Development of Cryptographic Algorithm

Deepak Bokati

London Metropolitan University

Abstract

This is the document contains the detailed description of the individual project "Development of Cryptographic Algorithm" developed for converting the plain text into cipher text and vice versa. In addition, the major purpose of the cryptographic algorithm is not just conversion of plain text but provides the working mechanism of the cryptographic algorithm. This report primarily consists of many distinct chapters start from introduction, history, different types of algorithms, and many more. This report highlights older versions of algorithms have many flaws which can be easily compromised by the attacker. While compromising data of organization, all the important data of organization got damaged. To overcome these types of flaws, we have done much research related to cryptographic algorithms and have modified the less secure algorithms. In simple word, this report highlights the importance of secure cryptographic algorithm in an organization for securing their data's.

Table of Contents

1. Introduction to Cryptographic Systems 1
  1.1. An overview of Security 1
  1.2. CIA Triad 4
  1.3. History of Cryptography 8
  1.4. Cryptography and Cryptographic Terminologies 9
    1.4.1. Modern Cryptographic System 11
  1.5. Aim and Objectives 13
    1.5.1. Aim 13
    1.5.2. Objectives 13
2. Background of Play fair Cryptographic Algorithm 14
  2.1. Encrypting and decrypting plain text using Playfair Algorithm 16
  2.2. Advantages and Disadvantages of Playfair Algorithm 18
    2.2.1. Advantages 18
    2.2.2. Disadvantages 18
3. Development of new Cryptographic Algorithm 19
  3.1. Algorithm of newly developed cryptographic system 25
    3.1.1. Algorithm for Encryption 25
    3.1.2. Algorithm for Decryption 29
  3.2. Necessary of modification 32
  3.3. Working mechanism of newly created ‘*PlayRS’* Cryptographic Algorithm 33
  3.4. Flowchart 35
    3.4.1. Flowchart for Encryption 36
    3.4.2. Flowchart for Decryption 37
4. TESTING 38

4.1. TEST: 1 38

4.2. TEST: 2 47

4.3. TEST: 3 55

4.4. TEST: 4 63

4.5. TEST: 5 74

5. Critical Evaluation of the new Cryptographic Algorithm 88

5.1. Advantages and Disadvantages of PlayRs Cryptographic Algorithm 88

5.1.1. Advantages of ‘*PLAYRS’* Algorithm 88

5.1.2. Disadvantages of *‘PLAYRS’* Algorithm 88

5.2. Application Areas of PlayRs Cryptographic Algorithm 89

6. Conclusion 90

7. Reference and Bibliography 91

8. Appendix 94

List of Figures

Figure 1: Total Malware Infection Growth rate (purplesec, 2020)................................... 2
Figure 2: Average cost of data breaches (purplesec, 2020)... 2
Figure 3: Most affected Organization by data breaches (digitalguardian, 2020) 3
Figure 4: Tools required for authentication (Tamassia & Goodrich, 2014) 5
Figure 5: Figure of Hieroglyph (tutorialspoint, 2021) 8
Figure 6: Example of Caesar Shift Cipher (tutorialspoint, 2021) 9
Figure 7: Example of Symmetric Encryption (Tamassia & Goodrich, 2014)................. 12
Figure 8: Example of Asymmetric encryption (Tamassia & Goodrich, 2014). 13
Figure 9: XOR Gate.. 20
Figure 10: Flowchart for Encryption Process... 36
Figure 11: Flowchart for Decryption Process 37
Figure 12: Screenshot 1 of Appendix (P et al., 2017)... 94
Figure 13: Screenshot 2 of Appendix (P et al., 2017)... 95
Figure 14: Screenshot 3 of Appendix (P et al., 2017)... 96
Figure 15: Screenshot 4 of Appendix (Srivastava et al., 2010) 97

List of Tables

Table 1: 5*5 Matrix table of Playfair algorithm used during Encryption process 16
Table 2: 5*5 Matrix table of Playfair algorithm used during Decryption process 17
Table 3: Table of XOR Logical Operator 19
Table 4: Table showing example of LSB 21
Table 5: Table of 8*8 matrixes 26
Table 6: Table of 8*8 Matrixes 30
Table 7: 8*8 matrix table 33
Table 8: Table of Test 1 38
Table 9: Making 8*8 matrix for encryption of testing 1 38
Table 10: Table of doing logical operations in testing 1 39
Table 11: Table for obtaining cipher text of testing 1 40
Table 12: Table of encrypting keys in testing 1 42
Table 13: Table of decrypting keys in testing 1 43
Table 14: Making 8*8 matrix for decryption of testing 1 44
Table 15: Table for doing XOR operator in testing 1 45
Table 16: Table for obtaining cipher text for decryption in testing 1 46
Table 17: Table of Test 2 47
Table 18: Making 8*8 matrix for encryption of testing 2 47
Table 19: Table of doing logical operations in testing 2 48
Table 20: Table for obtaining cipher text of testing 2 49
Table 21: Table of encrypting keys in testing 2 51
Table 22: Table of decrypting keys in testing 2 52
Table 23: Making 8*8 matrix for decryption of testing 2 52
Table 24: Table for doing XOR operator in testing 2 53
Table 25: Table for obtaining cipher text for decryption in testing 2 54
Table 26: Table of Test 3 55
Table 27: Making 8*8 matrix for encryption of testing 3 55
Table 28: Table of doing logical operations in testing 3 56
Table 29: Table for obtaining cipher text of testing 3 57

Table 30: Table of encrypting keys in testing 3 59
Table 31: Table of decrypting keys in testing 3 60
Table 32: Making 8*8 matrix for decryption of testing 3.. 60
Table 33: Table for doing XOR operator in testing 3 61
Table 34: Table for obtaining cipher text for decryption in testing 3 62
Table 35: Table of Test 4 63
Table 36: Making 8*8 matrix for encryption of testing 4.. 63
Table 37: Table of doing logical operations in test 4 65
Table 38: Table for obtaining cipher text of testing 4.. 66
Table 39: Table of encrypting keys in testing 4 68
Table 40: Table of decrypting keys in testing 4 70
Table 41: Making 8*8 matrix for decryption of testing 4.. 71
Table 42: Table for doing XOR operator in testing 4 72
Table 43: Table for obtaining cipher text for decryption in testing 4 73
Table 44: Table of Test 5 74
Table 45: Making 8*8 matrix for encryption of testing 5.. 74
Table 46: Table of doing logical operations in test 5 76
Table 47: Table for obtaining cipher text of testing 5.. 78
Table 48: Table of encrypting keys in testing 5 80
Table 49: Table of decrypting keys in testing 5 82
Table 50: Making 8*8 matrix for decryption of testing 5.. 83
Table 51: Table for doing XOR operator in testing 5 84
Table 52: Table for obtaining cipher text for decryption in testing 5 86

1. Introduction to Cryptographic Systems

1.1. An overview of Security

The foundation of security is started with the concept of computer security. The need for computer security emerges from WWII when the first mainframe computers were developed and utilized for communication to prevent breaking of a message from the enemy.

“Security is protection from those who would harm, intentionally or otherwise (Whitman & Mattord, 2018)”. In simple words, security can be defined as the prevention/protection from ones who try to damage, steal the company or individual important documents, hardware component, intentionally or non-intentionally.

In this present time, the total number of internet users is increasing rapidly. Internet allows us to communicate with each other’s. There are so many pros of using internet, but there are cons also. With the increase in the numbers of users, it is very difficult for us to protect our information and data’s. At present time internet is the most crowded and is not free from cybercriminals. Because of the crowd and cybercriminals we are failed to protect our information. The number of cyber-attack is also increasing in the last past years. The best method choose by attacker is phishing for compromising any data of an information.

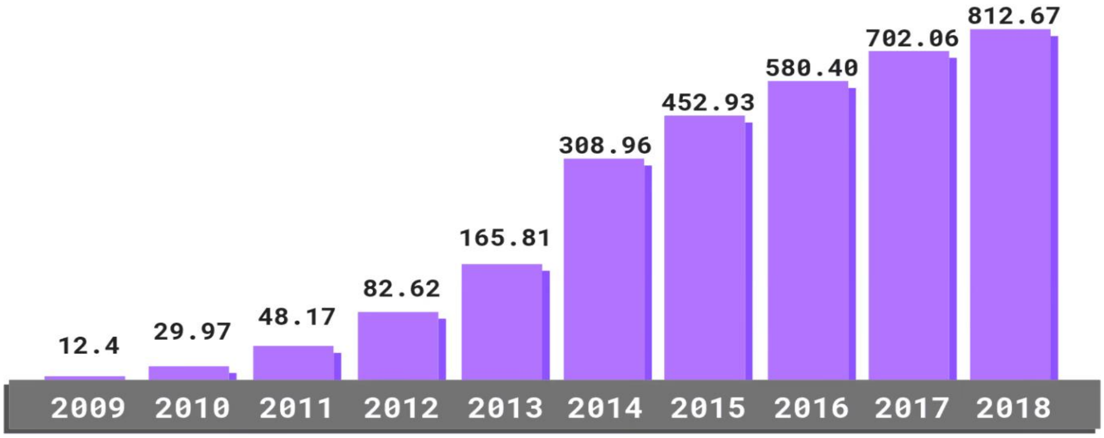


Figure 1: Total Malware Infection Growth rate (purplesec, 2020)

This chart clearly shows the malware infection has been increasing per year. Other methods such as ransom ware attack is also increased by 350% in 2018, social engineering is the one of the most methods used by cyber criminals. According to one report around 98% of cyber-attack is done using social engineering technique (csoonline, 2020). According to statics from IBM the total loss from cyber-attack is shown in figure below:

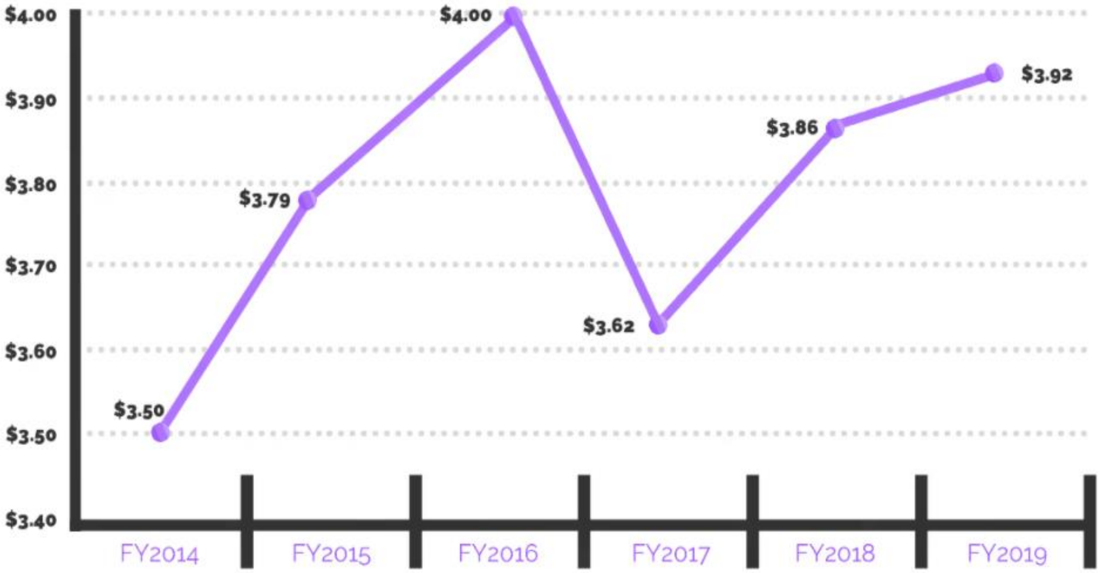


Figure 2: Average cost of data breaches (purplesec, 2020)

Among all countries, USA experiences the highest data breach in the past year. According to research paper the most affected organization by data breaches is health industry. The average cost of data breaches in health industry is around $6.45 million (digitalguardian, 2020). The detailed view of average loss is shown in figure below:

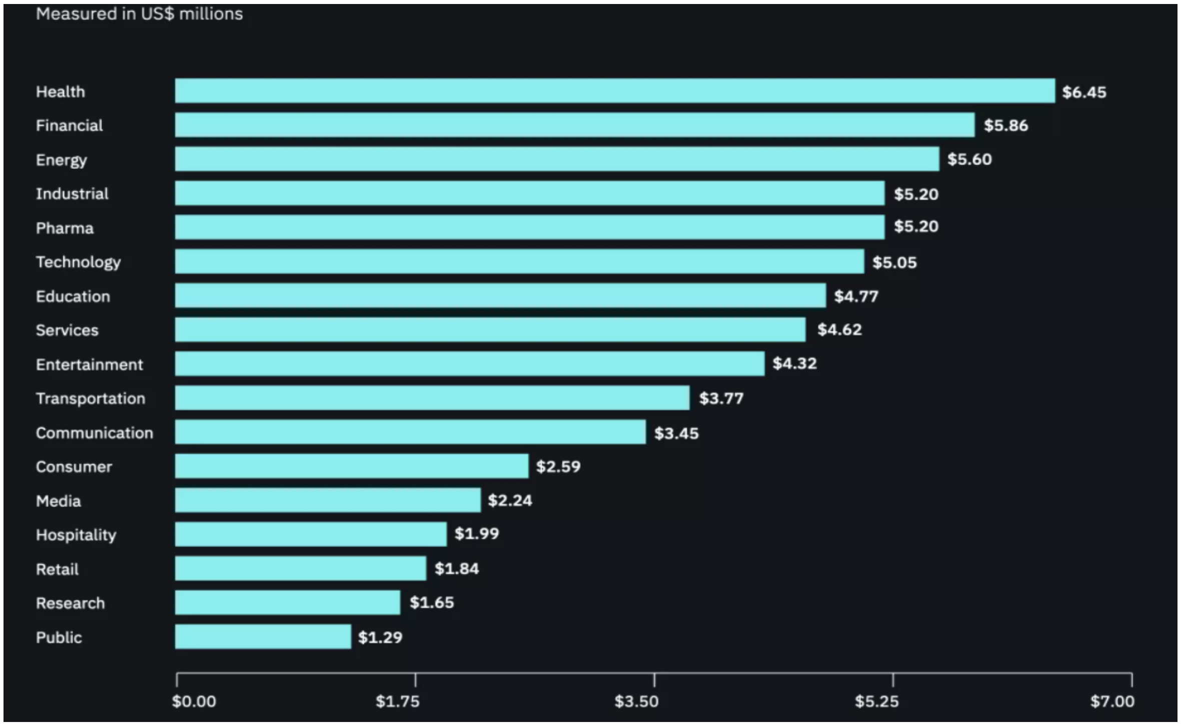


Figure 3: Most affected Organization by data breaches (digitalguardian, 2020)

Above given charts shows that total number of cyber-attack and costs of data breaches in the past year. We can clearly see that the total number of attack is increasing rapidly. So, we have to protect our information at this time by giving different security to our information and data. To maintain security of the data/information of assets or individual, there are three critical components: Confidentiality, Integrity, and Availability, which must be followed to protect our information (Whitman & Mattord, 2018).

1.2. CIA Triad

- **Confidentiality**

Confidentiality simply means avoidance of unauthorized access to information. It protects data by providing information to those who needed it like authorized users and block unauthorized users from accessing information. When the information or data is accessible to unauthorized systems or users, then this is a breach of confidentiality. The main concept of this process is to keep data secret between client and user. For example, if I wanted to send any confidential information to the client, then it must be only sent to that person. If any third person is also able to access that information then the data sent by me is not confidential. This is a violation of confidentiality. So, confidentiality is an important element for building and developing trust (csoonline, 2020). There are some of the mechanisms to keep confidentiality:

- **Encryption**

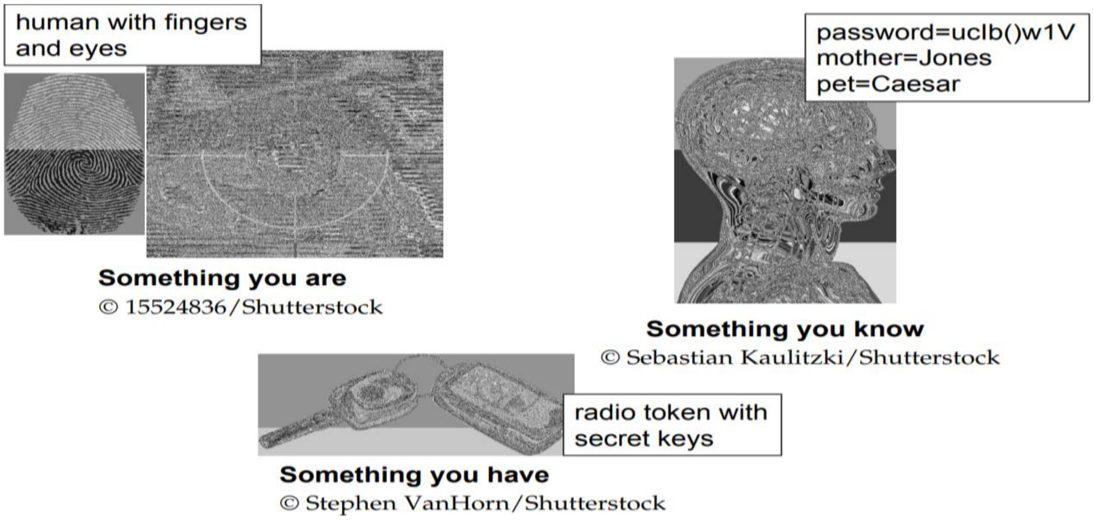
  It is the process of converting normal message to cipher text using encryption key so that the information cannot be stolen by unauthorized users while transmitting it over the network. When the information is sent from the sender, then information is converted using encryption key and that information is send to the receiver through internet. At receiver side converted information is transmitted into the original message using decryption key. Those keys sometime are same for encryption and decryption of data, and those keys are kept secure (ibm, 2020).

- **Identification**

This is simply a rule for blocking unauthorized user from accessing authorized information. Identification is the first step for accessing access to authorized material. In this process, you have to provide something to known for accessing information. Identification can be done by various mechanisms such as username, name, and so on (Tamassia & Goodrich, 2014). This method helps to block unauthorized user form accessing authorized information (Whitman & Mattord, 2018).

- **Authentication**

This is also a rule for accessing authorized information. In this process, you have to give identity or prove that you are the person whom you are claiming to be. Authentication can be done through various mechanisms such as something person has like keys, passwords, pins, smart key, Fingerprints, Tokens, and other as shown in the figure below (Tamassia & Goodrich, 2014). This is method after identification for accessing authorized information.



Something you are
© 15524836/Shutterstock

Something you know
© Sebastian Kaulitzki/Shutterstock

Something you have
© Stephen VanHorn/Shutterstock

Figure 4: Tools required for authentication (Tamassia & Goodrich, 2014)

- **Authorization**

  This is the rules designed for users to limit the accessing of the resources according to their account type. It is a process which allows authentic users to get access to use resources by checking whether the user has access right to the system or not (Tamassia & Goodrich, 2014). For example if we want to see the code of any platform like YouTube, Facebook and so on then we are not able to access those because we are not allowed to get access of those resources (Whitman & Mattord, 2018).

- **Integrity**

  Integrity simply means avoidance of information alteration by unauthorized access. It protects data by providing only access to an authorized person. The main concept of this process is that only authorized users are allowed to alter/change the content of data. Many entities are designed to damage or steal or change the data while information is transmitted over the network. For example, if I send confidential data to the client, then the only client is allowed to change the content of the information. If the third person successfully changes the content of information then this is a violation of integrity (certmike, 2019). There are some mechanisms to keep integrity:

- **Backups**

  This is the process for securing our data even the data are stolen or damaged by the attacker. It is simply a process of saving our files in multiple palaces so that the altered information from unauthorized access must be restored. Backups must be done periodically to secure our information. If we have not done any backups of the required file and are sending it to the client. If attacker is able to change the content of that file at that time it is very difficult for us to restore the original content. So we must have to do backups (Whitman & Mattord, 2018).

- **Checksums**

  It is the most popular method for identifying weather the content of the information is altered or not. Even a small change in the file may result in different outputs. Assume we send any confidential information to client. If client want to check whether the information is changed or not then he uses checksum for identifying whether the information that I received is original or altered. In simple word checksum are used to identify whether the integrity of data is breached or not (Whitman & Mattord, 2018).

- **Availability**

Availability simply means that whenever the necessary information of assets or organization is required, then it must be available to the designated user. It allows accessing of information to all the authorized users without interruption. For example, if I send confidential information to a client. If he is not able to access the confidential information that I have sent, then this is a violation of availability (certmike, 2019). There are certain mechanisms to keep availability:

- **Physical protection**

  In this process, physical protection is provided to the data storing assets such as from physical damage. Maintenance of servers, routers, storage devices, and so on must be done periodically to keep information available all time. There are many mechanisms to give physical protection such as backup servers, desktops, and so on (Whitman & Mattord, 2018).

1.3. History of Cryptography

The phrase cryptography consists of combination of two words, ‘Crypo’ and ‘graphic’ meaning secret and written. The history of cryptography is thought to have developed before evolution of the art of the writing. Since many kingdoms are changing, ideas are required to interact secretly. As there are evolvements of many kingdoms, there is need of ideas for communicating secretly. Such principles are more evolved and cryptography is developing and is being used thus far. Cryptography is thought to be first used in Roman and Egyptian cultures, according to several academic papers. Egyptians used ***Hieroglyph*** around 4000 years ago when interacting with each other, and it is the oldest method of cryptography. The code must be kept confidential and known only to sender and recipient (tutorialspoint, 2021).

Figure 5: Figure of Hieroglyph (tutorialspoint, 2021)

Around 500 to 600 BC the use of simple mono alphabetic substitution ciphers are increased. Roman uses the common cryptographic method known as Caesar Shift Cipher at that time, operating the concept of shifting the message letter by three (tutorialspoint, 2021).

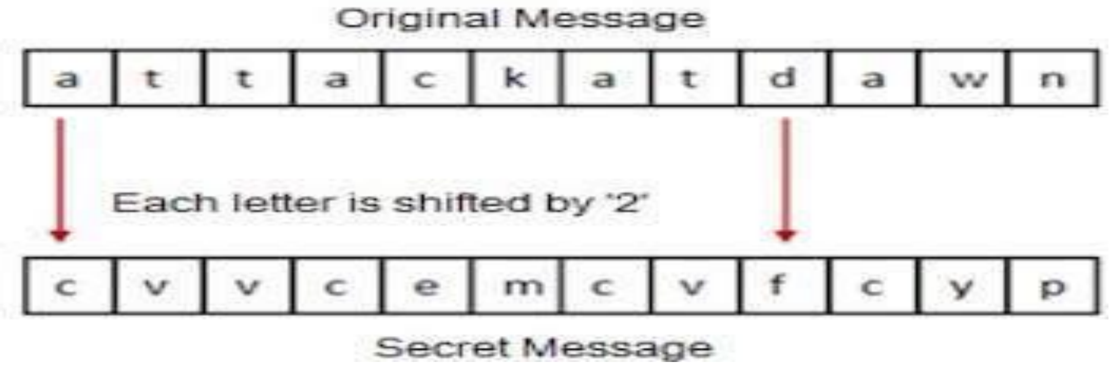


Figure 6: Example of Caesar Shift Cipher (tutorialspoint, 2021)

Other techniques such as ***Steganography*** were later developed and used to secure information by concealing information from unauthorized individuals. In this way, cryptography has advanced and new methods have been developed and used so far to encrypt and protect information (tutorialspoint, 2021).

1.4. Cryptography and Cryptographic Terminologies

Cryptography is simply hiding the original data. In other words, it is a set of rules which is used to ***conceal*** (i.e. hide) confidential information except for user and client by converting ***plain text*** (i.e. Original message) to ***cipher text***. Cipher text also known as ***encryption*** process is the conversion of the original message to the cipher text. Generally, we encrypt the plain plaintext by encrypting all bits at a time or a single bit at a time. When we encrypt all the plaintext at a time by creating blocks of bits then this process is known as ***Block Cipher***. And when we encrypt a single bit of the plain text at a time then this process is called ***Steam Cipher*** (Tamassia & Goodrich, 2014). Suppose, if I want to send confidential information ‘HELLO’ to the client. In this case, the confidential information ‘HELLO’ is called the ***plain text*** and it is in the readable format so, we use an encryption algorithm that gives an output of ***cipher text*** (i.e. @#Qw). We can encrypt plain text by using:

- **Substitution method**

  In this method of the encryption process, we generally replace the letter of the plain text with another letter, numbers, or symbols. For example, there is plain text = HELLO, and the corresponding of H=*, E=#, L=1, O=~. Now the ciphertext = *#11~. In this way, plain text is converted to ciphertext. Caesar cipher, play fair ciphers are some of the examples of the substitution method (Tamassia & Goodrich, 2014).

- **Transposition method**

  In this method of the encryption process, we generally interchange the position of alphabets. For example, there is a plain text = MIKE. Now the positions of alphabets are changed to the following: M=4, I=3, K=1, E=2 to get cipher text=KEIM. In this way, the plain text gets converted to the ciphertext. Generally, the transposition method of encryption is considered to be weak because the frequency of the plain text and same text are equal, and it contains as much information for decryption (Tamassia & Goodrich, 2014).

When the plain text is converted to the ciphertext then we have to again convert cipher text to plain text. This process is called ***Cryptanalysis*** also known as decryption. The decryption process uses some conceal information that must be known to the sender and receiver, but not to the third person. The decryption algorithm uses an assistance input which must be confidential numbers, alphabets, and so on to decrypt the message known as the ***decryption key.*** Similarly, the encryption algorithm also uses assistance input which must be confidential numbers, alphabets, and strings to encrypt the message known as an ***encryption key***. Both the key should be kept confidential. There are many algorithms for encryption and decryption such as Caesar cipher, play fair cipher, and so on. They all use both plain text and secret key for both encryption and decryption (Tamassia & Goodrich, 2014).

1.4.1. Modern Cryptographic System

Modern cryptographic systems are far better and complicated to break than the older cryptographic system. For example, DES (i.e. Data Encryption System) uses both encryption and decryption key of length 48 bits, and also uses 16 rounds for encryption and decryption of data (geeksforgeeks, 2020). In the modern cryptographic system encryption and decryption is done by two mechanisms (Tamassia & Goodrich, 2014):

- **Symmetric Encryption**

  This is the most common cryptographic algorithm in which the same key is used for encryption and decryption of plain text. This is also known as a shared key cryptosystem. Suppose the sender wants to send information to the receiver. The key must be shared by the sender and receiver to communicate. In this process of communication ***public key*** is used for encrypting and decrypting of data (Tamassia & Goodrich, 2014).

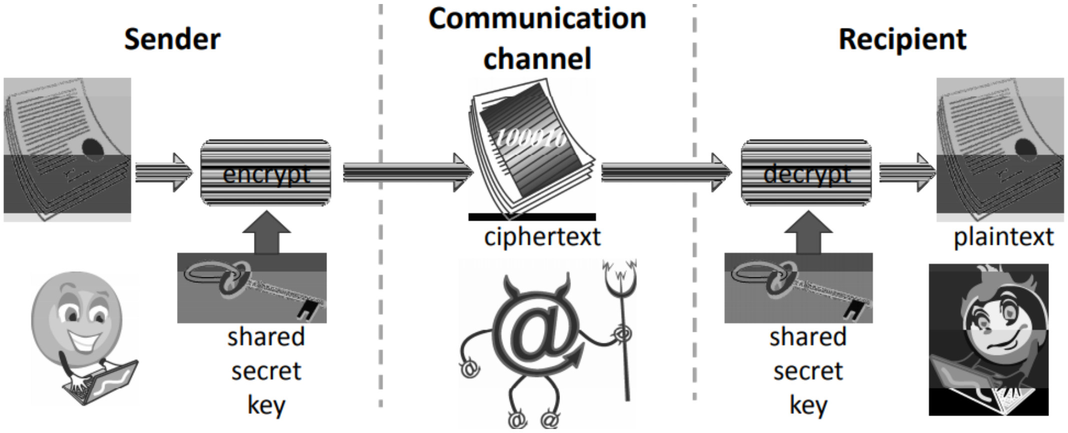


Figure 7: Example of Symmetric Encryption (Tamassia & Goodrich, 2014).

- ### Asymmetric Encryption

  It is also the type of cryptographic algorithm used for encryption and decryption of data where different keys are used. Public and private keys are used in this algorithm for encryption and decryption of data. Suppose, if the sender/receiver wanted to send/receive confidential information then at first he/she has two keys: a ***private key*** and a ***public key***. The public key must be broadcast for encryption of the plain text and the private key must be kept conceal and only known to the receiver for the decryption of the cipher text (Tamassia & Goodrich, 2014).

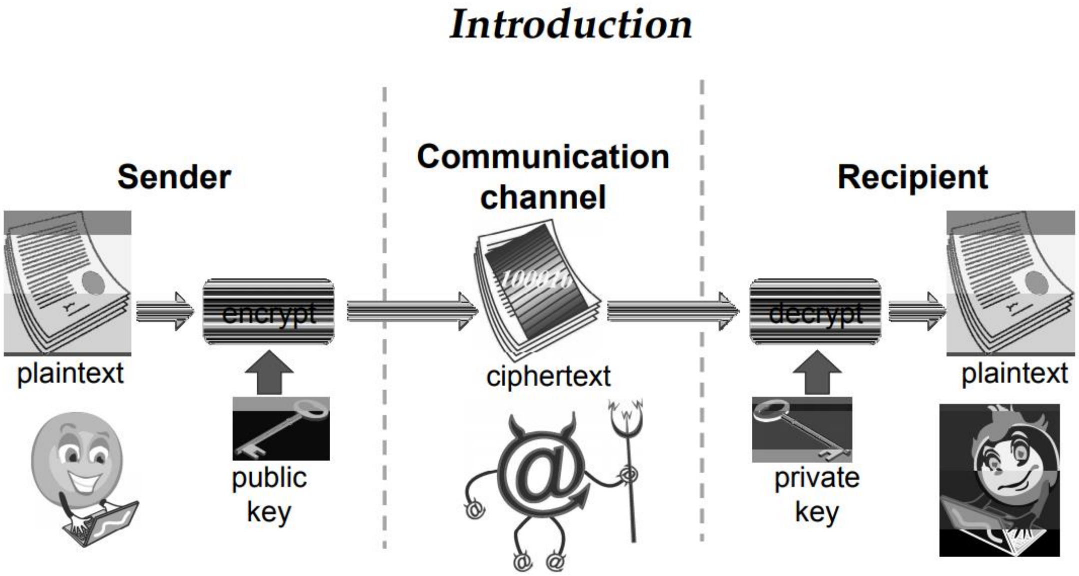


Figure 8: Example of Asymmetric encryption (Tamassia & Goodrich, 2014).

1.5. Aim and Objectives

1.5.1. Aim

The main aim of this report is to learnt about different types of cryptographic algorithm and working mechanism of their algorithm and to develop our own algorithm by modifying existing algorithms.

1.5.2. Objectives

- Development of new cryptographic algorithm
- Development of algorithm, flowchart of new cryptographic algorithm
- Briefly description of working mechanism of newly developed algorithm
- Doing black box testing

2. Background of Play fair Cryptographic Algorithm

In World War 1 nation utilizes a field of a cipher as their correspondence communication system till the finish of the war. In 1854 sir Charles Wheatstone, the physicist, mathematician, and architect invented the play fair code. The name play fair is acquired by the noble Lyon Playfair who spends numerous years promoting the code. In 1890, the British armed force used the play fair code, and use during the Boer war and was as yet utilized (geeksforgeeks, 2019) (Dooley, 2018).

The play fair cryptography system is a substitution and symmetric system that encrypts two letters at a time like block cipher. During this algorithm blocks are made per the plain text which consists of only two letters for encryption. This algorithm relies on the five by five matrix table that uses total alphabets 25 of the 26 letters. Alphabet I and J are either placed within the single box, or we've to merely drop one Alphabet either I or J. The given keyword (i.e. KEY) is placed in column by column, dropping duplicates letters. Then the remainder of the letters is filled into the five by five matrix table in keeping with the given key to complete the matrix table (Bishop, 2003).

The plain texts are encrypted according to the certain rules and the rules are:

1) The plain text is broken into blocks that encompass two letters. If the letter repeats within the same block then remove one letter from that block and insert a null letter like 'X'. That removed letter must be inserted into the text block (geeksforgeeks, 2019).

2) Each letters within the same block must be encrypted separately.

3) If the 2 letters of the same block are found in the matrix table inside the same column, then the letter is encrypted by converting plain text to cipher text with the letter immediately below (geeksforgeeks, 2019).

4) If the 2 letters of the same block are found in the matrix table inside same row, then the letter is encrypted by replacing the plain text to cipher text with the letter immediately below (geeksforgeeks, 2019).

5) If the 2 letters of the same block are found in the matrix table inside various row and column then the letter is encrypted by replacing plain text to cipher text by completing the rectangle. We need to move the plain text to a letter on its own row or a column that is filled by other plain text (geeksforgeeks, 2019).

The encrypted text (i.e. Cipher text) is decrypted according to the certain rules:

1) The encrypted texts are broken into blocks that encompass two letters.

2) Each letters within the block must be decrypted separately.

3) If the 2 letters of the same block are attendance within the same column in the matrix table then the letter are encrypted by changing plain text with the instant upper letter (geeksforgeeks, 2019).

4) If the 2 letters of the same block are present within the same row in the matrix table then the letter is encrypted by changing plain text with the instant left letter (geeksforgeeks, 2019).

5) If the 2 letters of the block are placed within the different row and column then the letter is encrypted by completing the rectangle. We've to switch the plain text with the letter located in its own row or column occupied by other plain text (geeksforgeeks, 2019).

2.1. Encrypting and decrypting plain text using Playfair Algorithm

While encrypting the plain text using Playfair cipher first of all we have to follow certain rules of the encryption process. Here is the example of encrypting plain text using Playfair cipher:

Plain Text = I AM DEEPAK
Key = MIXEDCIPHERS

First of all delete the duplicates letter of the key.
Now, after dropping = MIXEDCPHRS

Now, making 5*5 matrix table and filling according to the rules.

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

Table 1: 5*5 Matrix table of Playfair algorithm used during Encryption process

Now, separating the blocks and replacing the duplicate letter of plain text using ‘X’
IA MD EX EP AK

Cipher text after encryption
AT IC DE DH BL

Hence, the encrypted text is ATICDEDHBL with key MIXEDCPHRS. In this way, we encrypt the plain text.

Now, decrypting the encrypted text ATICDEDHBL using key MIXEDCPHRS

Now, making 5*5 matrix table and filling according to the rules.

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

Table 2: 5*5 Matrix table of Playfair algorithm used during Decryption process

Making block of encrypted text with two letters at each block:
AT IC DE DH BL

Obtained plain text after decryption using Playfair cipher is:
IA MD EX EP AK

Hence, the original plain text = I AM DEEPAK
Using key = MIXEDCPHRS

So, in this way, we Encryption and Decryption of plain text is done using the Playfair algorithm.

There is Advantages as well as disadvantages of every cryptographic algorithm. Talking about the advantages and disadvantages of playfair cryptographic algorithm, here are some of the pros and cons of the Play fair cryptographic algorithm given below:

2.2. Advantages and Disadvantages of Playfair Algorithm

2.2.1. Advantages

i. It is faster cryptographic algorithm while encrypting and decrypting data.
ii. It is extremely secured than other symmetric cryptographic algorithm.
iii. It is easy to use.
iv. It does not require lot of computer resources.

2.2.2. Disadvantages

i. The alphabets I and J are either considered as a one or one alphabet is dropped.
ii. New key must be generated while sending information to different groups.
iii. Special characters, numbers cannot be inserted for encryption.
iv. It is a symmetric algorithm. So, the encryption and decryption keys are same.
v. It is a diagraph substitution cipher (i.e. the inverse of the plain text AB is BA). This will help attackers to explode with the help of frequency analysis (techrejects, 2014).

3. Development of new Cryptographic Algorithm

Developing/modifying new cryptographic algorithm is not easy. We modify playfair cryptographic algorithm using different logical operations. After using logical operator into the playfair algorithm it seems to be more secure than previous. Here are all of the mathematical and logical operations that are used during modifying the selected playfair cryptographic algorithm:

- **XOR logical operator**

This is one of the famous Boolean logical operators among the entire which is widely used in cryptography. We have selected XOR logical operator for modifying the selected playfair algorithm. We use XOR while encrypting the plain text and decrypting the cipher text. The working mechanism of XOR logical operator is it simply compares two binary bits and as a result it gives single output in the form of single binary bit. Here, is the example of XOR operator between two binary numbers with Boolean expression:

Input		**Output**
A	**B**	**Y**
0	0	0
0	1	1
1	0	1
1	1	0

Table 3: Table of XOR Logical Operator

Boolean Expression of XOR logical operator

$$\mathbf{Y = (A \oplus B) = A.B + A.B}$$

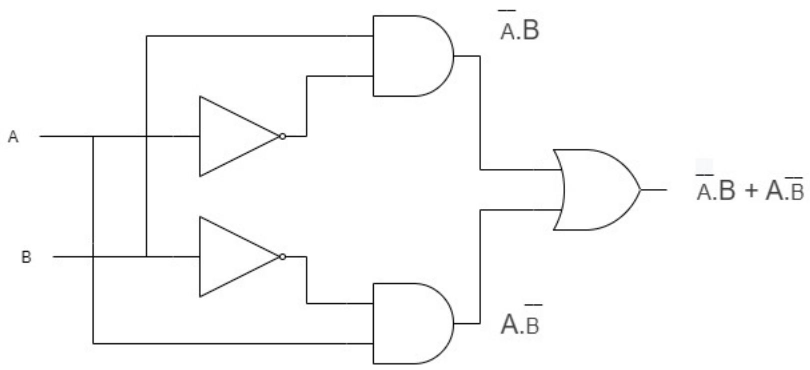


Figure 9: XOR Gate

Explanation of working mechanism of XOR logical operator

XOR logical operator takes two binary numbers as an input while giving output of single binary number. It only accepts binary bits as an input. When the two input numbers are same then it gives result as 0. When the two input numbers are different then it gives result as 1. For example when user gives two binary 0 and 0 or 1 and 1 as an input. Then it will give 0 as an output. Similarly, when user gives 1 and 0 or 0 as an input binary bit. Then it will give output as 1. In this way, XOR logical operator works (allaboutcircuits, 2020).

- **2's Complement**

This is one of the most important mathematical operators used while modifying the selected playfair cipher. We have selected 2's complement for modifying playfair cryptographic algorithm. We use 2's complement for encrypting plain text and decrypting. The working mechanism of 2's complement is simple that it only requires minimum input as single binary bit (tutorialspoint, 2020). Here is the example and detail explanation of the working mechanism of 2's complement with examples.

Explanation of working mechanism of 2's Complement

For finding 2's complement we have to first calculate the 1's complement. Calculation of 1's complement is very easy. For calculating 1's complement we simply have to invert the binary digit. There are two binary digits: 0 and 1. If the given binary digit is 1 then its 2's complement is 0. And if the given binary digit is 0 then its 2's complement is 1. Here is the example of converting simple binary digit into 1's complement.

Binary number = 1 0 0 1 1 0 1

Now inverting the given binary number and result after inverting the binary number:

1's complement = 0 1 1 0 0 1 0

In this way 1's complement is calculated. After calculating 1's complement now we have to calculate 2's complement. For calculating 2's complement we simply have to add binary digit 1 to the least significant bit (LSB). In the above 1's complement LSB is shown in the given figure below:

LSB ↓

0	1	1	0	0	1	0

Table 4: Table showing example of LSB

Now adding 1 to the LSB.

$$
\begin{array}{rccccccc}
 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
 & & & & & & + & 1 \\
\hline
= & 0 & 1 & 1 & 0 & 0 & 1 & 1
\end{array}
$$

Hence, the 2’s complement of the binary bit 1001101 is 0110011. In this way we calculate 2’s complement (tutorialspoint, 2020).

- **RSA Algorithm**

This is the one and only cryptographic algorithm used for modification of playfair cryptographic algorithm. This RSA algorithm is specially used for encrypting and decrypting keys. RSA cryptographic algorithm is an asymmetric algorithm which uses two keys for encryption and decryption of plain text. Those two keys are called private key and public key. When the sender wants to send any information then the receiver must publish its public key. So, that sender is able to encrypt the plain text. When the sender sends encrypted text to the receiver then receiver uses its private key for decrypting the encrypted text. In simple word public key must be distributed to the sender and is used for encrypting the plain text and private key must kept secret and only known to the receiver for decrypting the encrypted text (geeksforgeeks, 2020). For encrypting the plain text and decrypting the encrypted text here are some algorithms which must be followed:

Algorithm for encryption of plain text:

Step1: Select two large prime number p and q, where p! =q

Step 2: Calculate the product of selected two prime numbers (i.e. n = p * q)

Step 3: Subtract 1 from the selected two prime numbers and then find the product after subtracting 1 (i.e. phi (n) = ((p - 1) * (q - 1)).

Step 4: Select the value of e in such a way that 1 must be greater than e and e must be smaller than the value obtained from step 2 (i.e. 1<e<n) and the GCD of e and the result obtained from step 3 must be 1 (i.e. GCD (e, phi (n)) =1).

Step 5: Select the value of d in such a way that 1 must be greater than d and d must be smaller than the result obtained from step 3 (i.e. 1<d<phi(n)) and the product of e, d and mod phi(n) must equals to 1 (i.e. e*d*mod phi(n) = 1).

Step 6: Public key is (e, n).

Step 7: Private Key is (d, n).

Working mechanism of RSA algorithm

Here is the example of encryption and decryption of plain text using RSA algorithm.

Find the encrypted and decrypted text using RSA algorithm.

Plain text = T

Solution: Let us assume two prime number as p=3 and q=11.

Now calculating the product of two selected prime numbers.

N = p * q

N = 3 * 11

N = 33

Now, finding the value of phi (n).

Phi (n) = (p - 1) * (q - 1)

Phi (n) = (3 – 1) * (11 - 1)

Phi (n) = 2 * 10

Phi (n) = 20

Now, calculating the value of 1 in such a way that 1<e<20 and GCD (e, 20) = 1

Assuming the value of e as 7.

1 < 7 < 33

And GCD (7, 20) = 1.

Here, when e = 7 it satisfy all the condition. So, e = 7

Also, now calculating the value of d in such a way that 1<d<n and e*d*mod phi (n) = 1

Assuming the value of d as 3.

1 < 3 < 33

7 * 3 * mod 20 = 1

Here, when d=3 it satisfy the condition.

Hence, **n = 33**, **e= 7**, **d = 3**

So, **Public key = (e, n) = (7, 33)**

And **Private Key = (d, n) = (3, 33)**

Now, encrypting the plain text T. For encrypting the plain Text, we have to write the position of alphabet. Here, the position of alphabet is 19.

So, encrypting the plain text using formula **c = m^e mod n**.

C = 19^7 * mod 33.

C = 13.

Now, changing the value of C into the alphabet according to their position.

Here, encryption of plain text = N.

Now, decrypting the encryption result using formula **m = c^d mod n.**

M = 13^3 * mod 33

M = 19

Now, changing the value of M into the alphabets according to their position.

Here, the original message is T.

Hence, by following the rules we generally encrypt the plain text and decipher the encrypted text.

3.1. Algorithm of newly developed cryptographic system

3.1.1. Algorithm for Encryption

```
Step 1: Start
Step 2: Enter Keyword (i.e. Key)
  If, Key contains numbers, then again enter Key
  Else, go to step 3
Step 3: Enter plain text.
Step 4: Make 8*8 matrix table using key
    Step 4.1: If key contain duplicate letter then remove
        Else, fill the matrix table according to the following priority:
            First priority is given to all letters.
            Second priority is given to all numbers.
            Third priority is given to keyword top row keys
            Fourth priority is given to operators
            Fifth priority is given to brackets
```

Sixth priority is given to remaining symbols as shown in table below.

A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 5: Table of 8*8 matrixes

Step 5: Encrypt plain text.

Step 5.1: Encrypt plain text using Playfair cipher.

Step 5.1.1: Plain text is broken into the blocks containing two Letters.

Step 5.1.2: If the letter repeats within the same block then remove one letter from that block and insert a null letter (i.e. X). That removed letter must be inserted into the next block.

Step 5.1.3: If the 2 letters within the block present within the same column then the letter is encrypted by replacing plain text with the immediate below letter.

Step 5.1.4: If the 2 letter with the block present within the same row then the letter is encrypted by replacing plain text with the immediate right letter.

Step 5.1.5: If the 2 letter with the block present within the different column and different row then we have to switch the plain text with the letter located in its own row or column occupied by other texts.

Step 5.2: Encrypt the message obtained from step 5.1

Step 5.2.1: If the length of the result obtained from step 5.1 is not equal to the length of the key. Then repeat the last alphabets till the length equals. Also if length of the key also not equals to the length of the result obtained from step 5.1 then repeat last alphabet till the length equals.

Step 5.2.2: Convert the result obtained from step 5.1 into decimal number according to their position in the matrix table.

Step 5.2.3: Convert the key into the decimal number according to their position in the matrix table.

Step 5.2.4: Convert the result obtained from step 5.2.2 into 6-bit binary number.

Step 5.2.5: Convert the result obtained from step 5.2.3 into 6-bit binary number.

Step 5.2.6: Apply XOR operator between the result obtained from step 5.2.4 and 5.2.5.

Step 5.2.7: Apply 2’s complement to the result obtained from step 5.2.6.

Step 5.2.8: Break the result obtained from step 5.2.7 into blocks. Each block consist of 6-bit binary number and map block from top to bottom into table.

Step 5.2.9: Convert the result obtained from step 5.2.8 into decimal number.

Step 5.2.10: Convert the result obtained from step 5.2.9 into alphabets according to the positions of decimal number in matrix table.

Step 6: Encrypt key.

Step 6.1: Encrypt keyword using RSA algorithm.

Step 6.1.1: Take two prime numbers.

Step 6.1.2: Apply product between the two numbers.

Step 6.1.3: If the result of step 6.1.2 is more than 63 then go to

step 6.1.1.

Step 6.1.3: Subtracting 1 from both the prime number.

Step 6.1.4: Multiplying the result obtained from step 6.1.3, and

storing into N.

Step 6.1.5: Calculating the value of E in such a way that E must be greater than 1, smaller than N and the GCD of E and N must be equals to 1.

Step 6.1.6: Calculate the value of D in such a way that product of D, E, and mod of N must be equals to 1.

Step 6.1.7: Now, Write down the ASCII-code of the key and store it into M.

Step 6.1.8: Calculate the value of C in such a way that the value of 'E' must be power of M and the result is multiplied with mod N.

Step 6.2: Convert the value of C into 6-bit binary number.

Step 6.3: Apply 2's complement into the result obtained from step 6.2.

3.1.2. Algorithm for Decryption

Step 1: Decrypting key

Step 1.1: Decrypt key using RSA algorithm

Step 1.1.1: Apply 2's complement on the encrypted key.

Step 1.1.2: Make block of 6-bit binary number and mapped it into one column.

Step 1.1.3: Convert 6-bit binary number into decimal number.

Step 1.1.4: Apply formula of encryption using private key, and the decimal number obtained from step 1.1.4.

Step 1.1.5: Write down the alphabets according to their position in ASCII.

Step 1.1.6: Remove the repeated letter from obtained alphabets.

Step 1.1.7: Make 8*8 matrix table using key.

Step 1.1.7.1 if key contain duplicate letter then remove

Else, fill the matrix table according to the following priority:

First priority is given to all letters.
Second priority is given to all numbers.
Third priority is given to keyword top row keys
Fourth priority is given to operators
Fifth priority is given to brackets
Sixth priority is given to remaining symbols as shown in table below.

A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 6: Table of 8*8 Matrixes

Step 2: Decrypting encrypted message.

Step 2.1: Decrypting encrypted message using modified Algorithm

Step 2.1.1: If the length of key is not equals to the text then apply step 5.2.1 from encryption process

Step 2.1.2: Convert encrypted text into number format according to their position using matrix table.

Step 2.1.3: Convert obtained result from step 2.1.2 into 6-bit binary number.

Step 2.1.4: Apply 2's complement into the result obtained from step 2.1.3.

Step 2.1.5: Convert the key into number format according to their ASCII-Code.

Step 2.1.5: Convert the result obtained from step 2.1.3 into block of 6-bit binary number.

Step 2.1.6: Make a block of result obtained from step 2.1.4 into 6-Bit binary number.

Step 2.1.7: Apply XOR operator between the result obtained from step 2.1.5 and 2.1.6.

Step 2.1.8: Convert the 6-bit result obtained from step 2.1.7 into decimal number.

Step 2.1.9: Convert the result obtained from step 2.1.8 into alphabets according to their position using matrix table.

Step 2.1.10: Remove repeated alphabets in last and second last place of alphabets and key.

Step 2.2: Decrypting using Playfair algorithm

Step 2.2.1: Plain text is broken into the blocks containing two Letters.

Step 2.2.2: If the letter repeats within the same block then remove one letter from that block and insert a null letter (i.e. X). That removed letter must be inserted into the next block.

Step 2.2.3: If the 2 letters within the block present within the same column then the letter is encrypted by replacing plain text with the immediate below letter.

Step 2.2.4: If the 2 letter with the block present within the same column then the letter is encrypted by replacing plain text with the immediate right letter.

Step 2.2.5: If the 2 letter with the block present within the different column and different row then we have to switch the plain text with the letter located in its own row or column occupied by other texts.

Using the above Algorithm we generally encrypt and decrypt the message.

3.2. Necessary of modification

We have modified the playfair cryptographic algorithm because of the certain disadvantages of it. After modification of selected playfair cryptographic algorithm at least some of the problems are solved. Disadvantages like keys are not encrypted so that attacker may decrypt the encrypted text, the cipher text obtained is not secure as much as it is to be, the cipher text carries as much information as possible which again is against the confidentiality of information, and one of the main problems of playfair cryptographic system is that it violates the rule of CIA (i.e. Confidentiality, Integrality, Availability). Because of the problems mention above we modified the playfair algorithm.

During modification of playfair algorithm we use many logical and mathematical operations such as 2's complement XOR operations as already mentioned above. Now we discuss about the working mechanism of our newly modified playfair cryptographic algorithm.

3.3. Working mechanism of newly created ‘*PlayRS*’ Cryptographic Algorithm

Generally the working mechanism of the newly developed algorithm is not much difficult. Now let’s talk about the ***PlayRS*** cryptographic algorithms. It is a symmetric cryptographic algorithm which uses similar key while encryption and decryption of data. While giving the plain text and key, the key must be alphabet and plain text must be numeric values, special symbols, or alphabets. Which means it is capable of encrypting both numeric values and alphabets. Talking about the working mechanism of the ***PlayRS*** algorithm, the working methodology is similar like playfair cryptographic algorithm but there is small difference. For working of this cryptographic algorithm we have to give both plain text and key whether key must be only in alphabets and plain text must be in alphabets, numeric values, or symbols. Then we have to make 8*8 matric table while giving priorities to alphabets, numbers, keyboard top row symbols, operators, brackets, and at last the remaining spaces are filled by remaining symbols left. After filling the spaces of the 8*8 matrix table the table must look like as shown in figure below:

A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 7: 8*8 matrix table

After filling the matrix table then we have to make block of two plain text and is encrypted using playfair cryptographic algorithm. When the plain text is converted to cipher text with the help of playfair cryptographic algorithm then we have to calculate the size of the key and encrypted text. If the size does not seem equal then we have to repeat the last letter till the size matches. After the size matches then we have to calculate the position of key and encrypted text according to 8*8 matrix table. After the position is calculated of both then we have to convert that position (which is in the number format) into 6-bit binary number. After converting into 6 bit binary number then we have to apply XOR logical operator between the 6 bit binary numbers of key and encrypted text. After applying XOR logical operator then we have to apply 2's complement into the result obtained from XOR logical operator and we have to make a block of 6 bit binary number obtained from the 2's complement. After making block of 6 bit binary number then we have to convert those 6 bit binary number into decimal number. Now the final result obtained is changed to alphabets or symbols according to 8*8 matrix table. This obtained result is the main cipher text of the plain text.

Now the second process of encryption is to encrypt the plain key. For encrypting key we use RSA algorithm for encryption. While entering two prime numbers in RSA algorithms if the product exceeds 63 then user again have to reenter. When encrypting key have to convert key into ASCII value. Then that numeric value of fed into the encryption formula. When the numeric value is come after encryption of key then numeric value of converted to 6 bit binary number. Now at last the 6 bit binary number is fed into the 2's complement for final encryption of keys.

This is the detail explanation for encryption of plain text and keys using ***PlayRS*** Cryptographic algorithms.

Now for decryption of the encrypted message first we have to decrypt the key. We decrypt the key by applying 2's complement on the binary bit of the encrypted key. After applying 2's complement we make a block of 6 bit binary number. Then the block of 6 bit binary number is converted to decimal number.

After converting to decimal number then the result of decimal number is fed into the decryption formula with private key. Then the result come after using decryption formula is converted to alphabets according to their position using ASCII code. Once the alphabets are discovered then they must kept in 8*8 matrix table as shown in the above table.

After decryption of key now we have to decrypt the plain text. For decrypting encrypted text first we have to calculate the length of the encrypted text and key. Is the size does not seems equal then we have to repeat last letter till the size equal. After calculating size then we have to convert encrypted text to number according to their position using matrix table. Then the number is converted to 6 bit binary number. After converting to 6 bit binary number then 2's complement is applied to the result and block of 6 bit binary number is created. Then key is also converted to number according to their position and then converted to 6 bit binary number. After that the XOR logical operator is used against the result obtained from 2's complement and separated in a block of 6 bit binary number, and the result obtained from key after converting it to 6 bit binary number. When the result is obtained from XOR then 6 bit binary number is converted to decimal number. After that the decimal number is converted to letters according to their position using 8*8 matrix table. If the last letter of obtained text repeats frequently then we have to remove it. And at last the letter obtained is decrypted using playfair cryptographic system.

In this way we decrypt the encrypted text and key.

3.4. Flowchart

The flowchart is simply a pictorial representation of an algorithm. We do a flowchart for a better understanding of the program. Here given below is the flowchart of the above algorithm.

3.4.1. Flowchart for Encryption

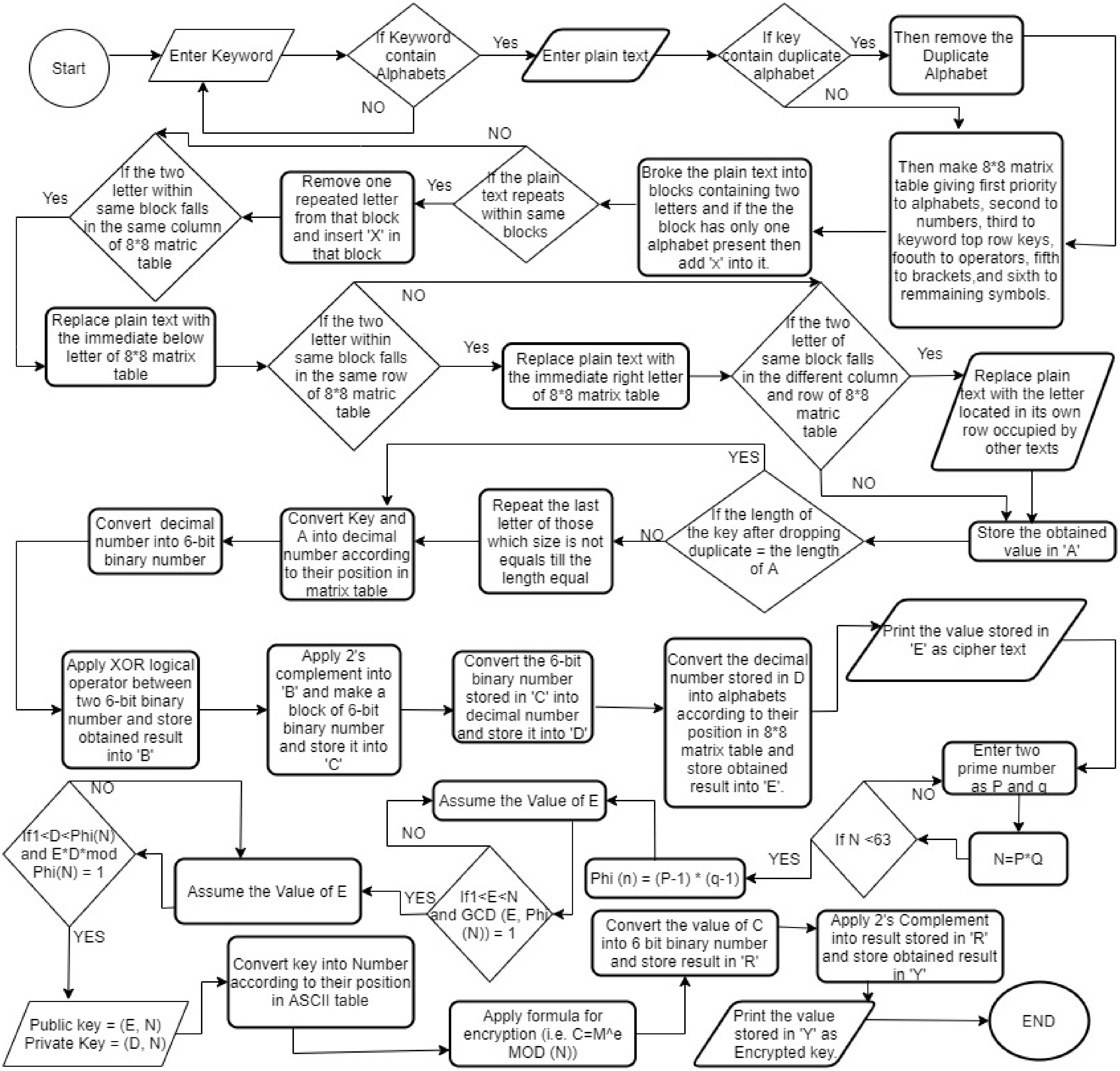

Start
Enter Keyword
If Keyword contain Alphabets
Yes
NO
Enter plain text
If key contain duplicate alphabet
Yes
NO
Then remove the Duplicate Alphabet
Then make 8*8 matrix table giving first priority to alphabets, second to numbers, third to keyword top row keys, foouth to operators, fifth to brackets,and sixth to remmaining symbols.
Broke the plain text into blocks containing two letters and if the the block has only one alphabet present then add 'x' into it.
If the plain text repeats within same blocks
NO
Yes
Remove one repeated letter from that block and insert 'X' in that block
If the two letter within same block falls in the same column of 8*8 matric table
Yes
Replace plain text with the immediate below letter of 8*8 matrix table
If the two letter within same block falls in the same row of 8*8 matric table
NO
Yes
Replace plain text with the immediate right letter of 8*8 matrix table
If the two letter of same block falls in the different column and row of 8*8 matric table
Yes
NO
Replace plain text with the letter located in its own row occupied by other texts
Store the obtained value in 'A'
If the length of the key after dropping duplicate = the length of A
YES
NO
Repeat the last letter of those which size is not equals till the length equal
Convert Key and A into decimal number according to their position in matrix table
Convert decimal number into 6-bit binary number
Apply XOR logical operator between two 6-bit binary number and store obtained result into 'B'
Apply 2's complement into 'B' and make a block of 6-bit binary number and store it into 'C'
Convert the 6-bit binary number stored in 'C' into decimal number and store it into 'D'
Convert the decimal number stored in D into alphabets according to their position in 8*8 matrix table and store obtained result into 'E'.
Print the value stored in 'E' as cipher text
Enter two prime number as P and q
N=P*Q
If N <63
NO
YES
Phi (n) = (P-1) * (q-1)
Assume the Value of E
NO
If1<E<N and GCD (E, Phi (N)) = 1
YES
Assume the Value of E
If1<D<Phi(N) and E*D*mod Phi(N) = 1
NO
YES
Public key = (E, N)
Private Key = (D, N)
Convert key into Number according to their position in ASCII table
Apply formula for encryption (i.e. C=M^e MOD (N))
Convert the value of C into 6 bit binary number and store result in 'R'
Apply 2's Complement into result stored in 'R' and store obtained result in 'Y'
Print the value stored in 'Y' as Encrypted key.
END

Figure 10: Flowchart for Encryption Process

3.4.2. Flowchart for Decryption

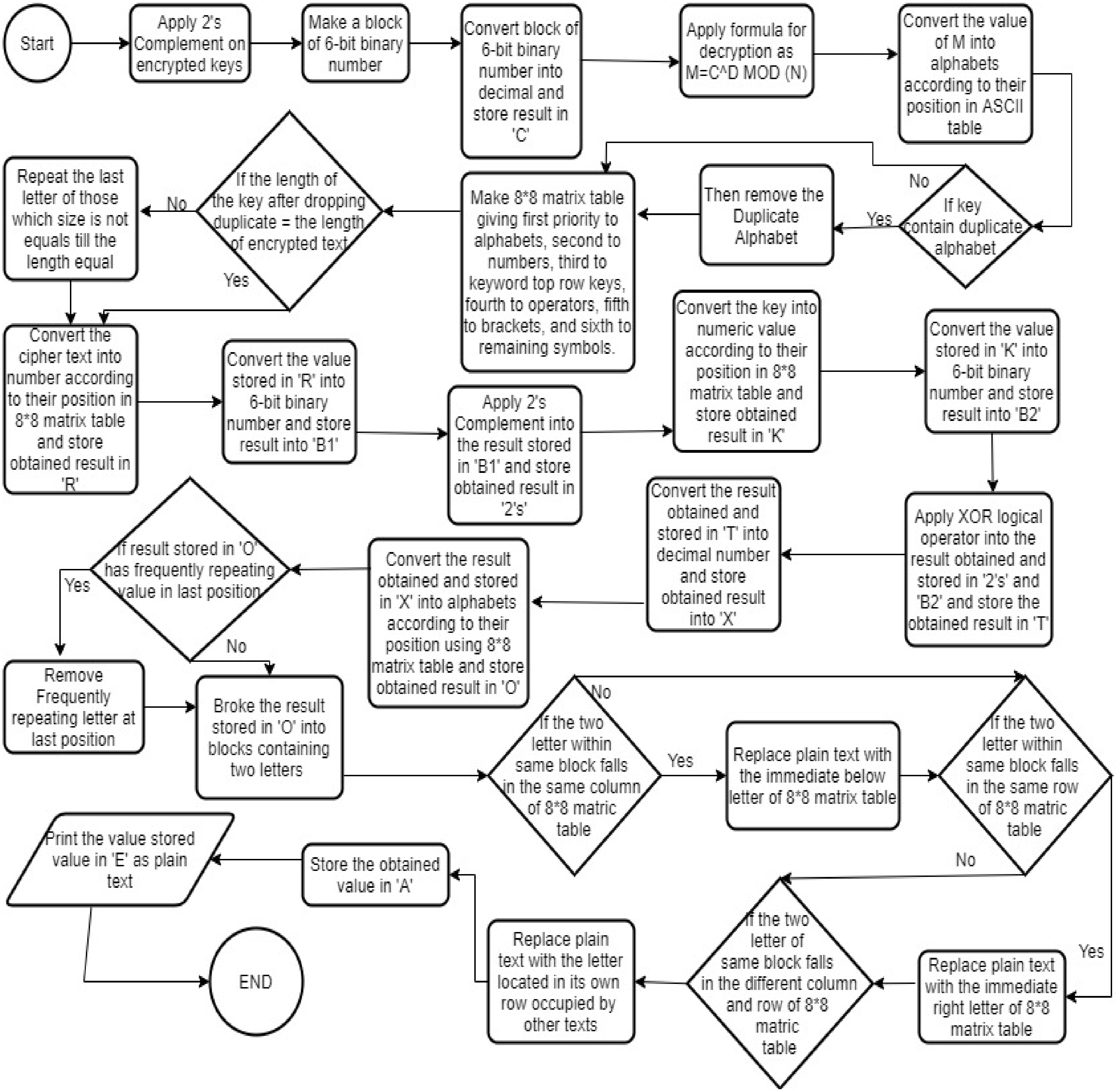

Start
Apply 2's Complement on encrypted keys
Make a block of 6-bit binary number
Convert block of 6-bit binary number into decimal and store result in 'C'
Apply formula for decryption as M=C^D MOD (N)
Convert the value of M into alphabets according to their position in ASCII table
If key contain duplicate alphabet
No
Yes
Then remove the Duplicate Alphabet
Make 8*8 matrix table giving first priority to alphabets, second to numbers, third to keyword top row keys, fourth to operators, fifth to brackets, and sixth to remaining symbols.
If the length of the key after dropping duplicate = the length of encrypted text
No
Yes
Repeat the last letter of those which size is not equals till the length equal
Convert the cipher text into number according to their position in 8*8 matrix table and store obtained result in 'R'
Convert the value stored in 'R' into 6-bit binary number and store result into 'B1'
Apply 2's Complement into the result stored in 'B1' and store obtained result in '2's'
Convert the key into numeric value according to their position in 8*8 matrix table and store obtained result in 'K'
Convert the value stored in 'K' into 6-bit binary number and store result into 'B2'
Apply XOR logical operator into the result obtained and stored in '2's' and 'B2' and store the obtained result in 'T'
Convert the result obtained and stored in 'T' into decimal number and store obtained result into 'X'
Convert the result obtained and stored in 'X' into alphabets according to their position using 8*8 matrix table and store obtained result in 'O'
If result stored in 'O' has frequently repeating value in last position
Yes
No
Remove Frequently repeating letter at last position
Broke the result stored in 'O' into blocks containing two letters
If the two letter within same block falls in the same column of 8*8 matric table
No
Yes
Replace plain text with the immediate below letter of 8*8 matrix table
If the two letter within same block falls in the same row of 8*8 matric table
No
Yes
Replace plain text with the immediate right letter of 8*8 matrix table
If the two letter of same block falls in the different column and row of 8*8 matric table
Replace plain text with the letter located in its own row occupied by other texts
Store the obtained value in 'A'
Print the value stored value in 'E' as plain text
END

Figure 11: Flowchart for Decryption Process

4. TESTING

4.1. TEST: 1

Test NO:	1
Objective:	Plain text must be encrypted and decrypted
Action:	❖ Plain text = I AM DEEPAK ❖ Key=MIXEDCIPHER
Expected Result:	Plain text should be converted to cipher text and vice versa.
Actual Result:	Plain text was encrypted and then decrypted
Conclusion:	The test is successful.

Table 8: Table of Test 1

Dropping the duplicates alphabets in key

So, NK=**MIXEDCPHRS**

Now, making 8*8 matrix table

M	I	X	E	D	C	P	H
R	S	A	B	F	G	J	K
L	N	O	Q	T	U	V	W
Y	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 9: Making 8*8 matrix for encryption of testing 1

Now, encrypting the plain text, 2 letter at a time

IA MD EX EP AK

Cipher text after encryption

XS IC DE DH BR

Now, calculating the size of the

Cipher text = 10

N.K = 10

Here, the size of the cipher text and NK is equal. So, it is not necessary to add new alphabets.

Now, converting N.K, cipher text, into decimal number according to their position in matrix table and then converting it into 6 bit binary number. After that implementing XOR operation between two 6 bit binary bits

Cipher Text	**Their position on matrix table**	**B1**	**N.K**	**Their position on matrix table**	**B2**	**XOR**
X	2	000010	M	0	000000	000010
S	9	001001	I	1	000001	001000
I	1	000001	X	2	000010	000011
C	5	000101	E	3	000011	000110
D	4	000100	D	4	000100	000000
E	3	000011	C	5	000101	000110
D	4	000100	P	6	000110	000010
H	7	000111	H	7	000111	000000
B	11	001011	R	8	001000	000011
R	8	001000	S	9	001001	000001

Table 10: Table of doing logical operations in testing 1

Here,

XOR = 000010001000000011000110000000000110000010000000000011000001

Now, applying 2’s complement into XOR and making block of 6-bit binary.

2’s complement = 111101 110111 111100 111001 111111 111001 111101 111111 111100 111111

Now, converting 6 bit binary into decimal number. After that assigning the value of decimal number using matrix table.

2’s complement	In decimal number	Alphabets according to matrix table
111101	61	_
110111	55	:
111100	60	\
111001	57	‘
111111	63	=
111001	57	‘
111101	61	_
111111	63	=
111100	60	\
111111	63	=

Table 11: Table for obtaining cipher text of testing 1

Hence, the final encrypted text is _**:\’=’_=\=**

Now, encrypting key using RSA algorithm

Let us take two prime numbers as p=3 and q=11

Now,

N=p*q

N=3*11

N=33

Again,

Phi (n) = (p-1)*(q-1)

Phi (n) = (3-1)*(11-1)

Phi (n) = 2*10

Phi (n) = 20

Now calculating the value of e in such that

1<e<N

i.e. 1<e<33

Assume, e = 7

So,

1<7<33

And GCD (7, 20) = 1

When e = 7 all condition are satisfied

Again,

D*e*mod phi (n) = 1

D*7*mod 20 = 1

Assume d=3

3*7*mod20 = 1

Also, 1<3<20

Hence, when d=3, condition is satisfied.

So, **n = 33, d = 3, e = 7**

Now, assigning ASCII code into key alphabet. Then assign it into RSA formula (i.e. c = m^e mod (n)) and convert obtained result into 6 bit binary.

key	ASCII-CODE	C=m^e mod (n)	6-bit binary
M	12	12	001100
I	8	2	000010
X	23	23	010111
E	4	16	010000
D	3	9	001001
C	2	29	011101
P	15	27	011011
H	7	28	011100
R	17	8	001000
S	18	6	000110

Table 12: Table of encrypting keys in testing 1

Applying 2’s complement into 6 bit binary

2’s complement is

110011111011010001011111101101000101001001000111101111111010

Hence, Encrypted text is _**:\’=’_=\=**

Encrypted key is **110011111011010001011111101101000101001001000111101111111010**

Decryption process

Applying 2’s complement on the encrypted key and making block of 6 bit binary number.

2’s complement is

001100 000010 010111010000 001001 011101 011011 011100 001000 000110

Converting 2’s complement into decimal number. Then applying private key as (3, 33) using formula m=c^d mod (n). Then the number is converted into the alphabets according to their position

2’s complement	**In decimal**	**M=c^e mod (n)**	**Aplhabets**
001100	12	12	M
000010	2	8	I
010111	23	23	X
010000	16	4	E
001001	9	3	D
011101	29	2	C
011011	27	15	P
011100	28	7	H
001000	8	17	R
000110	6	18	S

Table 13: Table of decrypting keys in testing 1

Hence, the key is **MIXEDCPHRS**

Now, making 8*8 matrix,

M	I	X	E	D	C	P	H
R	S	A	B	F	G	J	K
L	N	O	Q	T	U	V	W
Y	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 14: Making 8*8 matrix for decryption of testing 1

Now, calculating the length of

Key = 10

Cipher text = 10

Hence, the size is equal. So, it is not necessary to add new alphabets.

Now, converting key, cipher text into decimal number according to their position using matrix table. Then converting decimal number into 6 bit binary number.

Cipher Text	Their position on matrix table	B1	N.K	Their position on matrix table	B2
_	61	111101	M	0	000000
:	55	110111	I	1	000001
\	60	111100	X	2	000010
’	57	111001	E	3	000011
=	63	111111	D	4	000100
’	57	111001	C	5	000101
_	61	111101	P	6	000110
=	63	111111	H	7	000111
\	60	111100	R	8	001000
=	63	111111	S	9	001001

Table 15: Table for doing XOR operator in testing 1

Calculating 2’s complement of B1 and making block of 6 bit binary number

2’s complement = 000010 001000 000011 000110 000000 000110 000010 000000 000011 000001

Now, applying XOR between B2 and 2’s complement. Then converting 6 bit binary into decimal number. Then converting decimal number to alphabets according to their position in matrix table.

2’s complement	B2	XOR	In decimal	Alphabets
000010	000000	000010	2	X
001000	000001	001001	9	S
000011	000010	000001	1	I
000110	000011	000101	5	C
000000	000100	000100	4	D
000110	000101	000011	3	E
000010	000110	000100	4	D
000000	000111	000111	7	H
000011	000100	001011	11	B
000001	001001	001000	8	R

Table 16: Table for obtaining cipher text for decryption in testing 1

If two letter at last repeats frequently then remove

In this case no any letter repeats frequently.

So, making block of 2 alphabets

XS IC DE DH BR

Now, applying playfair in above,

IA MD EX EP AK

Hence, plain text = **IAMDEEPAK**

KEY = **MIXEDCPHRS**

4.2. TEST: 2

Test NO:	2
Objective:	Plain text must be encrypted and decrypted
Action:	❖ Plain text = 9868806098 ❖ Key=NUMBER
Expected Result:	Plain text should be converted to cipher text and vice versa.
Actual Result:	Plain text was encrypted and then decrypted
Conclusion:	The test is successful.

Table 17: Table of Test 2

Dropping the duplicates alphabets in key

So, NK=**NUMBER**

Now, making 8*8 matrix table

N	U	M	B	E	R	A	C
D	F	G	H	I	J	K	L
O	P	Q	S	T	V	W	X
Y	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 18: Making 8*8 matrix for encryption of testing 2

Now, encrypting the plain text, 2 letter at a time

98 68 80 60 98

Cipher text after encryption

!9 79 &8 8Y !9

Now, calculating the size of the

Cipher text = 10

N.K = 6

Here, the size of the cipher text and NK is not equal. So, it is necessary to add new alphabets in NK.

So, NK= NUMBERRRRR

Now, converting N.K, cipher text, into decimal number according to their position in matrix table and then converting it into 6 bit binary number. After that implementing XOR operation between two 6 bit binary bits

Cipher Text	Their position on matrix table	B1	N.K	Their position on matrix table	B2	XOR
!	36	100100	N	0	000000	100100
9	35	100011	U	1	000001	100010
7	33	100001	M	2	000010	100011
9	35	100011	B	3	000011	100000
&	42	101010	E	4	000100	101110
8	34	100010	R	5	000101	100111
8	34	100010	R	5	000101	100111
Y	24	011000	R	5	000101	011101
!	36	100100	R	5	000101	100001
9	35	100011	R	5	000101	100110

Table 19: Table of doing logical operations in testing 2

Here,

XOR = 1001001000101000111000001011101001111001110111011000001100110

Now, applying 2’s complement into XOR and making block of 6-bit binary.

2’s complement = 011011 011101 011100 011111 010001 011000 011000 100010 011110 011010

Now, converting 6 bit binary into decimal number. After that assigning the value of decimal number using matrix table.

2’s complement	In decimal number	Alphabets according to matrix table
011011	27	1
011101	29	3
011100	28	2
011111	31	5
010001	17	P
011000	24	Y
011000	24	Y
100010	34	8
011110	30	4
011001	26	0

Table 20: Table for obtaining cipher text of testing 2

Hence, the final encrypted text is **1325PYY840**

Now, encrypting key using RSA algorithm

Let us take two prime numbers as p=5 and q=7

Now,

N=p*q

N=5*7

N=35

Again,

Phi (n) = (p-1)*(q-1)

Phi (n) = (5-1)*(7-1)

Phi (n) = 4*6

Phi (n) = 24

Now calculating the value of e in such that

1<e<N

i.e. 1<e<35

Assume, e = 11

So,

1<11<35

And GCD (11, 24) = 1

When e = 11 all condition are satisfied

Again,

D*e*mod phi (n) = 1

D*11*mod 24 = 1

Assume d=11

11*11*mod24 = 1

Also, 1<11<24

Hence, when d=11, condition is satisfied.

So, **n = 35, d = 11, e = 11**

Now, assigning ASCII code into key alphabet. Then assign it into RSA formula (i.e. c = m^e mod (n)) and convert obtained result into 6 bit binary.

key	ASCII-CODE	C=m^e mod (n)	6-bit binary
N	13	27	011011
U	20	20	010100
M	12	3	000011
B	1	1	000001
E	4	9	001001
R	17	33	100001

Table 21: Table of encrypting keys in testing 2

Applying 2’s complement into 6 bit binary

2’s complement is

100100101011111100111110110110011111

Hence, Encrypted text is **1325PYY840**

Encrypted key is **100100101011111100111110110110011111**

Decryption process

Applying 2’s complement on the encrypted key and making block of 6 bit binary number.

2’s complement is 011011 010100 000011 000001 001001 100001

Converting 2’s complement into decimal number. Then applying private key as (11, 35) using formula m=c^d mod (n). Then the number is converted into the alphabets according to their position

2’s complement	In decimal	M=c^e mod (n)	Aplhabets
011011	27	13	N
010100	20	20	U
000011	3	12	M
000001	1	1	B
001001	9	4	E
100001	33	17	R

Table 22: Table of decrypting keys in testing 2

Hence, the key is ‘***NUMBER***’

Now, making 8*8 matrix,

N	U	M	B	E	R	A	C
D	F	G	H	I	J	K	L
O	P	Q	S	T	V	W	X
Y	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 23: Making 8*8 matrix for decryption of testing 2

Now, calculating the length of

Key = 6

Cipher text = 10

Hence, the size is not equal. So, it is necessary to add new alphabets in NK.

NK = NUMBERRRRR

Now, converting key, cipher text into decimal number according to their position using matrix table. Then converting decimal number into 6 bit binary number.

Cipher Text	Their position on matrix table	B1	N.K	Their position on matrix table	B2
1	27	111101	N	0	000000
3	29	110111	U	1	000001
2	28	111100	M	2	000010
5	31	111001	B	3	000011
P	17	111111	E	4	000100
Y	24	111001	R	5	000101
Y	24	111101	R	5	000101
8	34	111111	R	5	000101
4	30	111100	R	5	000101
0	26	111111	R	5	000101

Table 24: Table for doing XOR operator in testing 2

Calculating 2’s complement of B1 and making block of 6 bit binary number

2’s complement = 100100 100010 100011 100000 101110 100111 100111 011101 100001 100110

Now, applying XOR between B2 and 2’s complement. Then converting 6 bit binary into decimal number. Then converting decimal number to alphabets according to their position in matrix table.

2’s complement	B2	XOR	In decimal	Alphabets
100100	000000	100100	36	!
100010	000001	100011	35	9
100011	000010	100001	33	7
100000	000011	100011	35	9
101110	000100	101010	42	&
100111	000101	100010	34	8
100111	000101	100010	34	8
011101	000101	011000	24	Y
100001	000101	100100	36	!
100110	000101	100011	35	9

Table 25: Table for obtaining cipher text for decryption in testing 2

If two letter at last repeats frequently then remove

In this case no any letter repeats frequently.

So, making block of 2 alphabets

!9 79 &8 8Y !9

Now, applying playfair in above,

98 68 80 60 98

Hence, plain text = **9868806098**

KEY = **NUMBER**

4.3. TEST: 3

Test NO:	3
Objective:	Plain text must be encrypted and decrypted
Action:	❖ Plain text = SURYANSH ❖ Key=TUTORIAL
Expected Result:	Plain text should be converted to cipher text and vice versa.
Actual Result:	Plain text was encrypted and then decrypted
Conclusion:	The test is successful.

Table 26: Table of Test 3

Dropping the duplicates alphabets in key

So, NK=**TUORIAL**

Now, making 8*8 matrix table

T	U	O	R	I	A	L	B
C	D	E	F	G	H	J	K
M	N	P	Q	S	V	W	X
Y	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 27: Making 8*8 matrix for encryption of testing 3

Now, encrypting the plain text, 2 letter at a time

SU RY AN SH

Cipher text after encryption

NI T1 UV VG

Now, calculating the size of the

Cipher text = 8

N.K = 7

Here, the size of the cipher text and NK is not equal. So, it is necessary to add new alphabets in NK.

So, NK= TUORIALL

Now, converting N.K, cipher text, into decimal number according to their position in matrix table and then converting it into 6 bit binary number. After that implementing XOR operation between two 6 bit binary bits.

Cipher Text	Their position on matrix table	B1	N.K	Their position on matrix table	B2	XOR
N	17	010001	T	0	000000	010001
I	4	000100	U	1	000001	000101
T	0	000000	O	2	000010	000010
1	27	011011	R	3	000011	011000
U	1	000001	I	4	000100	000101
V	21	010101	A	5	000101	010000
V	21	010101	L	6	000110	010011
G	12	001100	L	6	000110	001010

Table 28: Table of doing logical operations in testing 3

Here,

XOR = 010001000101000010011000000101010000010011001010

Now, applying 2’s complement into XOR and making block of 6-bit binary.

2’s complement = 101110 111010 111101 100111 111010 101111 101100 110110

Now, converting 6 bit binary into decimal number. After that assigning the value of decimal number using matrix table.

2’s complement	In decimal number	Alphabets according to matrix table
101110	46	-
111010	58	“
111101	61	_
100111	39	$
111010	58	“
101111	47	[
101100	44	*
110110	54	;

Table 29: Table for obtaining cipher text of testing 3

Hence, the final encrypted text is **–“_$“[*;**

Now, encrypting key using RSA algorithm

Let us take two prime numbers as p=11 and q=5

Now,

N=p*q

N=11*5

N=55

Again,

Phi (n) = (p-1)*(q-1)

Phi (n) = (11-1)*(5-1)

Phi (n) = 10*4

Phi (n) = 40

Now calculating the value of e in such that

1<e<N

i.e. 1<e<40

Assume, e = 7

So,

1<7<40

And GCD (7, 40) = 1

When e = 7 all condition are satisfied

Again,

D*e*mod phi (n) = 1

D*7*mod 40 = 1

Assume d=23

23*7*mod 40 = 1

Also, 1<23<40

Hence, when d=23, condition is satisfied.

So, **n = 55, d = 23, e = 7**

Now, assigning ASCII code into key alphabet. Then assign it into RSA formula (i.e. c = m^e mod (n)) and convert obtained result into 6 bit binary.

key	ASCII-CODE	C=m^e mod (n)	6-bit binary
T	19	24	011000
U	20	15	001111
O	14	9	001001
R	17	8	001000
I	8	2	000010
A	0	0	000000
L	11	11	001011

Table 30: Table of encrypting keys in testing 3

Applying 2's complement into 6 bit binary

2's complement is

100111110000110110110111111101111111110101

Hence, Encrypted text is –**"_$"[*;**

Encrypted key is **100111110000110110110111111101111111110101**

Decryption process

Applying 2’s complement on the encrypted key and making block of 6 bit binary number.

2’s complement is 011000 001111 001001 001000 000010 000000 001011

Converting 2’s complement into decimal number. Then applying private key as (11, 35) using formula m=c^d mod (n). Then the number is converted into the alphabets according to their position.

2’s complement	In decimal	M=c^e mod (n)	Aplhabets
011000	24	19	T
001111	15	20	U
001001	9	14	O
001000	8	17	R
000010	2	8	I
000000	0	0	A
001011	11	11	L

Table 31: Table of decrypting keys in testing 3

Hence, the key is **TUORIAL**

Now, making 8*8 matrix,

T	U	O	R	I	A	L	B
C	D	E	F	G	H	J	K
M	N	P	Q	S	V	W	X
Y	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 32: Making 8*8 matrix for decryption of testing 3

Now, calculating the length of

Key = 7

Cipher text = 8

Hence, the size is not equal. So, it is necessary to add new alphabets in NK.

NK = TUORIALL

Now, converting key, cipher text into decimal number according to their position using matrix table. Then converting decimal number into 6 bit binary number.

Cipher Text	Their position on matrix table	B1	N.K	Their position on matrix table	B2
–	46	101110	T	0	000000
“	58	111010	U	1	000001
_	61	111101	O	2	000010
$	39	100111	R	3	000011
“	58	111010	I	4	000100
[	47	101111	A	5	000101
*	44	101100	L	6	000110
;	54	110110	L	6	000110

Table 33: Table for doing XOR operator in testing 3

Calculating 2’s complement of B1 and making block of 6 bit binary number

2’s complement = 010001 000101 000010 011000 000101 010000 010011 001010

Now, applying XOR between B2 and 2’s complement. Then converting 6 bit binary into decimal number. Then converting decimal number to alphabets according to their position in matrix table.

2’s complement	B2	XOR	In decimal	Alphabets
010001	000000	010001	17	N
000101	000001	000100	4	I
000010	000010	000000	0	T
011000	000011	011011	27	1
000101	000100	000001	1	U
010000	000101	010101	21	V
010011	000101	010101	21	V
001010	000101	001100	12	G

Table 34: Table for obtaining cipher text for decryption in testing 3

If two letter at last repeats frequently then remove

In this case no any letter repeats frequently.

So, making block of 2 alphabets

NI T1 UV VG

Now, applying playfair in above,

SU RY AN SH

Hence, plain text = **SURYANSH**

KEY = **TUORIAL**

4.4. TEST: 4

Test NO:	4
Objective:	Plain text must be encrypted and decrypted
Action:	❖ Plain text = ISLINGTON COLLEGE ❖ Key= LONDONMETROPOLITIANUNIVERSITY
Expected Result:	Plain text should be converted to cipher text and vice versa.
Actual Result:	Plain text was encrypted and then decrypted
Conclusion:	The test is successful.

Table 35: Table of Test 4

Dropping the duplicates alphabets in key

So, NK=**LONDMETRPIAUVSY**

Now, making 8*8 matrix table

L	O	N	D	M	E	T	R
P	I	A	U	V	S	Y	B
C	F	G	H	J	K	Q	W
X	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 36: Making 8*8 matrix for encryption of testing 4

Now, encrypting the plain text, 2 letter at a time

IS LI NG TO NC OL LE GE

Cipher text after encryption

AY OP A0 RN LG NO OT KN

Now, calculating the size of the

Cipher text = 16

N.K = 15

Here, the size of the cipher text and NK is not equal. So, it is necessary to add new alphabets in NK.

So, NK= LONDMETRPIAUVSYY

Now, converting N.K, cipher text, into decimal number according to their position in matrix table and then converting it into 6 bit binary number. After that implementing XOR operation between two 6 bit binary bits.

Cipher Text	Their position on matrix table	B1	N.K	Their position on matrix table	B2	XOR
A	10	001010	L	0	000000	001010
Y	14	001110	O	1	000001	001111
O	1	000001	N	2	000010	000011
P	8	001000	D	3	000011	001011
A	10	001010	M	4	000100	001110
0	26	011010	E	5	000101	011111
R	7	000111	T	6	000110	000001
N	2	000010	R	7	000111	000101
L	0	000000	P	8	001000	001000
G	18	010010	I	9	001001	011011
N	2	000010	A	10	001010	001000
O	1	000001	U	11	001011	001010
O	1	000001	V	12	001100	001101
T	7	000111	S	13	001101	001010
K	21	010101	Y	14	001110	011011
N	2	000010	Y	14	001110	001100

Table 37: Table of doing logical operations in test 4

Here,

XOR = 0010100011110000110010110011100111110000010001010010000110110010

00001010001101001010011011001100

Now, applying 2’s complement into XOR and making block of 6-bit binary.

2’s complement = 110101 110000 111100 110100 110001 100000 111110 111010 110111 100100 110111 110101 110010 110101 100100 110100

Now, converting 6 bit binary into decimal number. After that assigning the value of decimal number using matrix table.

2’s complement	In decimal number	Alphabets according to matrix table
110101	53	.
110000	48	]
111100	60	\
110100	52	)
110001	49	{
100000	32	6
111110	62	£
111010	58	“
110111	55	:
100100	36	!
110111	55	:
110101	53	.
110010	50	}
110101	53	.
100100	36	!
110100	52	)

Table 38: Table for obtaining cipher text of testing 4

Hence, the final encrypted text is **.]\){6£”:!:.}.!)**

Now, encrypting key using RSA algorithm

Let us take two prime numbers as p=3 and q=17

Now,

N=p*q

N=3*17

N=51

Again,

Phi (n) = (p-1)*(q-1)

Phi (n) = (3-1)*(17-1)

Phi (n) = 2*16

Phi (n) = 32

Now calculating the value of e in such that

1<e<N

i.e. 1<e<51

Assume, e = 11

So,

1<11<51

And GCD (11, 32) = 1

When e = 11 all condition are satisfied

Again,

D*e*mod phi (n) = 1

D*11*mod 32 = 1

Assume d=3

3*11*mod 32 = 1

Also, 1<3<32

Hence, when d=3, condition is satisfied.

So, **n = 51, d = 3, e = 11**

Now, assigning ASCII code into key alphabet. Then assign it into RSA formula (i.e. c = m^e mod (n)) and convert obtained result into 6 bit binary.

key	ASCII-CODE	C=m^e mod (n)	6-bit binary
L	11	29	011101
O	14	44	101100
N	13	4	000100
D	3	24	011000
M	12	6	000110
E	4	13	001101
T	19	25	011001
R	17	17	010001
P	15	9	001001
I	8	2	000010
A	0	0	000000
U	20	41	101001
V	21	30	011110
S	18	18	010010
Y	24	48	110000

Table 39: Table of encrypting keys in testing 4

Applying 2’s complement into 6 bit binary

2’s complement is

10001001001111101110011111100111001010011010111011011011110111111110101
10100001101101010000

Hence, Encrypted text is **.]\){6£”:!:.}.!)**

Encrypted key is **10001001001111101110011111100111001010011010111011**

0110111101111111010110100001101101010000

Decryption process

Applying 2’s complement on the encrypted key and making block of 6 bit binary number.

2’s complement is

011101 101100 000100 011000 000110 001101 011001 010001 001001 000010 000000 101001 011110 010010 110000

Converting 2’s complement into decimal number. Then applying private key as (31, 57) using formula m=c^d mod (n). Then the number is converted into the alphabets according to their position

2’s complement	In decimal	M=c^e mod (n)	Alphabets
011101	29	11	L
101100	44	14	O
000100	4	13	N
011000	24	3	D
000110	6	12	M
001101	13	4	E
011001	25	19	T
010001	17	17	R
001001	9	15	P
000010	2	8	I
000000	0	0	A
101001	41	20	U
011110	30	21	V
010010	18	18	S
110000	48	24	Y

Table 40: Table of decrypting keys in testing 4

Hence, the key is **LONDMETRPIAUVSY**

Now, making 8*8 matrix,

L	O	N	D	M	E	T	R
P	I	A	U	V	S	Y	B
C	F	G	H	J	K	Q	W
X	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 41: Making 8*8 matrix for decryption of testing 4

Now, calculating the length of

Key = 15

Cipher text = 16

Hence, the size is not equal. So, it is necessary to add new alphabets in NK.

NK = LONDMETRPIAUVSYY

Now, converting key, cipher text into decimal number according to their position using matrix table. Then converting decimal number into 6 bit binary number.

Cipher Text	Their position on matrix table	B1	N.K	Their position on matrix table	B2
.	53	110101	L	0	000000
]	48	110000	O	1	000001
\	60	111100	N	2	000010
)	52	110100	D	3	000011
{	49	110001	M	4	000100
6	32	100000	E	5	000101
£	62	111110	T	6	000110
”	58	111010	R	7	000111
:	55	110111	P	8	001000
!	36	100100	I	9	001001
:	55	110111	A	10	001010
.	53	110101	U	11	001011
}	50	110010	V	12	001100
.	53	110101	S	13	001101
!	36	100100	Y	14	001110
)	52	110100	Y	14	001110

Table 42: Table for doing XOR operator in testing 4

Calculating 2’s complement of B1 and making block of 6 bit binary number

2’s complement = 001010 001111 000011 001011 001110 011111 000001 000101 001000 011011 001000 001010 001101 001010 011011 001100

Now, applying XOR between B2 and 2’s complement. Then converting 6 bit binary into decimal number. Then converting decimal number to alphabets according to their position in matrix table.

2’s complement	B2	XOR	In decimal	Alphabets
001010	000000	001010	10	A
001111	000001	001110	14	Y
000011	000010	000001	1	O
001011	000011	001000	8	P
001110	000100	001010	10	A
011111	000101	011010	26	0
000001	000110	000111	7	R
000101	000111	000010	2	N
001000	001000	000000	0	L
011011	001001	010010	18	G
001000	001010	000010	2	N
001010	001011	000001	1	O
001101	001100	000001	1	O
001010	001101	000111	7	T
011011	001110	010101	21	K
001100	001110	000010	2	N

Table 43: Table for obtaining cipher text for decryption in testing 4

If two letter at last repeats frequently then remove

In this case no any letter repeats frequently.

So, making block of 2 alphabets

AY OP A0 RN LG NO OT KN

Now, applying playfair in above,

IS LI NG TO NC OL LE GE

Hence, plain text = **ISLINGTONCOLLEGE**

KEY = **LONDMETRPIAUVSY**

4.5. TEST: 5

Test NO:	5
Objective:	Plain text must be encrypted and then decrypted
Action:	❖ Plain text = DEEPAKBOKATI87@GMAIL.COM ❖ Key=MYMAILINGADDRESS
Expected Result:	Plain text should be converted to cipher text and vice versa.
Actual Result:	Plain text was encrypted and then decrypted
Conclusion:	The test is successful.

Table 44: Table of Test 5

Dropping the duplicates alphabets in key

So, NK=**MYAILNGDRES**

Now, making 8*8 matrix table

M	Y	A	I	L	N	G	D
R	E	S	B	C	F	H	J
K	O	P	Q	T	U	V	W
Y	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 45: Making 8*8 matrix for encryption of testing 5

Now, encrypting the plain text, 2 letter at a time

DE EP AK BO KA TI 87 @G MA IL .C OM

Cipher text after encryption

JY SO MP EQ PM QL 78 #N YI LN)F KY

Now, calculating the size of the

Cipher text = 24

N.K = 11

Here, the size of the cipher text and NK is not equal. So, it is necessary to add new alphabets in NK.

So, NK= MYAILNGDRESSSSSSSSSSSSSS

Now, converting N.K, cipher text, into decimal number according to their position in matrix table and then converting it into 6 bit binary number. After that implementing XOR operation between two 6 bit binary bits

Cipher Text	Their position on matrix table	B1	N.K	Their position on matrix table	B2	XOR
J	15	001111	M	0	000000	001111
Y	1	000001	Y	1	000001	000000
S	10	001010	A	2	000010	001000
O	17	010001	I	3	000011	010010
M	0	000000	L	4	000100	000100
P	18	010010	N	5	000101	010111
E	9	001001	G	6	000110	001111
Q	19	010011	D	7	000111	010100
P	18	010010	R	8	001000	011010
M	0	000000	E	9	001001	001001
Q	19	010011	S	10	001010	011001
L	4	000100	S	10	001010	001110
7	33	100001	S	10	001010	101011
8	34	100010	S	10	001010	101000
#	38	100110	S	10	001010	101100
N	5	000101	S	10	001010	001111
Y	1	000001	S	10	001010	001011
I	3	000011	S	10	001010	001001
L	4	000100	S	10	001010	001110
N	5	000101	S	10	001010	001111
)	52	110100	S	10	001010	111110
F	13	001101	S	10	001010	000111
K	16	010000	S	10	001010	011010
Y	1	000001	S	10	001010	001011

Table 46: Table of doing logical operations in test 5

Here,

XOR = 0011110000000010000100100001000101110011110101000110100010010110

0100111010101110100010110000111100101100100100111000111111111000
0111011010001011

Now, applying 2’s complement into XOR and making block of 6-bit binary.

2’s complement = 110000 111111 110111 101101 111011 101000 110000 101011 100101 110110 100110 110001 010100 010111 010011 110000 110100 110110 110001 110000 000001 111000 100101 110101

Now, converting 6 bit binary into decimal number. After that assigning the value of decimal number using matrix table.

2’s complement	In decimal number	Alphabets according to matrix table
110000	48	]
111111	63	=
110111	55	:
101101	45	+
111011	59	?
101000	40	‘
110000	48	]
101011	43	/
100101	37	@
110110	54	,
100110	38	#
110001	49	{
010100	20	T
010111	23	W
010011	19	Q
110000	48	]
110100	52	)
110110	54	,
110001	49	{
110000	48	]
000001	1	Y
111000	56	;
100101	37	@
110101	52	)

Table 47: Table for obtaining cipher text of testing 5

Hence, the final encrypted text is **]=:+?’]/@,#{TWQ]),{]Y;@)**

Now, encrypting key using RSA algorithm

Let us take two prime numbers as $p=3$ and $q=19$

Now,

$N=p*q$

$N=3*19$

$N=57$

Again,

$\text{Phi } (n) = (p-1)*(q-1)$

$\text{Phi } (n) = (3-1)*(19-1)$

$\text{Phi } (n) = 2*18$

$\text{Phi } (n) = 36$

Now calculating the value of e in such that

$1<e<N$

i.e. $1<e<57$

Assume, $e = 7$

So,

$1<7<57$

And GCD (7, 36) = 1

When $e = 7$ all condition are satisfied

Again,

D*e*mod phi (n) = 1

D*7*mod 36 = 1

Assume d=31

1*7*mod 40 = 1

Also, 1<31<40

Hence, when d=31, condition is satisfied.

So, **n = 57, d = 31, e = 7**

Now, assigning ASCII code into key alphabet. Then assign it into RSA formula (i.e. c = m^e mod (n)) and convert obtained result into 6 bit binary.

key	ASCII-CODE	C=m^e mod (n)	6-bit binary
M	12	12	001100
Y	24	54	110110
A	0	0	000000
I	8	8	001000
L	11	11	001011
N	13	10	001010
G	6	9	001001
D	3	21	010101
R	17	5	000101
E	4	25	011001
S	18	18	010010

Table 48: Table of encrypting keys in testing 5

Applying 2’s complement into 6 bit binary

2’s complement is

11001100100111111111110111110100110101110110101010111010100110101110

Hence, Encrypted text is **]=:+?’]/@,#{TWQ]),{]Y;@)**

Encrypted key is **11001100100111111111110111110100110101110110101010**

111010100110101110

Decryption process

Applying 2’s complement on the encrypted key and making block of 6 bit binary number.

2’s complement is

001100 110110 000000 001000 001011 001010 001001 010101 000101 011001 010010

Converting 2’s complement into decimal number. Then applying private key as (31, 57) using formula m=c^d mod (n). Then the number is converted into the alphabets according to their position

2’s complement	In decimal	M=c^e mod (n)	Aplhabets
001100	12	12	M
110110	54	24	Y
000000	0	0	A
001000	8	8	I
001011	11	11	L
001010	10	13	N
001001	9	6	G
010101	21	3	D
000101	5	17	R
011001	25	4	E
010010	18	18	S

Table 49: Table of decrypting keys in testing 5

Hence, the key is **MYAILNGDRES**

Now, making 8*8 matrix,

M	Y	A	I	L	N	G	D
R	E	S	B	C	F	H	J
K	O	P	Q	T	U	V	W
Y	Z	0	1	2	3	4	5
6	7	8	9	!	@	#	$
%	^	&	/	*	+	-	[
]	{	}	(	)	.	,	:
;	‘	“	?	\	_	£	=

Table 50: Making 8*8 matrix for decryption of testing 5

Now, calculating the length of

Key = 11

Cipher text = 24

Hence, the size is not equal. So, it is necessary to add new alphabets in NK.

NK = MYAILNGDRESSSSSSSSSSSSSS

Now, converting key, cipher text into decimal number according to their position using matrix table. Then converting decimal number into 6 bit binary number.

Cipher Text	Their position on matrix table	B1	N.K	Their position on matrix table	B2
]	48	110000	M	0	000000
=	63	111111	Y	1	000001
:	55	110111	A	2	000010
+	45	101101	I	3	000011
?	59	111011	L	4	000100
’	40	101000	N	5	000101
]	48	110000	G	6	000110
/	43	101011	D	7	000111
@	37	100101	R	8	001000
,	54	110110	E	9	001001
#	38	100110	S	10	001010
{	49	110001	S	10	001010
T	20	010100	S	10	001010
W	23	010111	S	10	001010
Q	19	010011	S	10	001010
]	48	110000	S	10	001010
)	52	110100	S	10	001010
,	54	110110	S	10	001010
{	49	110001	S	10	001010
]	48	110000	S	10	001010
Y	1	000001	S	10	001010
;	56	111000	S	10	001010
@	37	100101	S	10	001010
)	52	110101	S	10	001010

Table 51: Table for doing XOR operator in testing 5

Calculating 2’s complement of B1 and making block of 6 bit binary number

2’s complement = 001111 000000 001000 010010 000100 010111 001111 010100 011010 001001 011001 001110 101011 101000 101100 001111 001011 001001 001110 001111 111110 000111 011010 001011

Now, applying XOR between B2 and 2’s complement. Then converting 6 bit binary into decimal number. Then converting decimal number to alphabets according to their position in matrix table.

2’s complement	B2	XOR	In decimal	Alphabets
001111	000000	001111	15	J
000000	000001	000001	1	Y
001000	000010	001010	10	S
010010	000011	010001	17	O
000100	000100	000000	0	M
010111	000101	010010	18	P
001111	000110	001001	9	E
010100	000111	010011	19	Q
011010	001000	010010	18	P
001001	001001	000000	0	M
011001	001010	010011	19	Q
001110	001010	000100	4	L
101011	001010	100001	33	7
101000	001010	100010	34	8
101100	001010	100110	38	#
001111	001010	000101	5	N
001011	001010	000001	1	Y
001001	001010	000011	3	I
001110	001010	000100	4	L
001111	001010	000101	5	N
111110	001010	110100	52	)
000111	001010	001101	13	F
011010	001010	010000	16	K
001011	001010	000001	1	Y

Table 52: Table for obtaining cipher text for decryption in testing 5

If two letter at last repeats frequently then remove

In this case no any letter repeats frequently.

So, making block of 2 alphabets

JY SO MP EQ PM QL 78 #N YI LN)F KY

Now, applying playfair in above,

DE EP AK BO KA TI 87 @G MA IL .C OM

Hence, plain text = **DEEPAKBOKATI87@GMAIL.COM**

KEY = **MYAILNGDRES**

5. Critical Evaluation of the new Cryptographic Algorithm

We have modified Playfair cryptographic algorithm and made similar type of another algorithm known as PlayRS algorithm. That algorithm must be having some positive and negative things. Talking about pros and cons of PlayRS algorithm here is the detail explanation given below:

5.1. Advantages and Disadvantages of PlayRs Cryptographic Algorithm

5.1.1. Advantages of '*PLAYRS'* Algorithm

I. It is stronger enough than other symmetric cryptographic algorithm.

II. It can encrypt all plain text like alphabets, numeric values, and symbols.

III. It maintains, and follows the principle of Confidentiality, integrity, and availability so that it is difficult to break encrypted text.

IV. It uses the public key for encrypting the key and uses a private key for decrypting the encrypted key.

V. It encrypts plain text as well as the key so that for the decryption process first we have to decrypt the key for decrypting encrypted text.

VI. It shows the properties of Avalanche effect, so that it is hard to change the bits of encrypted key.

5.1.2. Disadvantages of *'PLAYRS'* Algorithm

I. It only accepts key having alphabets only.

II. It only accepts either capital alphabets or small alphabets.

III. We are not able to use prime number having product more than 63 while encrypting key.

IV. It does not support certain character like space between two words.

V. Since the algorithm is very strong, but the frequency of plain text and cipher text is identical to each other.

5.2. Application Areas of PlayRs Cryptographic Algorithm

We have talked about the working mechanism, advantages, disadvantages, and more about PlayRS cryptographic algorithm. Now talking about the application areas of PlayRS cryptographic algorithm system, it is used in the banking system, digital certificates, e commerce business and many more. Because we all know that the how this algorithm is modified and made. This algorithm consists of complex algorithm which is much stronger than other algorithms. Because of the complex algorithm, attackers are unable to find the as much as information of the confidential data, so that they are unable to destroy any information of an organization or on enterprises. Now again taking more about the application areas this algorithm can be used by the government offices, in a security system, and many more. In simple word PlayRS cryptographic system can be used in the security filed where the data of organization or enterprises must be kept secure.

6. Conclusion

This is the first course work of Security in computing module which was released on the 5th week of our regular class. This course work was very tough as compared to other subject course work. This course work was about developing new cryptographic algorithm using old algorithm. During development process lots of research is done while completing this work. While doing the course work lots of problems occurred and then solved all the problems that are occurred while developing a program.

Completing this task helped me to learn about the cryptography and working mechanism of cryptographic algorithm, and it also boosts my skills. I am grateful to my module leader, tutor, and my seniors, friends who helped me to develop a well-managed report. To complete this course work on due date, time is managed and all required is research. This course work was very challenging. After my hard work, I got to know more about algorithms.

7. Reference and Bibliography

allaboutcircuits. (2020) *The Exclusive-OR Function: The XOR Gate* [Online]. Available from: https://www.allaboutcircuits.com/textbook/digital/chpt-7/the-exclusive-or-function-xor/ [Accessed 29 November 2020].

Bishop, D. (2003) *Introduction to Cryptography with JAVA APPLETS*. 2nd ed. Sudbury: Jones and Bartlett Publicers.

certmike. (2019) *Confidentiality, Integrity And Availability – The CIA Triad* [Online]. Available from: https://www.certmike.com/confidentiality-integrity-and-availability-the-cia-triad/ [Accessed 05 January 2021].

csoonline. (2020) *The CIA triad: Definition, components and examples* [Online]. Available from: https://www.csoonline.com/article/3519908/the-cia-triad-definition-components-and-examples.html [Accessed 05 January 2021].

csoonline. (2020) *Top cybersecurity facts, figures and statistics for 2020* [Online]. Available from: https://www.csoonline.com/article/3153707/top-cybersecurity-facts-figures-and-statistics.html [Accessed 01 January 2021].

digitalguardian. (2020) *What's the Cost of a Data Breach in 2019?* [Online]. Available from: https://digitalguardian.com/blog/whats-cost-data-breach-2019 [Accessed 01 January 2021].

Dooley, J.F. (2018) *History of Cryptography and Cryptanalysis*. 1st ed. Switzerland: Springer International Publishing.

geeksforgeeks. (2019) *Playfair Cipher with Examples* [Online]. Available from: https://www.geeksforgeeks.org/playfair-cipher-with-examples/ [Accessed 05 December 2020].

geeksforgeeks. (2020) *Data encryption standard (DES)* | *Set 1* [Online]. Available from: https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/ [Accessed 10 January 2021].

geeksforgeeks. (2020) *How to solve RSA Algorithm Problems?* [Online]. Available from: https://www.geeksforgeeks.org/how-to-solve-rsa-algorithm-problems/ [Accessed 04 December 2020].

Gollmann, D. (2011) *COMPUTER SECURITY*. 3rd ed. Chichester, West Sussex, United Kingdom: A John Wiley and Sons, Ltd., Publication.

ibm. (2020) *Security overview* [Online]. Available from: https://www.ibm.com/support/knowledgecenter/SSYJCD_1.0.0/com.ibm.help.meigV100.doc/com.ibm.help.meg.securing.doc/meg_sec_overview.html [Accessed 03 January 2020].

Kaur, A., Verma, H.K. & Singh, R.K. (2013) 3D - Playfair Cipher with additional Bitwise Operation. *2013 International Conference on Control, Computing, Communication and Materials (ICCCCM)*.

P, N.H., N., A.R. & Yadav, N. (2017) Secure Message Transfer using RSA algorithm and Improved Playfair cipher in Cloud Computing. *2017 2nd International Conference for Convergence in Technology (I2CT)*, pp.931-36.

purplesec. (2020) *Cyber security statistics* [Online]. Available from: https://purplesec.us/resources/cyber-security-statistics/ [Accessed 01 January 2021].

Srivastava, S.S., Gupta, N. & Jaiswal, R. (2010) Modified Version of Playfair Cipher by using 8x8 Matrix and Random Number Generation. *2011 3rd International Conference on Computer Modeling and Simulation (ICCMS 2011)*, pp.615-17.

Tamassia, R. & Goodrich, M.T. (2014) *Introduction to Computer Security*. 1st ed. Harlow: Pearson.

techrejects. (2014) *Advantages and Disadvantages of Symmetric and Asymmetric Key Encryption Methods* [Online]. Available from: https://techrejects.blogspot.com/2014/08/advantages-disadvantages-symmetric-asymmetric-key-encryption-methods.html?m=1 [Accessed 03 January 2021].

techtarget. (2020) *confidentiality, integrity, and availability (CIA triad)* [Online]. Available from: https://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA [Accessed 03 January 2021].

tutorialspoint. (2020) *two-s-complement* [Online]. Available from: https://www.tutorialspoint.com/two-s-complement [Accessed 29 November 2020].

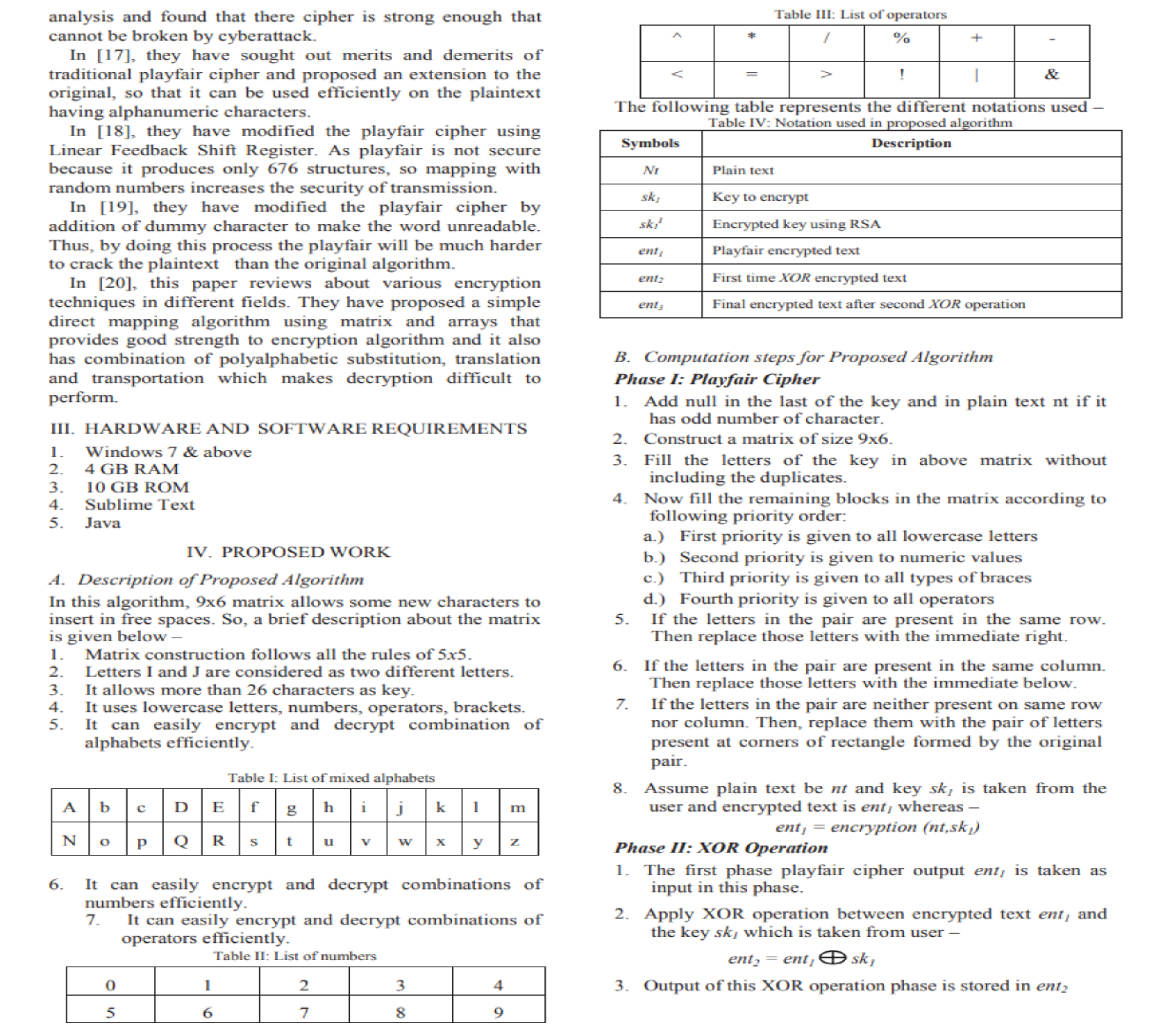
tutorialspoint. (2021) *Cryptography - Quick Guide* [Online]. Available from: https://www.tutorialspoint.com/cryptography/cryptography_quick_guide.htm [Accessed 10 January 2021].

Whitman, M.E. & Mattord, H.J. (2018) *MANAGEMENT OF INFORMATION SECURITY*. 6th ed. Boston, USA: CENGAGE.

Whitman, M.E. & Mattord, H.J. (2018) *Principle of Information Security*. 4th ed. Boston, USA: CENGAGE Learning. Available at: https://www.britannica.com/.

8. Appendix

Here are the some of the screenshot of the journal articles through which I have taken idea while modifying playfair cryptographic algorithm. Screenshot's are given below:



analysis and found that there cipher is strong enough that cannot be broken by cyberattack.

In [17], they have sought out merits and demerits of traditional playfair cipher and proposed an extension to the original, so that it can be used efficiently on the plaintext having alphanumeric characters.

In [18], they have modified the playfair cipher using Linear Feedback Shift Register. As playfair is not secure because it produces only 676 structures, so mapping with random numbers increases the security of transmission.

In [19], they have modified the playfair cipher by addition of dummy character to make the word unreadable. Thus, by doing this process the playfair will be much harder to crack the plaintext than the original algorithm.

In [20], this paper reviews about various encryption techniques in different fields. They have proposed a simple direct mapping algorithm using matrix and arrays that provides good strength to encryption algorithm and it also has combination of polyalphabetic substitution, translation and transportation which makes decryption difficult to perform.

III. HARDWARE AND SOFTWARE REQUIREMENTS

1. Windows 7 & above
2. 4 GB RAM
3. 10 GB ROM
4. Sublime Text
5. Java

IV. PROPOSED WORK

A. Description of Proposed Algorithm

In this algorithm, 9x6 matrix allows some new characters to insert in free spaces. So, a brief description about the matrix is given below –

1. Matrix construction follows all the rules of 5x5.
2. Letters I and J are considered as two different letters.
3. It allows more than 26 characters as key.
4. It uses lowercase letters, numbers, operators, brackets.
5. It can easily encrypt and decrypt combination of alphabets efficiently.

Table I: List of mixed alphabets

A	b	c	D	E	f	g	h	i	j	k	l	m
N	o	p	Q	R	s	t	u	v	w	x	y	z

6. It can easily encrypt and decrypt combinations of numbers efficiently.
7. It can easily encrypt and decrypt combinations of operators efficiently.

Table II: List of numbers

0	1	2	3	4
5	6	7	8	9

Table III: List of operators

^	*	/	%	+	-
<	=	>	!	|	&

The following table represents the different notations used –

Table IV: Notation used in proposed algorithm

Symbols	Description
Nt	Plain text
sk_1	Key to encrypt
sk_1^1	Encrypted key using RSA
ent_1	Playfair encrypted text
ent_2	First time *XOR* encrypted text
ent_3	Final encrypted text after second *XOR* operation

B. Computation steps for Proposed Algorithm

Phase I: Playfair Cipher

1. Add null in the last of the key and in plain text nt if it has odd number of character.
2. Construct a matrix of size 9x6.
3. Fill the letters of the key in above matrix without including the duplicates.
4. Now fill the remaining blocks in the matrix according to following priority order:
   a.) First priority is given to all lowercase letters
   b.) Second priority is given to numeric values
   c.) Third priority is given to all types of braces
   d.) Fourth priority is given to all operators
5. If the letters in the pair are present in the same row. Then replace those letters with the immediate right.
6. If the letters in the pair are present in the same column. Then replace those letters with the immediate below.
7. If the letters in the pair are neither present on same row nor column. Then, replace them with the pair of letters present at corners of rectangle formed by the original pair.
8. Assume plain text be nt and key sk_1 is taken from the user and encrypted text is ent_1 whereas –

$$ent_1 = encryption\ (nt,sk_1)$$

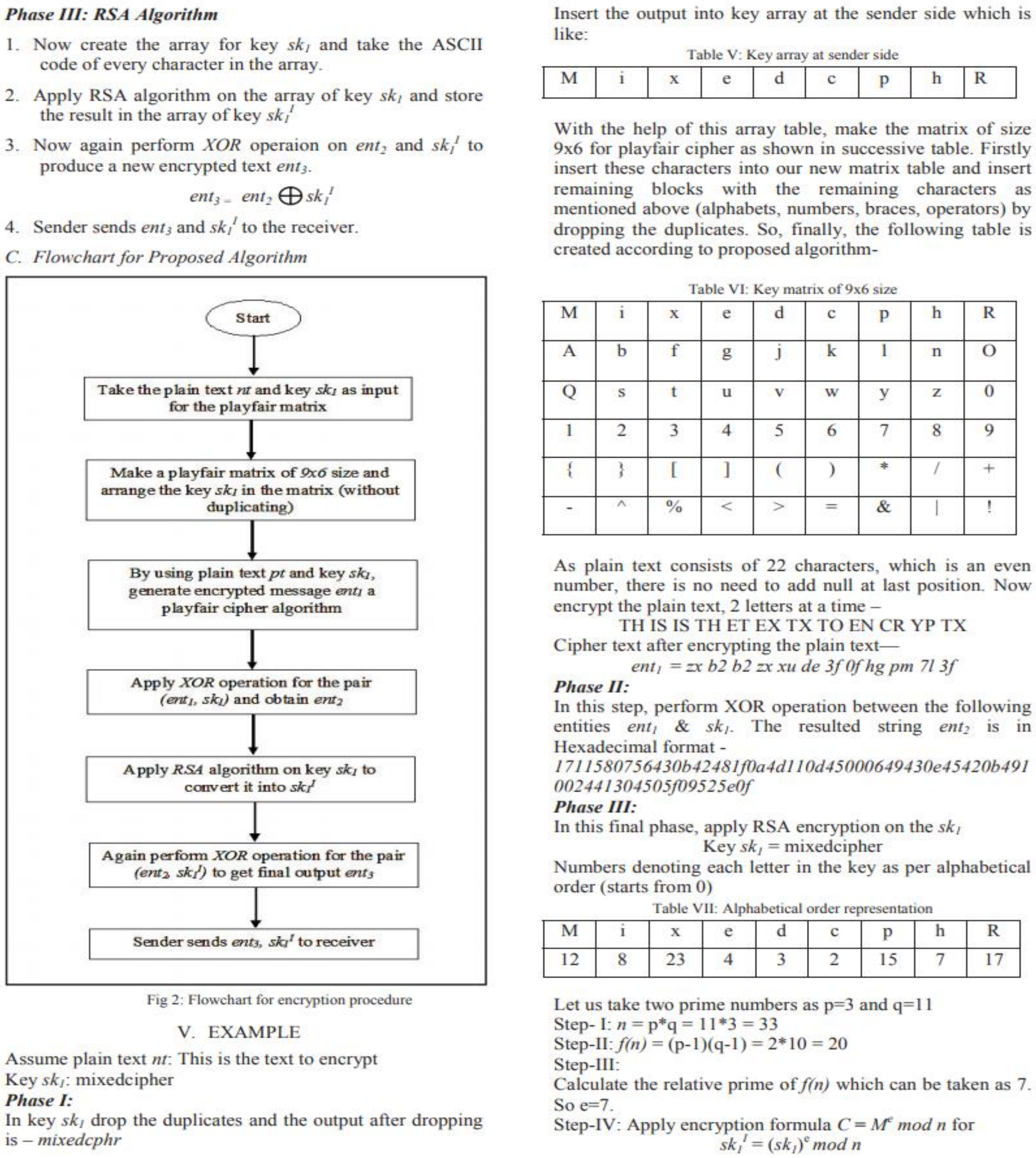
Phase II: XOR Operation

1. The first phase playfair cipher output ent_1 is taken as input in this phase.
2. Apply XOR operation between encrypted text ent_1 and the key sk_1 which is taken from user –

$$ent_2 = ent_1 \oplus sk_1$$

3. Output of this XOR operation phase is stored in ent_2

Figure 12: Screenshot 1 of Appendix (P et al., 2017)



Phase III: RSA Algorithm

1. Now create the array for key sk_1 and take the ASCII code of every character in the array.
2. Apply RSA algorithm on the array of key sk_1 and store the result in the array of key sk_1^I
3. Now again perform *XOR* operaion on ent_2 and sk_1^I to produce a new encrypted text ent_3.

$$ent_3 = ent_2 \oplus sk_1^I$$

4. Sender sends ent_3 and sk_1^I to the receiver.

C. Flowchart for Proposed Algorithm

Start
Take the plain text nt and key sk_1 as input for the playfair matrix
Make a playfair matrix of $9x6$ size and arrange the key sk_1 in the matrix (without duplicating)
By using plain text pt and key sk_1, generate encrypted message ent_1 a playfair cipher algorithm
Apply *XOR* operation for the pair (ent_1, sk_1) and obtain ent_2
Apply *RSA* algorithm on key sk_1 to convert it into sk_1^I
Again perform *XOR* operation for the pair (ent_2, sk_1^I) to get final output ent_3
Sender sends ent_3, sk_1^I to receiver

Fig 2: Flowchart for encryption procedure

V. EXAMPLE

Assume plain text *nt*: This is the text to encrypt
Key sk_1: mixedcipher

Phase I:

In key sk_1 drop the duplicates and the output after dropping is – *mixedcphr*

Insert the output into key array at the sender side which is like:

Table V: Key array at sender side

M	i	x	e	d	c	p	h	R

With the help of this array table, make the matrix of size 9x6 for playfair cipher as shown in successive table. Firstly insert these characters into our new matrix table and insert remaining blocks with the remaining characters as mentioned above (alphabets, numbers, braces, operators) by dropping the duplicates. So, finally, the following table is created according to proposed algorithm-

Table VI: Key matrix of 9x6 size

M	i	x	e	d	c	p	h	R
A	b	f	g	j	k	l	n	O
Q	s	t	u	v	w	y	z	0
1	2	3	4	5	6	7	8	9
{	}	[	]	(	)	*	/	+
-	^	%	<	>	=	&	|	!

As plain text consists of 22 characters, which is an even number, there is no need to add null at last position. Now encrypt the plain text, 2 letters at a time –

TH IS IS TH ET EX TX TO EN CR YP TX

Cipher text after encrypting the plain text—

ent_1 = *zx b2 b2 zx xu de 3f 0f hg pm 7l 3f*

Phase II:

In this step, perform XOR operation between the following entities ent_1 & sk_1. The resulted string ent_2 is in Hexadecimal format -
1711580756430b42481f0a4d110d45000649430e45420b491 002441304505f09525e0f

Phase III:

In this final phase, apply RSA encryption on the sk_1

Key sk_1 = mixedcipher

Numbers denoting each letter in the key as per alphabetical order (starts from 0)

Table VII: Alphabetical order representation

M	i	x	e	d	c	p	h	R
12	8	23	4	3	2	15	7	17

Let us take two prime numbers as p=3 and q=11
Step- I: n = p*q = 11*3 = 33
Step-II: $f(n)$ = (p-1)(q-1) = 2*10 = 20
Step-III:
Calculate the relative prime of $f(n)$ which can be taken as 7. So e=7.
Step-IV: Apply encryption formula $C = M^e \ mod \ n$ for

$$sk_1^I = (sk_1)^e \ mod \ n$$

Figure 13: Screenshot 2 of Appendix (P et al., 2017)

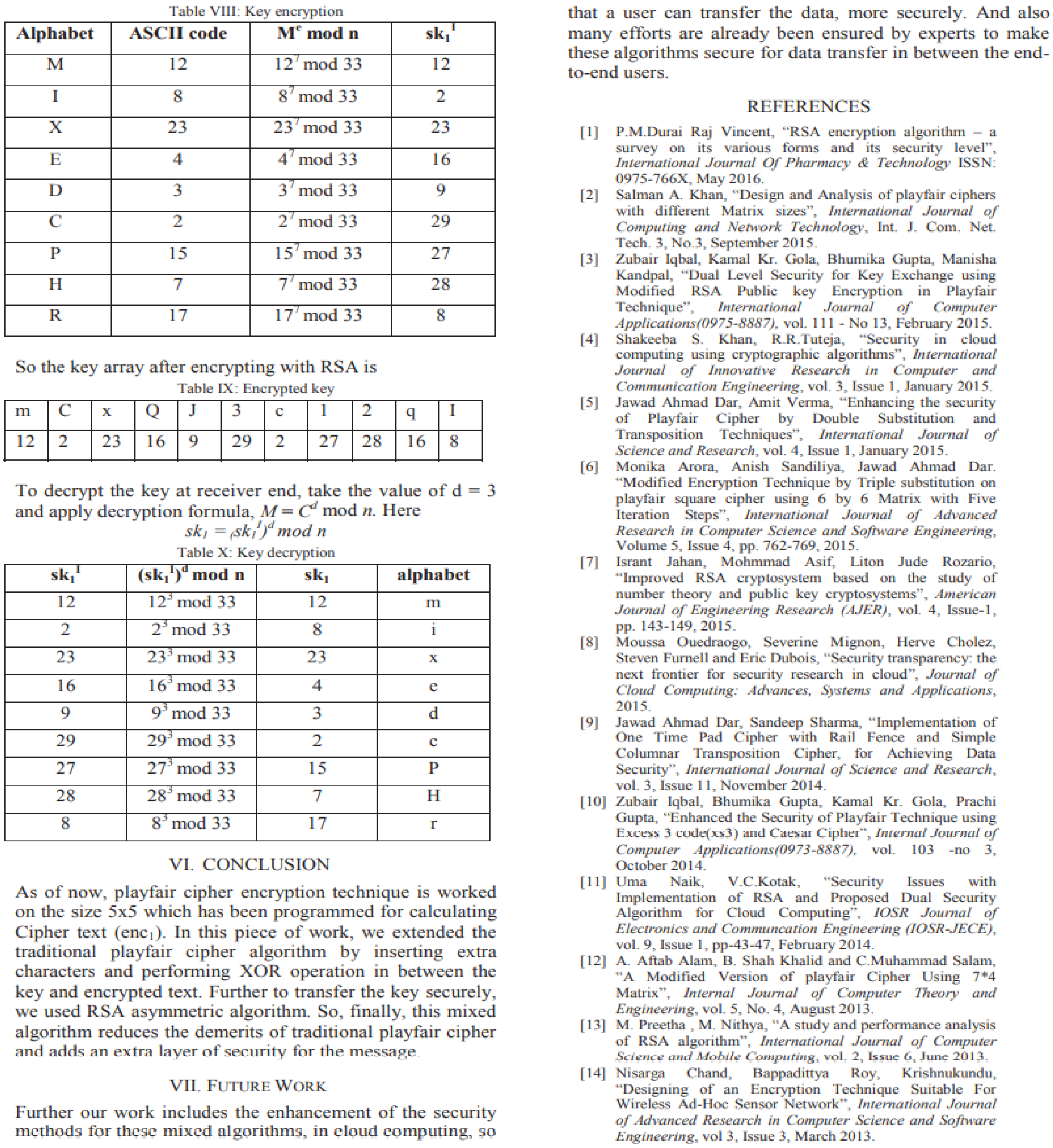


Table VIII: Key encryption

Alphabet	ASCII code	M^e mod n	sk_1^1
M	12	12^7 mod 33	12
I	8	8^7 mod 33	2
X	23	23^7 mod 33	23
E	4	4^7 mod 33	16
D	3	3^7 mod 33	9
C	2	2^7 mod 33	29
P	15	15^7 mod 33	27
H	7	7^7 mod 33	28
R	17	17^7 mod 33	8

So the key array after encrypting with RSA is

Table IX: Encrypted key

m	C	x	Q	J	3	c	1	2	q	I
12	2	23	16	9	29	2	27	28	16	8

To decrypt the key at receiver end, take the value of d = 3 and apply decryption formula, $M = C^d$ mod n. Here

$$sk_1 = (sk_1^1)^d \, mod \, n$$

Table X: Key decryption

sk_1^1	$(sk_1^1)^d$ mod n	sk_1	alphabet
12	12^3 mod 33	12	m
2	2^3 mod 33	8	i
23	23^3 mod 33	23	x
16	16^3 mod 33	4	e
9	9^3 mod 33	3	d
29	29^3 mod 33	2	c
27	27^3 mod 33	15	P
28	28^3 mod 33	7	H
8	8^3 mod 33	17	r

VI. CONCLUSION

As of now, playfair cipher encryption technique is worked on the size 5x5 which has been programmed for calculating Cipher text (enc_1). In this piece of work, we extended the traditional playfair cipher algorithm by inserting extra characters and performing XOR operation in between the key and encrypted text. Further to transfer the key securely, we used RSA asymmetric algorithm. So, finally, this mixed algorithm reduces the demerits of traditional playfair cipher and adds an extra layer of security for the message.

VII. FUTURE WORK

Further our work includes the enhancement of the security methods for these mixed algorithms, in cloud computing, so that a user can transfer the data, more securely. And also many efforts are already been ensured by experts to make these algorithms secure for data transfer in between the end-to-end users.

REFERENCES

[1] P.M.Durai Raj Vincent, "RSA encryption algorithm – a survey on its various forms and its security level", *International Journal Of Pharmacy & Technology* ISSN: 0975-766X, May 2016.

[2] Salman A. Khan, "Design and Analysis of playfair ciphers with different Matrix sizes", *International Journal of Computing and Network Technology*, Int. J. Com. Net. Tech. 3, No.3, September 2015.

[3] Zubair Iqbal, Kamal Kr. Gola, Bhumika Gupta, Manisha Kandpal, "Dual Level Security for Key Exchange using Modified RSA Public key Encryption in Playfair Technique", *International Journal of Computer Applications(0975-8887),* vol. 111 - No 13, February 2015.

[4] Shakeeba S. Khan, R.R.Tuteja, "Security in cloud computing using cryptographic algorithms", *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, Issue 1, January 2015.

[5] Jawad Ahmad Dar, Amit Verma, "Enhancing the security of Playfair Cipher by Double Substitution and Transposition Techniques", *International Journal of Science and Research*, vol. 4, Issue 1, January 2015.

[6] Monika Arora, Anish Sandiliya, Jawad Ahmad Dar. "Modified Encryption Technique by Triple substitution on playfair square cipher using 6 by 6 Matrix with Five Iteration Steps", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 5, Issue 4, pp. 762-769, 2015.

[7] Isrant Jahan, Mohmmad Asif, Liton Jude Rozario, "Improved RSA cryptosystem based on the study of number theory and public key cryptosystems", *American Journal of Engineering Research (AJER)*, vol. 4, Issue-1, pp. 143-149, 2015.

[8] Moussa Ouedraogo, Severine Mignon, Herve Cholez, Steven Furnell and Eric Dubois, "Security transparency: the next frontier for security research in cloud", *Journal of Cloud Computing: Advances, Systems and Applications*, 2015.

[9] Jawad Ahmad Dar, Sandeep Sharma, "Implementation of One Time Pad Cipher with Rail Fence and Simple Columnar Transposition Cipher, for Achieving Data Security", *International Journal of Science and Research*, vol. 3, Issue 11, November 2014.

[10] Zubair Iqbal, Bhumika Gupta, Kamal Kr. Gola, Prachi Gupta, "Enhanced the Security of Playfair Technique using Excess 3 code(xs3) and Caesar Cipher", *Internal Journal of Computer Applications(0973-8887),* vol. 103 -no 3, October 2014.

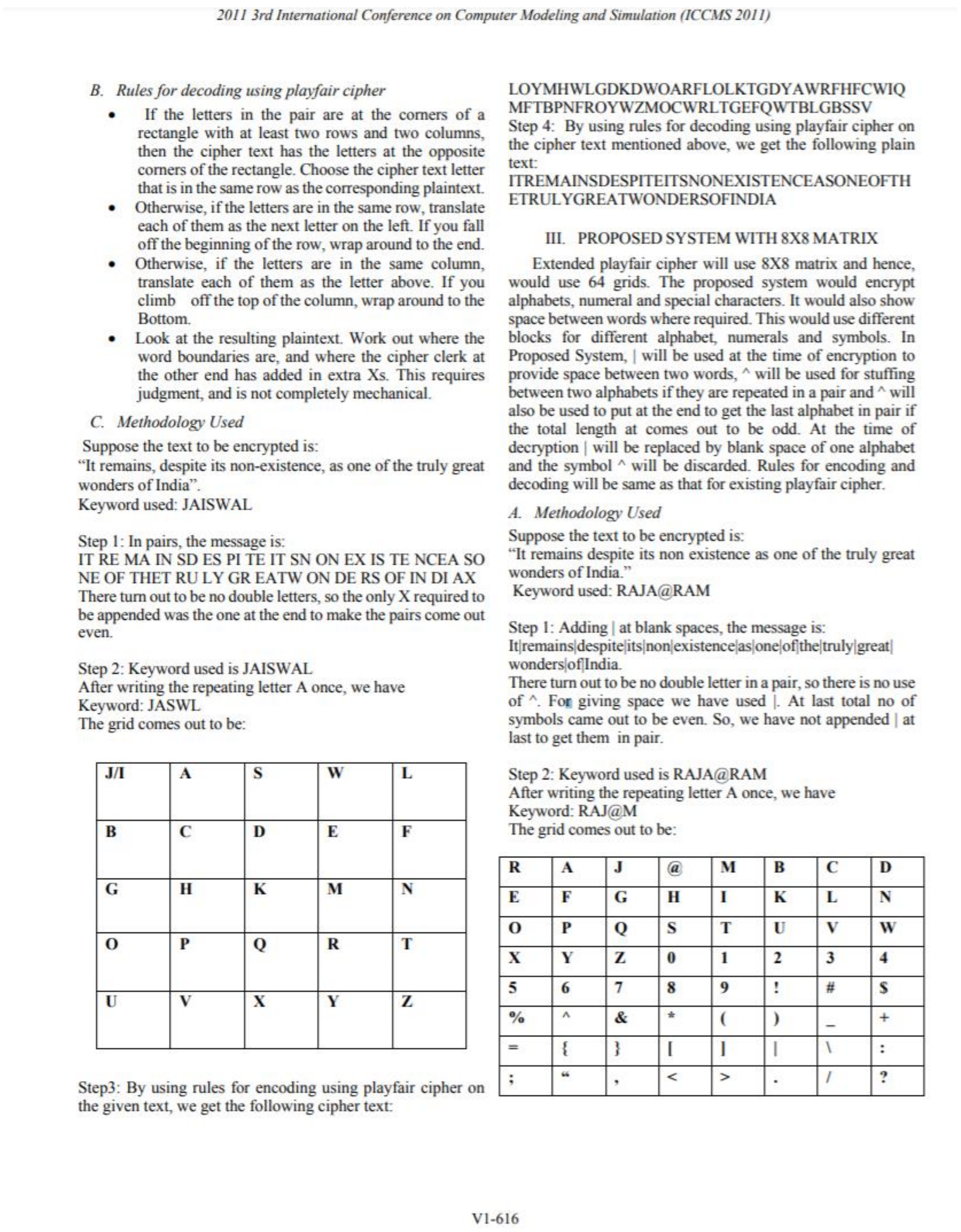
[11] Uma Naik, V.C.Kotak, "Security Issues with Implementation of RSA and Proposed Dual Security Algorithm for Cloud Computing", *IOSR Journal of Electronics and Communcation Engineering (IOSR-JECE)*, vol. 9, Issue 1, pp-43-47, February 2014.

[12] A. Aftab Alam, B. Shah Khalid and C.Muhammad Salam, "A Modified Version of playfair Cipher Using 7*4 Matrix", *Internal Journal of Computer Theory and Engineering*, vol. 5, No. 4, August 2013.

[13] M. Preetha , M. Nithya, "A study and performance analysis of RSA algorithm", *International Journal of Computer Science and Mobile Computing*, vol. 2, Issue 6, June 2013.

[14] Nisarga Chand, Bappadittya Roy, Krishnukundu, "Designing of an Encryption Technique Suitable For Wireless Ad-Hoc Sensor Network", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol 3, Issue 3, March 2013.

Figure 14: Screenshot 3 of Appendix (P et al., 2017)



B. *Rules for decoding using playfair cipher*

- If the letters in the pair are at the corners of a rectangle with at least two rows and two columns, then the cipher text has the letters at the opposite corners of the rectangle. Choose the cipher text letter that is in the same row as the corresponding plaintext.
- Otherwise, if the letters are in the same row, translate each of them as the next letter on the left. If you fall off the beginning of the row, wrap around to the end.
- Otherwise, if the letters are in the same column, translate each of them as the letter above. If you climb off the top of the column, wrap around to the Bottom.
- Look at the resulting plaintext. Work out where the word boundaries are, and where the cipher clerk at the other end has added in extra Xs. This requires judgment, and is not completely mechanical.

C. *Methodology Used*

Suppose the text to be encrypted is:
"It remains, despite its non-existence, as one of the truly great wonders of India".
Keyword used: JAISWAL

Step 1: In pairs, the message is:
IT RE MA IN SD ES PI TE IT SN ON EX IS TE NCEA SO NE OF THET RU LY GR EATW ON DE RS OF IN DI AX
There turn out to be no double letters, so the only X required to be appended was the one at the end to make the pairs come out even.

Step 2: Keyword used is JAISWAL
After writing the repeating letter A once, we have
Keyword: JASWL
The grid comes out to be:

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

Step3: By using rules for encoding using playfair cipher on the given text, we get the following cipher text:
LOYMHWLGDKDWOARFLOLKTGDYAWRFHFCWIQ
MFTBPNFROYWZMOCWRLTGEFQWTBLGBSSV
Step 4: By using rules for decoding using playfair cipher on the cipher text mentioned above, we get the following plain text:
ITREMAINSDESPITEITSNONEXISTENCEASONEOFTH
ETRULYGREATWONDERSOFINDIA

III. PROPOSED SYSTEM WITH 8X8 MATRIX

Extended playfair cipher will use 8X8 matrix and hence, would use 64 grids. The proposed system would encrypt alphabets, numeral and special characters. It would also show space between words where required. This would use different blocks for different alphabet, numerals and symbols. In Proposed System, | will be used at the time of encryption to provide space between two words, ^ will be used for stuffing between two alphabets if they are repeated in a pair and ^ will also be used to put at the end to get the last alphabet in pair if the total length at comes out to be odd. At the time of decryption | will be replaced by blank space of one alphabet and the symbol ^ will be discarded. Rules for encoding and decoding will be same as that for existing playfair cipher.

A. *Methodology Used*

Suppose the text to be encrypted is:
"It remains despite its non existence as one of the truly great wonders of India."
Keyword used: RAJA@RAM

Step 1: Adding | at blank spaces, the message is:
It|remains|despite|its|non|existence|as|one|of|the|truly|great|
wonders|of|India.
There turn out to be no double letter in a pair, so there is no use of ^. For giving space we have used |. At last total no of symbols came out to be even. So, we have not appended | at last to get them in pair.

Step 2: Keyword used is RAJA@RAM
After writing the repeating letter A once, we have
Keyword: RAJ@M
The grid comes out to be:

R	A	J	@	M	B	C	D
E	F	G	H	I	K	L	N
O	P	Q	S	T	U	V	W
X	Y	Z	0	1	2	3	4
5	6	7	8	9	!	#	$
%	^	&	*	(	)	_	+
=	{	}	[	]	|	\	:
;	"	,	<	>	.	/	?

Figure 15: Screenshot 4 of Appendix (Srivastava et al., 2010)