	04	EOT	010011	19	13	DC3
000101	5	05	ENQ	010100	20	14	DC4
000110	6	06	ACK	010101	21	15	NAK
000111	7	07	BEL	010110	22	16	SYN
001000	8	08	BS	010111	23	17	ETB
001001	9	09	HT	011000	24	18	CAN
001010	10	0A	LF	011001	25	19	EM
001011	11	0B	VT	011010	26	1A	SUB
001100	12	0C	FF	011011	27	1B	ESC
001101	13	0D	CR	011100	28	1C	FS
001110	14	0E	SO	011101	29	1D	GS
001111	15	0F	SI	011110	30	1E	RS

www.SwiftTips.com

Figure 4 Binary to Decimal

The following figure shows the Conversion of Binary to Decimal and other (SwiftTips, 2024).

The following table shows the conversion from Binary to Decimal.

XOR Result	Decimal Value
10000	16
10010	18
10100	20
00010	2
11011	27
01010	10

Table 7 Binary to Decimal Conversion

This process gets the output 16,16,20,2,27 and 10 by converting binary into decimal.

- Decimal to Alphabet using Custom Mapping Table

  Decimal to Alphabet using a mapping table This step involves conversion of decimal values to characters in a manner that the table used can be created for each session if desired. Self-healing maps obtained from shared keys or hashes guarantee unpredictability of the various dynamic tables. They become widely applicable and more flexible where the extended character sets are included. These enhancements guarantee the originality of the mapping process as well as its flexibility.

  The process involves converting the decimal values into numeric alphabets using a custom mapping table. Each decimal value is matched to its corresponding alphabet or character based on the predefined mapping rules in the table.

  The Way it Works:

  - Decimal values are converted to characters by means of the predefined custom table of characters.
  - The mapping includes only letters a-z and A-Z as well as numerous special signs.
  - Every number is properly related to a character ensuring there is no distortion of figures.
  - It is the final process in which numerical values are converted into textual text.

Character	Number	Character	Number	Character	Number
a	0	s	18	K	36
b	1	t	19	L	37
c	2	u	20	M	38
d	3	v	21	N	39
e	4	w	22	O	40
f	5	x	23	P	41
g	6	y	24	Q	42
h	7	z	25	R	43
i	8	A	26	S	44
j	9	B	27	T	45
k	10	C	28	U	46
l	11	D	29	V	47
m	12	E	30	W	48
n	13	F	31	X	49
o	14	G	32	Y	50
p	15	H	33	Z	51
q	16	I	34	.	52
r	17	J	35	spasi	53
				,	54
				Enter	55

Figure 5 Number to Character

The table represents a custom character-to-number mapping used for encoding purposes. It assigns numbers to characters, starting with lowercase letters (a = 0 to z = 25), followed by uppercase letters (A = 26 to Z = 51). Special symbols like periods (. = 52), {spaces (spasi) = 53}, (commas = 54), and (Enter = 55) are also included. This mapping simplifies converting text into numeric form for cryptographic or computational use. It's an alternative to ASCII encoding (ResearchGate, 2024).

The following table shows the conversion from Decimal to Alphabet using Custom Mapping Table.

Decimal Value	Numeric Alphabet
16	q
18	s
20	u
2	c
27	B
10	k

Table 8 Decimal to Alphabet Conversion

The final output after all encryption process is **qsucBk**.

In addition, these modifications focus on the restrictions of the original cipher and make it appropriate for useful in being used to protect sensitive information in a modern environment.

3.1.4. Decryption Process

The decryption process systematically reverses each of the encryption steps, restoring the original plaintext by undoing the transformations applied during encryption. To Decrypt value **qsucBk** there are some steps to follow

Alphabet to Decimal using Custom Mapping Table

The process involves converting the numeric alphabets into Decimal Value using a custom mapping table. Each alphabet or character is matched to its corresponding Decimal value based on the predefined mapping rules in the table. The following table shows the conversion from Alphabet to Decimal using Custom Mapping Table.

Numeric Alphabet	Decimal Value
q	16
s	18
u	20
c	2
B	27
k	10

Table 6 Alphabet to Decimal Conversion

After the process of conversing from Alphabet to Decimal Value the value is: 16,18,20,2,27,10.

Now, we must convert decimal into binary by following step.

Converting Decimal to XOR Result (Binary)

The process involves reversing the encryption steps. The decimal value obtained during encryption must be converted back to binary to retrieve the original XOR result. This is achieved by dividing the decimal value by 2 repeatedly and recording the remainders, which represent the binary digits starting from the least significant bit (rightmost). Once the original binary value is reconstructed, it can be used for further decryption steps.

The following table shows the conversion from Binary to Decimal.

Decimal Value	XOR Result (Binary)
16	10000
18	10010
20	10100
20	00010
27	11011
10	01010

Table 7 Decimal to Binary Conversion

This process helps to get XOR Result (Binary) by conversion of Decimal value into Binary.

Now, next step is to get Binary by comparing key and XOR Result.

Binary XOR Calculation

This table shows how the XOR result is calculated for decryption using the cipher value, its binary equivalent, and the key. Each cipher value is converted to a 5-bit binary number, and the XOR operation is performed between this binary value and the 5-bit key. The XOR result is obtained by comparing corresponding bits, following the rule:

- 1 + 1 = 0
- 0 + 0 = 0
- 1 + 0 = 1

- 0 + 1 = 1

XOR with Key

Key = 10011

The following table shows the calculation of XOR Operations.

XOR Result	Key	XOR with Key	Decrypted Binary
10000	10011	10000 XOR 10011	00011
10010	10011	10010 XOR 10011	00001
10100	10011	10100 XOR 10011	00111
00010	10011	00010 XOR 10011	10001
11011	10011	11011 XOR 10011	01000
01010	10011	01010 XOR 10011	11001

Table 8 XOR Decryption Calculation Table

This step helps how the XOR operation is used to decrypt the cipher text. By applying the XOR operation between the binary representation of each cipher value and the given key, the Binary is recovered.

The next step is to convert the decrypted binary values back into their corresponding characters, ensuring the correct sequence and positioning in the plaintext message.

Convert Binary into Cipher Value.

This process involves converting the binary results from the XOR operation back into their corresponding Decimal then after in alphabet positions and then mapping them to the original cipher values. Each binary result, obtained from the XOR operation with the key, is converted back to its decimal, then after position in the alphabet, and then the character is recovered. Decimal and Alphabet Position will be same after converting Binary into Decimal Insert Cipher Value according to its Alphabet Position.

Binary	Decimal	Alphabet Position	Cipher Value
00011	3	3	D
00001	1	1	B
00111	7	7	H
10001	17	17	R
01000	8	8	I
11001	25	25	Z

Table 9 Binary into Cipher Value

This step helps to reverse the XOR operation by converting the decrypted binary values back into their corresponding alphabet positions. By doing so, we recover the original cipher text. The characters are mapped to their correct positions in the message, and we retrieve the original ciphertext which is DBHRIZ.

Diagonal Shift

To decrypt the ciphertext **"DBHRIZ"** using the diagonal process, first, arrange the characters of the ciphertext in a grid. For a 6-character string, create a 3x2 grid and fill it diagonally from top-left to bottom-right. Change it into GRID Structure: $\begin{matrix} D & B & H \\ R & I & Z \end{matrix}$

$$\begin{matrix} D & B & H \\ R & I & Z \end{matrix} = \begin{matrix} I & Z & H \\ R & D & B \end{matrix}$$

Output after diagonally change to ciphertext is $\begin{matrix} I & Z & H \\ R & D & B \end{matrix}$.

Step 6: Reverse Left-Shift

Ciphertext for Reverse Left Shift is $\begin{matrix} I & Z & H \\ R & D & B \end{matrix}$. In this process left shift method should done by performing reverse, which is right shift, successfully placing the letters again in their original sequence.

$$\begin{matrix} I & Z & H \\ R & D & B \end{matrix} = \begin{matrix} H & I & Z \\ B & R & D \end{matrix}$$

Decrypted Text after applying Reverse left-shift process is $\begin{matrix} H & I & Z \\ B & R & D \end{matrix}$.

Reverse Caesar Cipher (Shifting by -5)

In this process, -5 Caesar shift involves shifting each ciphertext letter five alphabetical places back. A character wraps around to 'Z' if it passes 'A'. As an example, "F" turns into "A," "N" turns into "I," and so on.

Change $\begin{matrix} H & I & Z \\ B & R & D \end{matrix}$ into one row which will be HIZBRD.

Now, reverse each ciphertext letter by -5.

H= C

I = D

Z = U

B = W

R = M

D = Y

After shifting ciphertext by -5 the output will be CDUWMY.

Decryption process by using Reverse Playfair cipher.

Rule

- In this process, the decryption will work by shifting each letter clockwise.

Ciphertext: CDUWMY

Make it into Digraphs.

CD = GU

UW = IT

MY = AR

After the process the final output of decryption process is GUITAR.

3.2. Encryption Algorithm

The following steps shows the algorithm process of Quadracrypt Cipher:

3.2.1. Algorithm for Encryption Process

Step 1: Develop a 6x5 key matrix using a keyword.

Step 2: Divide the plaintext into digraphs (pairs of letters).

Step 3: Apply Playfair Cipher rules on each digraph.

Step 4: Apply Caesar Cipher with a shift of 5 to the Playfair Cipher output.

Step 5: Organize the resulting ciphertext into a grid layout.

Step 6: Perform Left Shift operation on the grid.

Step 7: Apply Diagonal Shift on the grid.

Step 8: Convert the ciphertext from the grid back to a string.

Step 9: Convert each letter in the ciphertext to its alphabet position.

Step 10: Convert the alphabet positions to 5-bit binary values.

Step 11: Perform XOR operation with the session key on the binary values.

Step 12: Convert the XOR result from binary to decimal.

Step 13: Convert the decimal values into alphabet characters using a custom mapping table.

3.2.2. Algorithm for Decryption Process

Step 1: Convert the ciphertext from the custom mapping table back to decimal values.

Step 2: Convert the decimal values to binary.

Step 3: Perform XOR operation with the session key on the binary values to reverse the encryption.

Step 4: Convert the binary result back to decimal.

Step 5: Convert the decimal values into alphabet positions.

Step 6: Convert the alphabet positions back into letters.

Step 7: Reverse the Diagonal Shift applied during encryption.

Step 8: Reverse the Left Shift operation applied during encryption.

Step 9: Apply the reverse Caesar Cipher (Shift by -5).

Step 10: Apply the reverse Playfair Cipher rules to the decrypted digraphs.

Step 11: Combine the decrypted pairs to recover the original plaintext.

3.3. Elaborate Newly created Quadracrypt Cipher

3.3.1. Elaborating Encryption Process

1. Necessity of Modifications

These changes were made to bring change to make the process more secure and complex in the encryption process. Playfair and Caesar, for example, offer a medium density protection strategy but have severe cryptography assaults today. By combining multiple encryption techniques and introducing unique transformations (e.g., Left Shift, Diagonal Shift, and XOR operations), the algorithm achieves:

- Increased Complexity: Superimposing gotten extra layers that make it difficult for today's information deciphering systems to decode the ciphertext without the key.
- Enhanced Robustness: By combining all types of randomizing, frequency analysis and pattern recognition attacking methods fail to penetrate through the algorithm.
- Flexibility and Customization: The use of a custom-mapping table and unique transformations enables such an algorithm to be fine-tuned to a particular application.
- Modern Relevance: In the present study, binary-based XOR operations are incorporated in line with the conventional techniques.

2. Proposed Methodology

The new methodology combines a typical cipher with the best features of modern math used in quantum computing. It is a multi-layered process consisting of the following stages:

- Playfair Cipher Encryption: 6x5 matrix is used to create digraph-based encryption, for that reason. Improvements including anticlockwise as well as clockwise corner replacements make the process more complicated.

- Caesar Cipher with Shift by 5

  An additional layer of substitution is practiced by shifting each letter of the intermediate ciphertext five places towards latter part of the alphabet.

- Left-Shift

  The text obtained from the letters is written in the shape of a grid; each row receives a left positional shift; and characters within the row are wrapped. It does this in a way that increases positional complexity.

- Diagonal Shift

  Images of major characters are rearranged diagonally across some rows in grid-like fashion, and this makes a new type of encryption.

- Binary Conversion

  The ciphertext is then processed into binary form and these would be the form suitable for logical manipulation before performing the necessary Boolean operations of the system; the characters in the ciphertext are then encoded in the form of its position in the alphabet with the help of 5 bits for each character.

- Binary XOR Operations

  Binary representations are then XORed with a certain key to make the message that is to be delivered even more inconspicuous and add the layer of computational encryption.

- Decimal Conversion

  For XOR results obtained as values of several bits, XOR results are transformed from binary to decimal format turning the values seen in the XOR operation as numbers.

- Custom Mapping Conversion

  The decimal values are converted to alphabet letters by looking at a translator table that has been created and the end executor of the layers is the final ciphertext.

The new Encryption Algorithm

Step 1: Playfair Cipher Encryption

- Create a 6 by 5 matrix with a focus on a given keyword.
- The plaintext was divided into digraphs to which fillers were added where necessary in the encoding process.
- Use rules for the digraph encryption which has been reviewed.

Step 2: Caesar Cipher Encryption

- Rotate each letter of the Playfear ciphertext by five places around the alphabet.
- By this, non-alphabetic characters must be left as they are.

Step 3: Left-Shift Transformation

- Arrange the results obtained from the Caesar output by developing a grid.
- Then shift column by 1 step towards left.

Step 4: Diagonal Shift Transformation

- Transverse the letters in the positions of the diagonal in the grid.
- The diagonal shift is done from top left to bottom right.

Step 5: Binary Conversion

- Represent each character in a binary system, with each character using five bits of information; each bit reflecting the position of the character in the alphabet chain.

Step 6: XOR Operation

- Perform the XOR operation between each of the binary value with a previously agreed upon binary key.
- Following a few XOR rules compute the transformed binary sequence.

Step 7: Binary-to-Decimal Conversion

- Now be converting the XOR result to decimal using binary powers of summation method.

Step 8: Decimal-to-Character Mapping

- These do with mapping of decimal values to characters through a custom table.
- Add characters one or more alphabet to the previous generated word to form the final ciphertext

3.3.2. Elaborating Decryption Process

The new decryption algorithm

Step 1: Alphabet to Decimal using Custom Mapping Table

- Convert the alphabetic characters of the ciphertext into their corresponding decimal values based on the custom mapping table.

Step 2: Converting Decimal to XOR Result (Binary)

- Convert the decimal values into their corresponding binary format.

Step 3: Binary XOR Calculation

- Perform XOR operation between the binary result and the predefined key.

Step 4: Convert Binary into Cipher Value

- Convert the XOR results from binary back into decimal and then map them to alphabet positions to get the decrypted characters.

Step 5: Diagonal Shift

- Arrange the ciphertext characters in a grid and apply the diagonal shift process.

Step 6: Reverse Left-Shift

- Perform the reverse of the left-shift (right shift) to return to the original sequence of characters.

Step 7: Reverse Caesar Cipher (Shifting by -5)

- Shift each letter of the ciphertext five positions back in the alphabet.

Step 8: Reverse Playfair Cipher

- Use the reverse Playfair Cipher decryption rules by shifting each letter clockwise and splitting the ciphertext into digraphs.

3.4. Flowchart

3.4.1. Flowchart of Encryption.

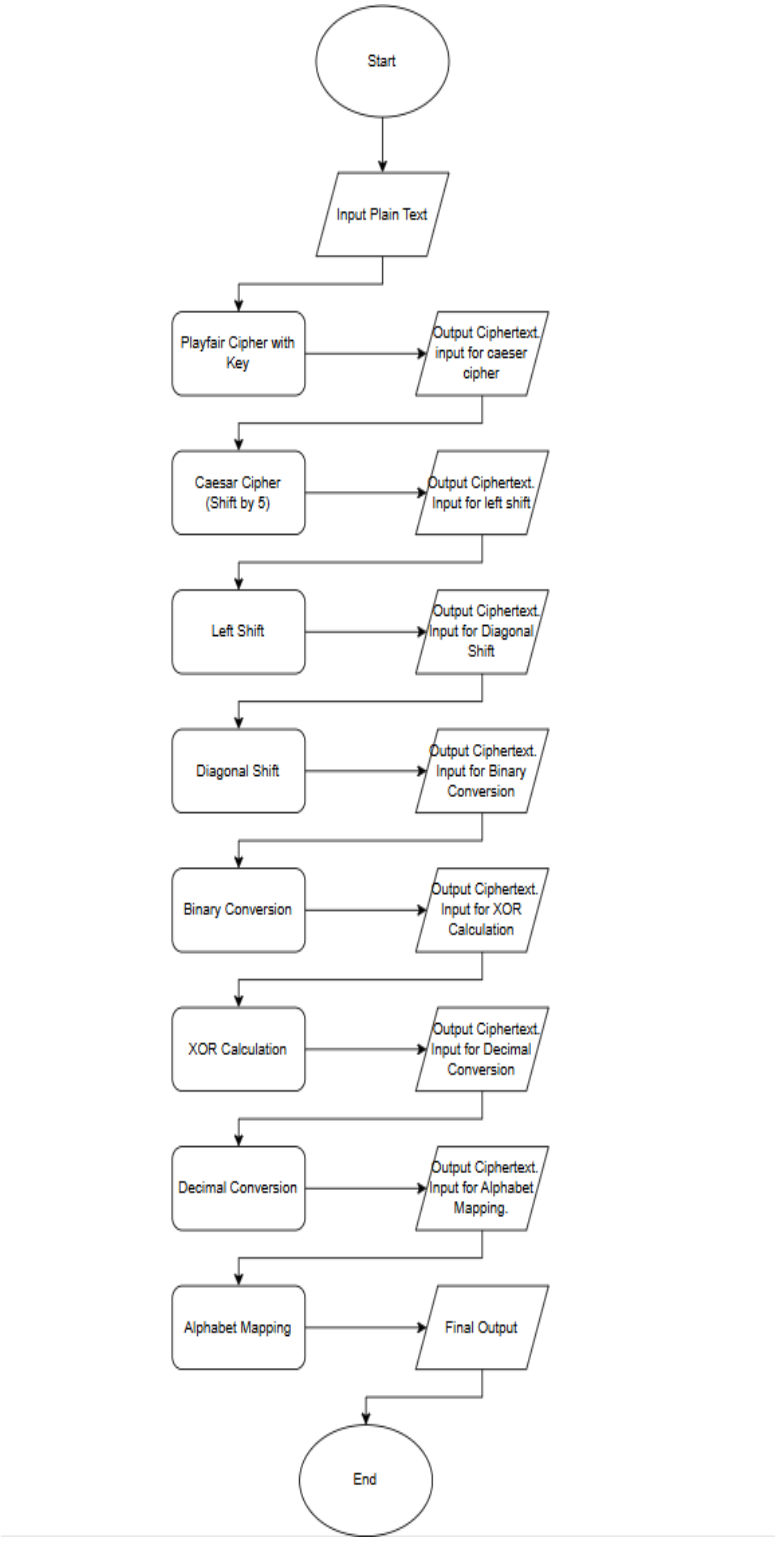


Figure 6 Flowchart of Encryption

3.4.2. Flowchart of Decryption

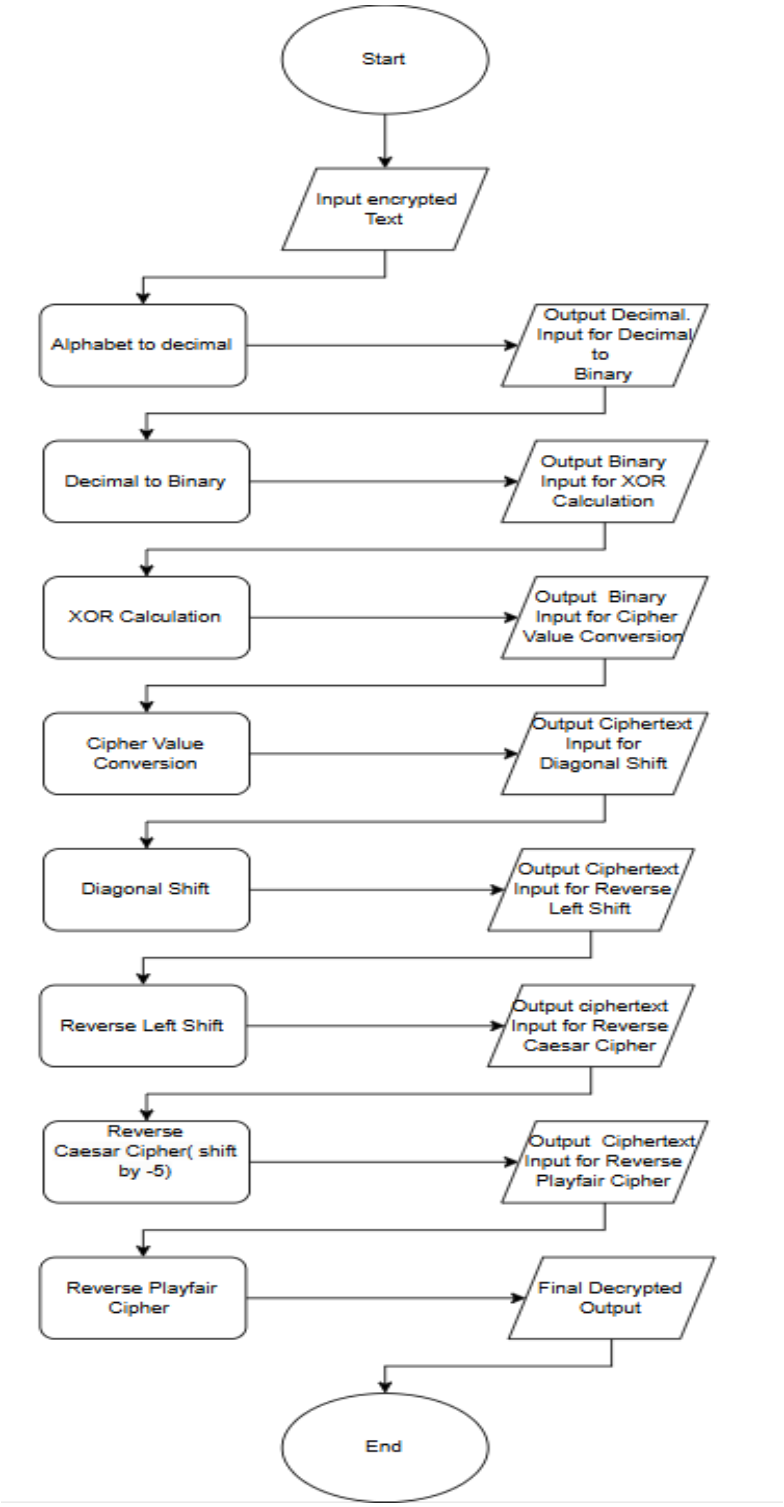


Figure 7 Figure of Decryption

4. Testing

4.1. Test 1

1. Key Preparation:

   Developing a matrix with dimensions 6x5 by choosing a keyword. The character 'J' is combined with 'I', and any unnecessary letters are removed.

   Keyword: IPHONE

   Key Matrix

I/J	P	H	O	N	E
A	B	C	D	F	G
K	L	M	Q	R	S
T	U	V	W	X	Y
Z	1	2	3	4	5

   Encryption Process

   Pairing the Plaintext:

   Plaintext: COPY

   Divide into Digraphs, that are composed of two pairs. If the word appears more than once and if it is not pairing, then use a letter X.

   Applying Playfair rules for encryption

- For **CO**:
  - Locate **C** and **O** in the matrix. They form a rectangle.
  - Replace each with the character according to anticlockwise direction: **DH**
- For **PY**:
  - Locate **P** and **Y** in the matrix. They form a rectangle.
  - Replace each with the character according to anticlockwise direction: **UE**

   After applying the rule, Intermediate Ciphertext: DHUE

2. Shift by 5

   Shift the ciphertext by 5 with help of the *Figure 3 Cipher Value Position.*

   D = I

   H = M

   U = Z

   E = J

   Ciphertext after shift: IMZJ

3. Left Shift

   Ciphertext = IMZJ

   In first, convert the ciphertext into grid.

$$\begin{matrix} I & M \\ Z & J \end{matrix}$$

   Now, shift first row in last and then last row will shift forward by one step.

   Ciphertext after left shift encryption MIJZ

4. Diagonal Shift

   Ciphertext = MIJZ

   To diagonal shift convert cipher text into grid: $\begin{matrix} M & I \\ J & Z \end{matrix}$

$$\begin{matrix} M & I \\ J & Z \end{matrix} = \begin{matrix} Z & I \\ J & M \end{matrix}$$

   After applying the diagonal shift the output is ZIJM.

5. Convert Cipher Value into Binary.

   Ciphertext = ZIJM. Convert it with the help of *Figure 3 Cipher Value* Position and *Figure 4 Binary to Decimal.*

Cipher Value	Alphabet Position	Decimal	Binary
Z	25	25	11001
I	8	8	01000
J	9	8	01001
M	12	12	01100

Table 10 Testing 1 convert Cipher value into Binary

Binary: 11001, 01000, 01001, 01100

6. Binary XOR Calculation.

   XOR with Key. Key = 10001. The following table shows the calculation of XOR Operations.

   Binary: 11001, 01000, 01001, 01100

Binary	Key	Binary XOR Key	XOR Result
11001	10001	11001 XOR 10001	01000
01000	10001	01000 XOR 10001	11001
01001	10001	01001 XOR 10001	11000
01100	10001	01100 XOR 10001	11101

Table 11 Testing 1 XOR Calculation Table

XOR Result in Binary: 01000, 11001, 11000, 11101

7. Converting Binary to decimal

   Convert XOR Result to Decimal by the help of the *Figure 4 Binary to Decimal*.

   XOR Result (Binary): 01000, 11001, 11000, 11101

XOR Result	Decimal
01000	8
11001	25
11000	24
11101	12

Table 12 Testing 1 Converting Binary to Decimal

The output after conversion of binary to decimal is 8, 25, 24 and 12.

8. Decimal to Alphabet using Custom Mapping Table

Now, convert Decimal into Alphabet using custom Mapping Table.

With the help of *Figure 5 Number to Character.* Decimal: 8, 25, 24, 12

Decimal Value	Numeric Alphabet
8	i
25	z
24	y
12	D

Table 13 Testing 1 Conversion of Decimal into Alphabet

The Final Ciphertext after encryption process is **izyD**

Decryption Process

1. Alphabet to Decimal Using Custom Mapping Table

The following table shows the conversion from Alphabet to Decimal Using Custom Mapping Table.

With the help of the *Figure 5 Number to Character* convert Alphabet to Decimal

Ciphertext = izyD

Numeric Alphabet	Decimal Value
i	8
z	25
y	24
D	12

Table 14 Testing 1 Conversion of Alphabet into Decimal

Decimal Value is 8, 25, 24, 12.

2. Converting Decimal Value to Binary.

   With the help *of Figure 4 Binary to Decimal* Convert Decimal Value into Binary (XOR Result).

   Decimal Value: 8, 25, 24, 12.

Decimal Value	Binary (XOR Result)
8	01000
25	11001
24	11000
12	11101

Table 15 Testing 1 Convert Decimal to Binary

   Binary (XOR Result): 01000, 11001, 11000, 11101

3. Binary XOR Calculation

   Now, with the help of XOR Result and key using XOR Operation decrypt the Binary Value.

   Key = 10001

   XOR Result: 01000, 11001, 11000, 11101

XOR Result	Key	XOR with Key	Decrypted Binary
01000	10001	01000 XOR 10001	11001
11001	10001	11001 XOR 10001	01000
11000	10001	11000 XOR 10001	01001
11101	10001	11101 XOR 10001	01100

Table 16 Testing 1 Binary XOR Calculation

   By using XOR operation Binary is recovered. Decrypted Binary: 11001, 01000, 01001, 01100

4. Convert Decrypted Binary into Cipher Value

   Converting Decrypted Binary into Cipher Value by the help of its position. With the help of *Figure 4 Binary to Decimal* convert binary into Decimal and with the help of *Table 4 Letter, Decimal and Alphabetic Positions* convert Decimal into Alphabet position and then after in Letter.

   Binary: 11001, 01000, 01001, 01100

Binary	Decimal	Alphabet Position	Cipher Value
11001	25	25	Z
01000	8	8	I
01001	9	9	J
01001	12	12	M

Table 17 Testing 1 Conversion of Binary to Cipher Value

Cipher Value after the conversion is ZIJM.

5. Diagonal Shift

   Ciphertext = ZIJM

   Convert it into Grid = $\begin{matrix} Z & I \\ J & M \end{matrix}$

   Shift it from top left to bottom right = $\begin{matrix} M & I \\ J & Z \end{matrix}$

   Ciphertext = MIJZ

6. Reverse Left Shift

   Ciphertext = MIJZ

   Convert it into Grid = $\begin{matrix} M & I \\ J & Z \end{matrix}$

   Shift the column by one step towards right

   $\begin{matrix} M & I \\ J & Z \end{matrix} = \begin{matrix} I & M \\ Z & J \end{matrix}$

   Ciphertext = IMZJ

7. Shift by -5

   With the help of Table 2 Shift letters by -5

   Ciphertext = IMZJ

   I = D

M = H

Z = U

J = E

Ciphertext = DHUE

8. Reverse Playfair Cipher

Key Matrix

I/J	P	H	O	N	E
A	B	C	D	F	G
K	L	M	Q	R	S
T	U	V	W	X	Y
Z	1	2	3	4	5

Ciphertext = DHUE

Make it into Digraphs.

DH UE

- **For DH**:
  - Locate **D** and **H** in the matrix. They form a rectangle.
  - Replace each with the character according to clockwise direction: **C** and **O**.
- **For UE**:
  - Locate **U** and **E** in the matrix. They form a rectangle.
  - Replace each with the character according to clockwise direction: **P** and **Y**.

Final Ciphertext after Decryption Process = **COPY**

4.2. Test 2

Key Preparation:

Developing a matrix with dimensions 6x5 by choosing a keyword. The character 'J' is combined with 'I', and any unnecessary letters are removed.

Keyword: UNIVERSITY

Key Matrix

U	N	I/J	V	E	R
S	T	Y	A	B	C
D	F	G	H	K	L
M	O	P	Q	W	X
Z	1	2	3	4	5

Encryption Process

1. Pairing the Plaintext:

   Plaintext: SECURITY

   Divide into Digraphs, that are composed of two pairs. If the word appears more than once and if it is not pairing, then use a letter X.

   SE CU RI TY

   Applying Playfair rules for encryption

   - For **SE**:
   - Locate **S** and **E** in the matrix. They form a rectangle.
   - Replace each with the character according to anticlockwise direction: **BU**
   - For **CU**:
   - Locate **C** and **U** in the matrix. They form a rectangle.
   - Replace each with the character according to anticlockwise direction: **RS**
   - For **RI**:
   - Locate **R** and **I** in the matrix. They form a rectangle.
   - Replace each with the character according to the rule (Right shift): **UV**
   - For **TY**:
   - Locate **T** and **Y** in the matrix. They form a rectangle.
   - Replace each with the character according to the rule (Right shift): **YA**

   After applying the rule, Intermediate Ciphertext: **BURSUVYA**

2. Shift by 5

   Ciphertext: **BURSUVYA**

   With the help of **Table 2** Shift all letters by 5.

- B = G
- U = Z
- R = W
- S = X
- U = Z
- V = A
- Y = D
- A = F

After shifting by 5, Ciphertext is **GZWXZADF**

3. Left Shift

Ciphertext: **GZWXZADF**

Convert ciphertext into grid.

$$\begin{matrix} G & Z & W & X \\ Z & A & D & F \end{matrix}$$

Now, shift all column by 1 step towards left.

$$\begin{matrix} Z & W & X & G \\ A & D & F & Z \end{matrix}$$

After left shift process, ciphertext is **ZWXGADFG.**

4. Diagonal Shift

Ciphertext: **ZWXGADFG**

Convert ciphertext into grid.

$$\begin{matrix} Z & W & X & G \\ A & D & F & Z \end{matrix}$$

Now, shift letter from top left to bottom right.

$$\begin{matrix} D & F & Z & G \\ A & Z & W & X \end{matrix}$$

After diagonal shift, ciphertext is **DFZGAZWX.**

5. Convert Cipher Value into Binary

   Ciphertext = **DFZGAZWX**

   Convert it into Binary with help of *Table 4 Letter, Decimal and Alphabetic Positions* and *Figure 4 Binary to Decimal*

Cipher Value	Alphabet Position	Decimal	Binary
D	3	3	00011
F	5	5	00101
Z	25	25	11001
G	6	6	00110
A	0	0	00000
Z	25	25	11001
W	22	22	10110
X	23	23	10111

Table 18 Testing 2 Convert Cipher Value into Binary

After converting cipher value into Binary, Binary: 00011, 00101, 11001, 00110, 00000, 11001, 10110, 10111

6. Binary XOR Calculation.

   XOR with Key.

   Key= 10001.

   The following table shows the calculation of XOR Operations.

   Binary: 00011, 00101, 11001, 00110, 00000, 11001, 10110, 10111

Binary	Key	Binary XOR Key	XOR Result
00011	10001	00011 XOR 10001	10010
00101	10001	00101 XOR 10001	10100
11001	10001	11001 XOR 10001	01000
00110	10001	00110 XOR 10001	10111
00000	10001	00000 XOR 10001	10001

11001	10001	11001 XOR 10001	01000
10110	10001	10110 XOR 10001	00111
10111	10001	10111 XOR 10001	00110

Table 19 Testing 2 XOR Calculation Table

XOR Result in Binary: 10010, 10100, 01000, 10111, 10001, 01000, 00111, 00110

7. Converting Binary to decimal

Convert XOR Result to Decimal by the help of the *Figure 4 Binary to Decimal*.

XOR Result	Decimal
10010	18
10100	20
01000	8
10111	23
10001	17
01000	8
00111	7
00110	6

Table 20 Testing 2 Converting Binary to Decimal

The output after conversion of binary to decimal is 18, 20, 8, 23, 17, 8, 7, 6

8. Decimal to Alphabet using Custom Mapping Table

Now, convert Decimal into Alphabet using custom Mapping Table. With the help of *Figure 5 Number to Character* convert decimal into alphabet.

Decimal Value: 18, 20, 8, 23, 17, 8, 7, 6

Decimal Value	Numeric Alphabet
18	s
20	u
8	i
23	x

17	r
8	i
7	h
6	g

Table 21 Testing 2 Conversion of Decimal into Alphabet

The Final Ciphertext after Encryption process is **suixrihg**

Decryption Process

1. Alphabet to Decimal Using Custom Mapping Table

   The following table shows the conversion from Alphabet to Decimal Using Custom Mapping Table. Insert ciphertext into Numeric Alphabet column. With the help of *Figure* 5 Number to Character convert decimal into alphabet.

   Ciphertext = **suixrihg**

Numeric Alphabet	Decimal Value
s	18
u	20
i	8
x	23
r	17
i	8
h	7
g	6

Table 22 Testing 2 Conversion of Alphabet into Decimal

Decimal Value is 18, 20, 8, 23. 17, 8, 7, 6

2. Converting Decimal Value to Binary.

   Converting Decimal Value into Binary, in which binary is XOR Result. Convert it with the help of *Figure 4 Binary to Decimal*.

   Decimal Value: 18, 20, 8, 23. 17, 8, 7, 6

Decimal Value	Binary (XOR Result)
18	10010
20	10100
8	01000
23	10111
17	10001
8	01000
7	00111
6	00110

Table 23 Testing 2 Convert Decimal to Binary

Binary (XOR Result): 10010, 10100, 01000, 10111, 10001, 01000, 00111, 00110

3. Binary XOR Calculation

Now, with the help of XOR Result and key using XOR Operation decrypt the Binary Value.

Key = 10001

XOR Result: 10010, 10100, 01000, 10111, 10001, 01000, 00111, 00110

XOR Result	Key	XOR with Key	Decrypted Binary
10010	10001	10010 XOR 10001	00011
10100	10001	10100 XOR 10001	00101
01000	10001	01000 XOR 10001	11001
10111	10001	10111 XOR 10001	00110
10001	10001	10001 XOR 10001	00000
01000	10001	01000 XOR 10001	11001
00111	10001	00111 XOR 10001	10110
00110	10001	00110 XOR 10001	10111

Table 24 Testing 2 Binary XOR Calculation

By using XOR operation Binary is recovered. Decrypted Binary: 00011, 00101, 11001, 00110, 00000, 11001, 10110, 10111

4. Convert Decrypted Binary into Cipher Value

   Converting Decrypted Binary into Cipher Value by the help of its position. With the help of *Figure 4 Binary to Decimal* convert binary into Decimal and with the help of *Table 4 Letter, Decimal and Alphabetic Positions* convert Decimal into Alphabet position and then after in Letter.

   Binary: 00011, 00101, 11001, 00110, 00000, 11001, 10110, 10111

Binary	Decimal	Alphabet Position	Cipher Value
00110	3	3	D
00101	5	5	F
11001	25	25	Z
00110	6	6	G
00000	0	0	A
11001	25	25	Z
10110	22	22	W
10111	23	23	X

Table 25 Testing 2 Conversion of Binary to Cipher Value

   Cipher Value after the conversion is **DFZGAZWX.**

5. Diagonal Shift

   Ciphertext = **DFZGAZWX**

   Convert it into Grid = $\begin{matrix} D & F & Z & G \\ A & Z & W & X \end{matrix}$

   Shift it from top left to bottom right = $\begin{matrix} Z & W & X & G \\ A & D & F & Z \end{matrix}$

   Ciphertext = **ZWXGADFZ**

6. Reverse Left Shift

   Ciphertext = **ZWXGADFZ**

   Convert it into Grid = $\begin{matrix} Z & W & X & G \\ A & D & F & Z \end{matrix}$

   Shift the column by one step towards right

$$\begin{matrix} Z & W & X & G \\ A & D & F & Z \end{matrix} = \begin{matrix} G & Z & W & X \\ Z & A & D & F \end{matrix}$$

Ciphertext = **GZWXZADF**

7. Shift by -5

With help of Table 2 shift all letters by -5.

Ciphertext = **GZWXZADF**

G = B

Z = U

W = R

X = S

Z = U

A = V

D = Y

F = A

Ciphertext = **BURSUVYA**

8. Reverse Playfair Cipher

Ciphertext = **BURSUVYA**

Key Matrix

U	N	I/J	V	E	R
S	T	Y	A	B	C
D	F	G	H	K	L
M	O	P	Q	W	X
Z	1	2	3	4	5

Make it into Digraphs.

BU RS UV YA

- **For BU**:
  - Locate **B** and **U** in the matrix. They form a rectangle.
  - Replace each with the character according to clockwise direction: **S** and **E**.
- **For RS**:
  - Locate **R** and **S** in the matrix. They form a rectangle.
  - Replace each with the character according to clockwise direction: **C** and **U**.
- **For UV**:
  - Locate **U** and **UV** in the matrix. They form a rectangle.
  - Replace each with the character by reverse right shift rule: **R** and **I**.
- **For YA**:
  - Locate **Y** and **A** in the matrix. They form a rectangle.
  - Replace each with the character by reverse right shift rule: **T** and **Y**.

Final Ciphertext = **SECURITY**

4.3. Test 3

Key Preparation:

Developing a matrix with dimensions 6x5 by choosing a keyword. The character 'J' is combined with 'I', and any unnecessary letters are removed.

Keyword: UNIVERSITY

Key Matrix

U	N	I/J	V	E	R
S	T	Y	A	B	C
D	F	G	H	K	L
M	O	P	Q	W	X
Z	1	2	3	4	5

Encryption Process

1. Pairing the Plaintext:

Plaintext: ATTACK

Divide into Digraphs, that are composed of two pairs. If the word appears more than once and if it is not pairing, then use a letter X.

AT TA CK

Applying Playfair rules for encryption

- For **AT**:
  - Locate **A** and **T** in the matrix. They form a rectangle.
  - Replace each with the character according to Right replace rule: **BY**
- For **TA**:
  - Locate **C** and **U** in the matrix. They form a rectangle.
  - Replace each with the character according to Right replace rule: **YB**
- For **CK**:
  - Locate **C** and **K** in the matrix. They form a rectangle.
  - Replace each with the character according to anticlockwise rule: **BL**

After applying the rule, Intermediate Ciphertext: **BYYBBL**

2. Shift by 5

   Ciphertext: **BYYBBL**

   With the help of Table 2 Shift all letters by 5.

   - B = G
   - Y = D
   - Y = D
   - B = G
   - B = G
   - L = Q

   After shifting by 5, Ciphertext is **GDDGGQ**

3. Left Shift

   Ciphertext: **GDDGGQ**

Convert ciphertext into grid.

$G \quad D \quad D$
$G \quad G \quad Q$

Now, shift all column by 1 step towards left.

$D \quad D \quad G$
$G \quad Q \quad G$

After left shift process, ciphertext is **DDGGQG.**

4. Diagonal Shift

Ciphertext: **DDGGQG.** Convert ciphertext into grid.

$D \quad D \quad G$
$G \quad Q \quad G$

Now, shift letter from top left to bottom right.

$Q \quad G \quad G$
$G \quad D \quad D$

After diagonal shift, ciphertext is **QGGGDD.**

5. Convert Cipher Value into Binary

Convert it into Binary with help of *Table 4 Letter, Decimal and Alphabetic Positions* and *Figure 4 Binary to Decimal.* Ciphertext = **QGGGDD.**

Cipher Value	Alphabet Position	Decimal	Binary
Q	16	16	10000
G	6	6	00110
G	6	6	00110
G	6	6	00110
D	3	3	00011
D	3	3	00011

Table 26 Testing 3 Convert Cipher Value into Binary

After converting cipher value into Binary, Binary: 10000, 00110, 00110, 00110, 00011, 00011

6. Binary XOR Calculation.

   XOR with Key. Key= 10001.

   The following table shows the calculation of XOR Operations. Binary: 10000, 00110, 00110, 00110, 00011, 00011

Binary	Key	Binary XOR Key	XOR Result
10000	10001	10000 XOR 10001	00001
00110	10001	00110 XOR 10001	10111
00110	10001	00110 XOR 10001	10111
00110	10001	00110 XOR 10001	10111
00011	10001	00011 XOR 10001	10010
00011	10001	00011 XOR 10001	10010

Table 18 Testing 3 XOR Calculation Table

   XOR Result in Binary: 00001, 10111, 10111, 10111, 10010, 10010

7. Converting Binary to decimal

   Convert XOR Result to Decimal by the help of the *Figure 4 Binary to Decimal.*

XOR Result	Decimal
00001	1
10111	23
10111	23
10111	23
10010	18
10010	18

Table 19 Testing 3 Converting Binary to Decimal

   The output after conversion of binary to decimal is 1, 23, 23, 23, 18, 18

8. Decimal to Alphabet using Custom Mapping Table

   Now, convert Decimal into Alphabet using custom Mapping Table. With the help of *Figure 5 Number to Character* convert decimal into alphabet.

   Decimal Value: 1, 23, 23, 23, 18, 18

Decimal Value	Numeric Alphabet
1	b
23	x
23	x
23	x
18	s
18	s

Table 27 Testing 3 Conversion of Decimal into Alphabet

The Final Ciphertext after the encryption process is **bxxxss**

Decryption Process

1. Alphabet to Decimal Using Custom Mapping Table

   The following table shows the conversion from Alphabet to Decimal Using Custom Mapping Table. Insert ciphertext into Numeric Alphabet column. With the help of *Figure 5 Number to Character* convert decimal into alphabet.

   Ciphertext = **bxxxss**

Numeric Alphabet	Decimal Value
b	1
x	23
x	23
x	23
s	18
s	18

Table 21 Testing 3 Conversion of Alphabet into Decimal

Decimal Value is 1, 23, 23, 23,18, 18

2. Converting Decimal Value to Binary.

   Converting Decimal Value into Binary, in which binary is XOR Result. Convert it with the help of *Figure 4 Binary to Decimal*.

   Decimal Value: 1, 23, 23, 23,18, 18

Decimal Value	Binary (XOR Result)
1	00001
23	10111
23	10111
23	10111
18	10010
18	10010

Table 22 Testing 3 Convert Decimal to Binary

   Binary (XOR Result): 00001, 10111, 10111, 10111, 10010, 10010

3. Binary XOR Calculation

   Now, with the help of XOR Result and key using XOR Operation decrypt the Binary Value.

   Key = 10001. XOR Result: 00001, 10111, 10111, 10111, 10010, 10010

XOR Result	Key	XOR with Key	Decrypted Binary
00001	10001	00001 XOR 10001	10000
10111	10001	10111 XOR 10001	00110
10111	10001	10111 XOR 10001	00110
10111	10001	10111 XOR 10001	00110
10010	10001	10010 XOR 10001	00011
10010	10001	10010 XOR 10001	00011

Table 23 Testing 3 Binary XOR Calculation

   By using XOR operation Binary is recovered. Decrypted Binary: 10000, 00110, 00110, 00110, 00011, 00011

4. Convert Decrypted Binary into Cipher Value

Converting Decrypted Binary into Cipher Value by the help of its position. With the help of *Figure 4 Binary to Decimal* and with the help of *Table 4 Letter, Decimal and Alphabetic Positions* Convert it into decimal-alphabet position-letter

Binary: 10000, 00110, 00110, 00110, 00011, 00011

Binary	Decimal	Alphabet Position	Cipher Value
10000	16	16	Q
00110	6	6	G
00110	6	6	G
00110	6	6	G
00011	3	3	D
00011	3	3	D

Table 24 Testing 3 Conversion of Binary to Cipher Value

Cipher Value after the conversion is **QGGGDD.**

5. Diagonal Shift

Ciphertext = **QGGGDD**

Convert it into Grid = $\begin{matrix} Q & G & G \\ G & D & D \end{matrix}$

Shift it from top left to bottom right = $\begin{matrix} D & D & G \\ G & Q & G \end{matrix}$

Ciphertext = **DDGGQG**

6. Reverse Left Shift

Ciphertext = **DDGGQG**

Convert it into Grid = $\begin{matrix} D & D & G \\ G & Q & G \end{matrix}$

Shift the column by one step towards right

$\begin{matrix} D & D & G \\ G & Q & G \end{matrix} = \begin{matrix} G & D & D \\ G & G & Q \end{matrix}$

Ciphertext = **GDDGGQ**

7. Shift by -5

With help of Table 2 shift all letters by -5.

Ciphertext = **GDDGGQ**

G = B

D = Y

D = Y

G = B

G = B

Q = L

Ciphertext = **BYYBBL**

8. Reverse Playfair Cipher

Ciphertext = **BYYBBL**

Key Matrix

U	N	I/J	V	E	R
S	T	Y	A	B	C
D	F	G	H	K	L
M	O	P	Q	W	X
Z	1	2	3	4	5

Make it into Digraphs.

BY YB BL

- **For BY**:
  - Locate **B** and **Y** in the matrix. They form a rectangle.
  - Replace each with the character according to reverse replaced by Right: **A** and **T**.

- **For YB**:

- Locate **Y** and **B** in the matrix. They form a rectangle.
- Replace each with the character according to reverse replaced by right rule: **T** and **A**.

- **For BL**:
  - Locate **B** and **L** in the matrix. They form a rectangle.
  - Replace each with the character according to clockwise direction: **C** and **K**.

Final Ciphertext after decryption = **ATTACK**

4.4. Test 4

Key Preparation:

Developing a matrix with dimensions 6x5 by choosing a keyword. The character 'J' is combined with 'I', and any unnecessary letters are removed.

Keyword: UNIVERSITY

Key Matrix

U	N	I/J	V	E	R
S	T	Y	A	B	C
D	F	G	H	K	L
M	O	P	Q	W	X
Z	1	2	3	4	5

Encryption Process

1. Pairing the Plaintext:

   Plaintext: **SECRET**

   Divide into Digraphs, that are composed of two pairs. If the word appears more than once and if it is not pairing, then use a letter X.

   SE CR ET

   Applying Playfair rules for encryption

   - For **SE**:
   - Locate **S** and **E** in the matrix. They form a rectangle.
   - Replace each with the character according to anticlockwise direction rule: **BU**

- For **CR**:
  - Locate **C** and **R** in the matrix. They form a rectangle.
  - Replace each with the character according to replace by below letter rule: **LC**
- For **ET**:
  - Locate **E** and **T** in the matrix. They form a rectangle.
  - Replace each with the character according to anticlockwise rule: **NB**

After applying the rule, Intermediate Ciphertext: **BULCNB**

2. Shift by 5

   Ciphertext: **BULCNB**

   With the help of Table 2 Shift all letters by 5.

   - B = G
   - U = Z
   - L = Q
   - C = H
   - N = S
   - B = G

   After shifting by 5, Ciphertext is **GZQHSG**

3. Left Shift

   Ciphertext: **GZQHSG**

   Convert ciphertext into grid.

   $$\begin{matrix} G & Z & Q \\ H & S & G \end{matrix}$$

   Now, shift all column by 1 step towards left.

   $$\begin{matrix} Z & Q & G \\ S & G & H \end{matrix}$$

   After left shift process, ciphertext is **ZQGSGH.**

4. Diagonal Shift

   Ciphertext: **ZQGSGH**

   Convert ciphertext into grid.

   $$\begin{matrix} Z & Q & G \\ S & G & H \end{matrix}$$

   Now, shift letter from top left to bottom right.

   $$\begin{matrix} G & H & G \\ S & Z & Q \end{matrix}$$

   After diagonal shift, ciphertext is **GHGSZQ.**

5. Convert Cipher Value into Binary

   Ciphertext = **GHGSZQ**

   Convert it into Binary with help of *Table 4 Letter, Decimal and Alphabetic Positions* and *Figure 4 Binary to Decimal*

Cipher Value	Alphabet Position	Decimal	Binary
G	6	6	00110
H	7	7	00111
G	6	6	00110
S	18	18	10010
Z	25	25	01100
Q	16	16	10000

Table 24 Testing 4 Convert Cipher Value into Binary

   Output Binary: 00110, 00111, 00110, 10010, 01100, 10000

6. Binary XOR Calculation.

   XOR with Key.

   Key= 10001.

   The following table shows the calculation of XOR Operations.

Binary: 00110, 00111, 00110, 10010, 01100, 10000

Binary	Key	Binary XOR Key	XOR Result
00110	10001	00110 XOR 10001	10111
00111	10001	00111 XOR 10001	10110
00110	10001	00110 XOR 10001	10111
10010	10001	10010 XOR 10001	00011
01100	10001	01100 XOR 10001	11101
10000	10001	10000 XOR 10001	00001

Table 25 Testing 4 XOR Calculation Table

XOR Result in Binary: 10111, 10110, 10111, 00011, 11101, 00001

7. Converting Binary to decimal

Convert XOR Result to Decimal by the help of the *Figure 4 Binary to Decimal*.

XOR Result (Binary): 10111, 10110, 10111, 00011, 11101, 00001

XOR Result	Decimal
10111	23
10110	22
10111	23
00011	3
11101	29
00001	1

Table 25 Testing 4 Converting Binary to Decimal

The output after conversion of binary to decimal is 23, 22, 23, 3, 29, 1

8. Decimal to Alphabet using Custom Mapping Table

Now, convert Decimal into Alphabet using custom Mapping Table. With the help of ***C****haracter convert* decimal into alphabet.

Decimal Value: 23, 22, 23, 3, 29, 1

Decimal Value	Numeric Alphabet
23	x
22	w
23	x
3	d
29	D
1	b

Table 28 Testing 4 Conversion of Decimal into Alphabet

The Final Ciphertext after encryption is **xwxdDb**

Decryption Process

1. Alphabet to Decimal Using Custom Mapping Table

   The following table shows the conversion from Alphabet to Decimal Using Custom Mapping Table. Insert ciphertext into Numeric Alphabet column. With the help of Figure 5 Number to Character convert decimal into alphabet. Ciphertext = **xwxdDb**

Numeric Alphabet	Decimal Value
x	23
w	22
x	23
d	3
D	29
b	1

Table 26 Testing 4 Conversion of Alphabet into Decimal

   Decimal Value is 23, 22, 23, 3, 29, 1

2. Converting Decimal Value to Binary.

   Converting Decimal Value into Binary, in which binary is XOR Result.

   Convert it with the help of *Figure 4 Binary to Decimal*.

   Decimal Value: 23, 22, 23, 3, 29, 1

Decimal Value	Binary (XOR Result)
23	10111
22	10110
23	10111
3	00011
29	11101
1	00001

Table 27 Testing 4 Convert Decimal to Binary

Binary (XOR Result): 10111, 10110, 10111, 00011, 11101, 00001

3. Binary XOR Calculation

   Now, with the help of XOR Result and key using XOR Operation decrypt the Binary Value.

   Key = 10001. XOR Result: 00001, 10111, 10111, 10111, 10010, 10010

XOR Result	Key	XOR with Key	Decrypted Binary
10111	10001	10111 XOR 10001	00110
10110	10001	10110 XOR 10001	00111
10111	10001	10111 XOR 10001	00110
00011	10001	00011 XOR 10001	10010
11101	10001	11101 XOR 10001	01100
00001	10001	00001 XOR 10001	10000

Table 29 Testing 4 Binary XOR Calculation

   By using XOR operation Binary is recovered. Decrypted Binary: 00110, 00111, 00110, 10010, 01100, 10000

4. Convert Decrypted Binary into Cipher Value

   Converting Decrypted Binary into Cipher Value by the help of its position. With the help of *Figure 4 Binary to Decimal* and with the help of *Table 4 Letter, Decimal and Alphabetic Positions* Convert it into decimal-alphabet position-letter.

Binary: 00110, 00111, 00110, 10010, 01100, 10000

Binary	Decimal	Alphabet Position	Cipher Value
00110	6	6	G
00111	7	7	H
00110	6	6	G
10010	18	18	S
01100	25	25	Z
10000	16	16	Q

Table 30 Testing 4 Conversion of Binary to Cipher Value

Cipher Value after the conversion is **GHGSZQ.**

5. Diagonal Shift

   Ciphertext = **GHGSZQ**

   Convert it into Grid = $\begin{matrix} G & H & G \\ S & Z & Q \end{matrix}$

   Shift it from top left to bottom right = $\begin{matrix} Z & Q & G \\ S & G & H \end{matrix}$

   Ciphertext = **ZQGSGH**

6. Reverse Left Shift

   Ciphertext = **ZQGSGH**

   Convert it into Grid = $\begin{matrix} Z & Q & G \\ S & G & H \end{matrix}$

   Shift the column by one step towards right

   $\begin{matrix} Z & Q & G \\ S & G & H \end{matrix} = \begin{matrix} G & Z & Q \\ H & S & G \end{matrix}$

   Ciphertext = **GZQHSG**

7. Shift by -5

   With help of Table 2 shift all letters by -5.

Ciphertext = **GZQHSG**

G = B

Z = U

Q = L

H = C

S = N

G = B

Ciphertext = **BULCNB**

8. Reverse Playfair Cipher

Ciphertext = **BULCNB**

Key Matrix

U	N	I/J	V	E	R
S	T	Y	A	B	C
D	F	G	H	K	L
M	O	P	Q	W	X
Z	1	2	3	4	5

Make it into Digraphs.

BU LC NB

- **For BU**:
  - Locate **B** and **U** in the matrix. They form a rectangle.
  - Replace each with the character according to reverse replaced by Right: **S** and **E**.

- **For LC**:
  - Locate **L** and **C** in the matrix. They form a rectangle.
  - Replace each with the character according to reverse replaced by right rule: **C** and **R**.
- **For NB**:
  - Locate **N** and **B** in the matrix. They form a rectangle.

- Replace each with the character according to clockwise direction: **E** and **T**.

Final Ciphertext after decryption = **SECRET**

Test 5

Key Preparation:

Developing a matrix with dimensions 6x5 by choosing a keyword. The character 'J' is combined with 'I', and any unnecessary letters are removed.

Keyword: UNIVERSITY

Key Matrix

U	N	I/J	V	E	R
S	T	Y	A	B	C
D	F	G	H	K	L
M	O	P	Q	W	X
Z	1	2	3	4	5

Encryption Process

1. Pairing the Plaintext:

   Plaintext: **WORLD**

   Divide into Digraphs, that are composed of two pairs. If the word appears more than once and if it is not pairing, then use a letter X.

   WO RL DX

   Applying Playfair rules for encryption

   - For **WO**:
   - Locate **W** and **O** in the matrix. They form a rectangle.
   - Replace each with the character according to replace by right rule: **XP**
   - For **RL**:
   - Locate **R** and **L** in the matrix. They form a rectangle.
   - Replace each with the character according to replace by below letter rule: **CX**

- For **DX**:
  - Locate **D** and **X** in the matrix. They form a rectangle.
  - Replace each with the character according to anticlockwise rule: **ML**

After applying the rule, Intermediate Ciphertext: **BUCXML**

2. Shift by 5

   Ciphertext: **XPCXML**

   With the help of Table 2 Shift all letters by 5.

   - X = C
   - P = U
   - C = H
   - X = C
   - M = R
   - L = Q

   After shifting by 5, Ciphertext is **CUHCRQ**

3. Left Shift

   Ciphertext: **CUHCRQ**

   Convert ciphertext into grid.

   $$\begin{matrix} C & U & H \\ C & R & Q \end{matrix}$$

   Now, shift all column by 1 step towards left.

   $$\begin{matrix} U & H & C \\ R & Q & C \end{matrix}$$

   After left shift process, ciphertext is **UHCRQC.**

4. Diagonal Shift

   Ciphertext: **UHCRQC**

Convert ciphertext into grid.

$U \quad H \quad C$
$R \quad Q \quad C$

Now, shift letter from top left to bottom right.

$Q \quad C \quad C$
$R \quad U \quad H$

After diagonal shift, ciphertext is **QCCRUH.**

5. Convert Cipher Value into Binary

Ciphertext = **QCCRUH**

Convert it into Binary with help of *Table 4 Letter, Decimal and Alphabetic Positions* and *Figure 4 Binary to Decimal.*

Cipher Value	Alphabet Position	Decimal	Binary
Q	17	17	10000
C	2	2	00010
C	2	2	00010
R	16	16	10001
U	20	20	10100
H	7	7	00111

Table 31 Testing 5 Convert Cipher Value into Binary

Output Binary: 10001, 00010, 00010, 10000, 10100, 00111

6. Binary XOR Calculation.

XOR with Key.

Key= 10001.

The following table shows the calculation of XOR Operations.

Binary: 10001, 00010, 00010, 10000, 10100, 00111

Binary	Key	Binary XOR Key	XOR Result
10000	10001	10000 XOR 10001	00001
00010	10001	00010 XOR 10001	10011
00010	10001	00010 XOR 10001	10011
10001	10001	10001 XOR 10001	00000
10100	10001	10100 XOR 10001	00101
00111	10001	00111 XOR 10001	10110

Table 32 Testing 5 XOR Calculation Table

XOR Result in Binary: 00001, 10011, 10011, 00000, 00101, 10110

7. Converting Binary to decimal

   Convert XOR Result to Decimal by the help of the *Figure 4 Binary to Decimal*.

   XOR Result (Binary): 00001, 10011, 10011, 00000, 00101, 10110

XOR Result	Decimal
00001	1
10011	19
10011	19
00000	0
00101	5
10110	22

Table 33 Testing 5 Converting Binary to Decimal

   The output after conversion of binary to decimal is 1, 19, 19, 0, 5, 22

8. Decimal to Alphabet using Custom Mapping Table

   Now, convert Decimal into Alphabet using custom Mapping Table. With the help of Error! Reference source not found.convert decimal into alphabet.

   Decimal Value: 1, 19, 19, 0, 5, 22

Decimal Value	Numeric Alphabet
1	b

19	t
19	t
0	a
5	f
22	w

Table 34 Testing 5 Conversion of Decimal into Alphabet

The Final Ciphertext after encryption is **bttafw.**

Decryption Process

1. Alphabet to Decimal Using Custom Mapping Table

The following table shows the conversion from Alphabet to Decimal Using Custom Mapping Table. Insert ciphertext into Numeric Alphabet column. With the help of *Error! Reference source not found.* convert decimal into alphabet. Ciphertext = **bttafw**

Numeric Alphabet	Decimal Value
b	1
t	19
t	19
a	0
f	5
w	22

Table 35 Testing 5 Conversion of Alphabet into Decimal

Decimal Value is 1, 19, 19, 0, 5, 22

2. Converting Decimal Value to Binary.

Converting Decimal Value into Binary, in which binary is XOR Result. Convert it with the help of *Figure 4 Binary to Decimal*.

Decimal Value: 1, 19, 19, 0, 5, 22

Decimal Value	Binary (XOR Result)
1	00001

19	10011
19	10011
0	00000
5	00101
22	10110

Table 36 Testing 5 Convert Decimal to Binary

Binary (XOR Result): 00001, 10011, 10011, 00000, 00101, 10110

3. Binary XOR Calculation

Now, with the help of XOR Result and key using XOR Operation decrypt the Binary Value.

Key = 10001. XOR Result: 00001, 10011, 10011, 00000, 00101, 10110

XOR Result	Key	XOR with Key	Decrypted Binary
00001	10001	00001 XOR 10001	10000
10011	10001	10011 XOR 10001	00010
10011	10001	10011 XOR 10001	00010
00000	10001	00000 XOR 10001	10001
00101	10001	00101 XOR 10001	10100
10110	10001	10110 XOR 10001	00111

Table 37 Testing 5 Binary XOR Calculation

By using XOR operation Binary is recovered. Decrypted Binary: 10000, 00010, 00010, 10001, 10100, 00111

4. Convert Decrypted Binary into Cipher Value

Converting Decrypted Binary into Cipher Value by the help of its position. With the help of *Figure 4 Binary to Decimal* and with the help of *Table 4 Letter, Decimal and Alphabetic Positions* Convert it into decimal-alphabet position-letter

Binary: 10000, 00010, 00010, 10001, 10100, 00111

Binary	Decimal	Alphabet Position	Cipher Value
10000	16	16	Q

00010	2	2	C
00010	2	2	C
10001	17	17	R
10100	20	20	U
00111	7	7	H

Table 38 Testing 5 Conversion of Binary to Cipher Value

Cipher Value after the conversion is **QCCRUH.**

5. Diagonal Shift

Ciphertext = **QCCRUH**

Convert it into Grid = $\begin{matrix} Q & C & C \\ R & U & H \end{matrix}$

Shift it from top left to bottom right = $\begin{matrix} U & H & C \\ R & Q & C \end{matrix}$

Ciphertext = **UHCRQC**

6. Reverse Left Shift

Ciphertext = **UHCRQC**

Convert it into Grid = $\begin{matrix} U & H & C \\ R & Q & C \end{matrix}$

Shift the column by one step towards right

$\begin{matrix} U & H & C \\ R & Q & C \end{matrix} = \begin{matrix} C & U & H \\ C & R & Q \end{matrix}$

Ciphertext = **CUHCRQ**

7. Shift by -5

With help of Table 2 shift all letters by -5.

Ciphertext = **CUHCRQ**

C = X

U = P

H = C

C = X

R = M

Q = L

Ciphertext = **XPCXML**

8. Reverse Playfair Cipher

Ciphertext = **XPCXML**

Key Matrix

U	N	I/J	V	E	R
S	T	Y	A	B	C
D	F	G	H	K	L
M	O	P	Q	W	X
Z	1	2	3	4	5

Make it into Digraphs.

XP CX ML

- **For XP**:
  - Locate **X** and **P** in the matrix. They form a rectangle.
  - Replace each with the character according to reverse replaced by Right: **W** and **O**.

- **For CX**:
  - Locate **C** and **X** in the matrix. They form a rectangle.
  - Replace each with the character according to reverse replaced by right rule: **R** and **L**.
- **For ML**:
  - Locate **M** and **L** in the matrix. They form a rectangle.
  - Replace each with the character according to clockwise direction: D and **X**.

Ciphertext after decryption = **WORLDX**

Remove the **X** to get **WORLD** as final decrypted result.

5. Evaluation of Strengths and Weaknesses in the Currently Developed Cryptographic Algorithm

This section is aimed at comparing the new encryption algorithm developed in this paper to establish the suitability of this current encryption algorithm in a practical usage and set up of constraints based on their security implications. The evaluation includes at least five advantages and five limitations of the algorithm discussing issues about security, performance, complexity, and key management.

5.1. Strengths of the Cryptographic Algorithm:

- Enhanced Security: Since the algorithm used includes features such as Playfair cipher, shifting cipher, and XOR processor, security is relatively strong than inherent in simple ciphers.
- Complex Encryption Process: The hierarchical approach of 8-4-2 binary and decimal conversions adds layer and difficulty and is rather difficult to decode unless they have the key.
- Flexible Key Use: What is more, the function enables the use of different keys for the XOR and for other cipher operations with flexibility, in addition to getting different encryption outputs with the same key.
- Resistance to Simple Attacks: The fact, that XOR operations are used alternated by shift operations, and both is intercalated with a substitute operation for some of the bytes, makes it all but more difficult to use very simple cryptanalytic methods like frequency analysis or simply add single plaintiff texts and compare the results.
- Customizable: The basic features of the algorithm are Its strength is in its flexibility for example one may choose to change the key length or shift pattern so that it fits the specific requirements.

5.2. Weaknesses of the Cryptographic Algorithm:

- Key Management: One of them is the program relying on a key for authentication and its secure storage and management is a relatively difficult question. The idea is that given the keys are compromised the originally encrypted content can be easily decrypted.
- Computationally Intensive: It is because shifting, binary conversion and XORing processes make the algorithm slower particularly for high volume of data.
- Vulnerability to Known-Plaintext Attacks: Ciphertext and a part of plaintext message are sometimes intercepted, and therefore improper utilization of this info might compromise a cipher.
- Repetitive Patterns: This means that for a single plaintext, the output algorithm produces a recurrent same pattern in the ciphertext if the same keys or shift values are used repeatedly, and therefore the cipher is vulnerable to pattern recognition attack.
- Limited Key Space: Depending upon the key size of the algorithm there could be fewer key possibilities if the key string length is fixed or less than a certain string or certain number of bits hence yielding lesser safety against brute force attack.

5.3. Application Area of the New Cryptographic Algorithm

- Financial Institutions: This algorithm creates a secure environment for customers to transact through Online banking since there is no way an unauthorized person can distort their account balance or execute an unlawful transaction. It can well secure financial data on the move and at rest. This can be done by banks to increase customer reliability and security of information.
- Government and Military: The algorithm can guarantee protection of sensitive information exchanged between various departments and the military. It guarantees that the classified information without divulging or offering them to the adversaries. Advanced encryption helps to protect the data of the state and the armed forces.
- Healthcare: In healthcare, the algorithm shields patient's medical records to meet legal standard such as the HITECH HIPAA. It helps to exclude unauthorized access to personal health information. This encryption enables the protection of conversation between a health care provider and the patients.

- E-commerce: The algorithm can encrypt the customer payment information and purchasing transactions when making purchases online. It has developed secure payment mechanisms for customer on online shopping sites. This goes a long way in protecting people's identifications and their money while shopping online.
- Cloud Storage: The algorithm further acts as a protective mechanism ensuring that data in cloud environments are not breached to grant access to unwanted individuals that may try and access sensitive files. It guarantees security for user data stored in the cloud to the later stages. By using this model users can easily enter secure and access their information since it is secure.

This are the application area where the algorithm can be implemented.

6. Conclusion

The Quadracrypt Cipher is a new generation cryptographic product which is a mix of traditional encryption and computation algorithms for dealing with the emerging issues in cryptography. Using the Playfair Cipher, Caesar Cipher, shifting transformations, and binary XOR operations, the proposed algorithm gains increased stability and increased capacity. Its complex design guarantees immunity to attempts on its penetration, frequency analysis and other forms of cryptanalysis, so it can be effectively used to protect information.

The applicability and flexibility of the algorithm are more observable across sectors ranging from finance, healthcare, government sectors, e-commerce and cloud hosting where data security is paramount. It offers a sound approach for safe and reliable exchange and storage of information, satisfying today's requirements. However, some limitations are quantitatively expensive, and vulnerable to over-fitting where patterns recur under defined circumstances.

In the next development of the Quadracrypt Cipher, developers can concentrate on improving speed and increase the number of key bits useful in providing protection against brute force attacks. Nonetheless, it asserts itself as a worthy innovation in cryptographic sciences proving to present an efficient approach toward addressing the ever-emerging threats in a rapidly globalizing society.

7. References

Authority, D., 2024. *Device Authority.* [Online] Available at: https://deviceauthority.com/symmetric-encryption-vs-asymmetric-encryption/ [Accessed 8 Decmber 2024].

Bacon, M., 2024. *TechTarget.* [Online] Available at: https://www.techtarget.com/searchsecurity/definition/security [Accessed 7 December 2024].

Britannica, 2024. *Britannica.* [Online] Available at: https://www.britannica.com/technology/quantum-computer [Accessed 7 December 2024].

CISSPPREP, 2024. *CISSPPREP.* [Online] Available at: https://cissprep.net/cryptography-terminology/ [Accessed 8 December 2024].

Cryptool, 2024. *Cryptool.* [Online] Available at: https://www.cryptool.org/en/education/history/ [Accessed 7 December 2024].

Fasulo, P., 2021. *SecurityScoreCard.* [Online] Available at: https://securityscorecard.com/blog/what-is-the-cia-triad/ [Accessed 7 December 2024].

GeeksForGeeks, 2024. *GeeksForGeeks.* [Online] Available at: https://www.geeksforgeeks.org/substitution-cipher/ [Accessed 8 December 2024].

GeeksForGeeks, 2024. *GeeksForGeeks.* [Online] Available at: https://www.geeksforgeeks.org/difference-between-symmetric-and-asymmetric-key-encryption/ [Accessed 8 December 2024].

Krichen, M., 2023. *Encylopedia.* [Online] Available at: https://encyclopedia.pub/entry/44469 [Accessed 7 December 2024].

point, t., 2024. *Tutorial Point.* [Online] Available at: https://www.tutorialspoint.com/what-are-the-substitution-techniques-in-information-security [Accessed 7 December 2024].

ResearchGate, 2024. *ResearchGate.* [Online] Available at: https://www.researchgate.net/figure/Character-to-number-modified_tbl2_350962289 [Accessed 22 December 2024].

Sidhpurwala, H., 2023. *Red Hat.* [Online] Available at: https://www.redhat.com/en/blog/brief-history-cryptography [Accessed 5 December 2024].

SwiftTips, 2024. *SwiftTips.* [Online] Available at: https://www.swifttips.com/preview/Binary [Accessed 22 December 2024].

8. Appendix

8.1. Cryptosystems

A cryptosystem is a system of methods, that involve algorithm, protocols, and keys that are used to safeguard information. It provides data security, data integrity, data authenticity and non-repudiation by use of encryption and decryption. Based on key type, those can be divided into the following: M: Symmetric key cryptography and F: Asymmetric key cryptography. A hybrid cryptosystem often uses both forms but incorporate the best part of both for secure communication. (Bernstein, 2025)

Cryptosystems involve key management in cases of generation, exchange and storage of the cryptographic keys. Managing the key is a decisive step of ensuring total security of the system and it prevents the keys to be disclosed or lost with no consent. The most well-known cryptographic algorithms are AES for symmetric system with different key sizes and RSA for asymmetric system widely used for security, communications, message/signatures and other secure activities such as finance. (Bernstein, 2025)

Today, cryptosystems have significant purposes in protecting various forms of information exchange, online finances, storage of information, and blockchain. They are useful in safeguarding data when in transit and to discourage people who have no business with that information from getting in touch with it. Cryptosystems are also constantly being developed with new algorithms and process to confront ever arising security issues in a world that is rapidly going digital. With the continued improvement in the cryptographic solutions, cryptosystems shall continue to be an integrated aspect in the protection of interactions or transactions in different fields. (Bernstein, 2025)

8.2. Components of a cryptosystem

A basic cryptosystem includes the following:

Plaintext. Information that is not fit to be transmitted in an encoded state.

Ciphertext. The supplementary communicant of the plaintext information, which is in an encrypted form, or, in other words, it is incomprehensible without a decryption key.

Encryption algorithm. The computing process by which a message is transformed from plaintext to ciphertext with a mathematical formula. It also generates a special encryption key for that text which is not possible with conventional algorithms.

Decryption algorithm. The cryptographic system that can be used in encrypting any given text/ciphertext to produce text/plaintext. It also employs the private decryption key for that text as well.

Encryption key. The value that is to be encrypted, and this value is usually held with the sender in computing the ciphertext value of the given plaintext.

Decryption key. The value understood by the receiver that is used to DE massage the received ciphertext into plaintext.

These are the components of Cryptosystems. (Bernstein, 2025)