

DEVELOPMENT OF AN IMPROVED PLAYFAIR CRYPTOSYSTEM USING RHOTRIX

BY

Rukayyat ABDULKARIM, B.Sc. (A.B.U ZARIA) 2011

MSC/SCI/20610/2012-2013

A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES,  
AHMADU BELLO UNIVERSITY, ZARIA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF  
MASTER DEGREE IN COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF PHYSICAL SCIENCES

AHMADU BELLO UNIVERSITY,

ZARIA, NIGERIA

APRIL, 2017

## DECLARATION

I declare that the work in this dissertation entitled “DEVELOPMENT OF AN IMPROVED PLAYFAIR CRYPTOSYSTEM USING RHOTRIX” has been carried out by me in the Department of Computer Science under the supervision of Prof. A.A.Obiniyi and Dr. S.E. Abdullahi. The information derived from the literature has been duly acknowledged in the text and a list of references provided. No part of this dissertation was previously presented for another degree or diploma at this or any other institution.

ABDULKARIM, Rukayyat

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Date

## CERTIFICATION

This dissertation entitled “DEVELOPMENT OF AN IMPROVED PLAYFAIR CRYPTOSYSTEM USING RHOTRIX” by RUKAYYAT ABDULKARIM meets the regulations governing the award of the degree of Masters in Computer Science of the Ahmadu Bello University, and is approved for its contribution to knowledge and literary presentation.

Prof. A.A. Obinyi

Chairman, Supervisory Committee

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Date

Dr. S.E. Abdullahi

Member, Supervisory Committee

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Date

Prof. S.B.Junaidu

Head of Department

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Date

Prof. K. Bala

Dean, School of Postgraduate Studies

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Date

## **DEDICATION**

This dissertation is dedicated to Allah (SWT) for His Grace and Mercy, and to my beloved parent Dr. Yunusa Abdulkareem and Hajiya F. Abdulkareem, who were very supportive both physically and spiritually. May Allah reward you both with Aljannah Firdaus. Amin

## ACKNOWLEDMENT

My sincere gratitude to Allah (SWT) for his guidance and strength, He granted me all through my study.

My special gratitude goes to my supervisor, Prof. A.A. Obiniyi for his relentless effort to make this work a success, though you have tight schedule but you sacrificed your time to see that my work was complete. I sincerely appreciate Dr. S.E Abdullahi who also showed moral support towards the supervision of this work, I say a big thank you.

My special gratitude goes to my beloved and most cherished husband Mr Saheed Shittu for his patient, love and support throughout my study. Not forgetting my son Muhammad Hanif thanks for your love. May Allah bless and keep us together.

To my beloved and most cherished family, Hajiya Rahila Abdulkareem, Nafisa Abdulkareem, and darling Shamsudeen Abdulkareem thanks for your love, understanding and kindness you showed me all through my study. Not forgetting my nephews and niece, Abdurrahman, Tahir, Mahmud, Rahama, and Fatima thanks for your love. May Allah reward you all abundantly. One love keeps us together.

I owe my sincerest gratitude to all the Departmental staff and the entire PG (2012-2013) students of Computer Science Department for their support; I say a big thank you.

I thank and appreciate the good work of Mr. A. Ibrahim and Muhammed who also contributed to the success of this work.

Finally to all those whom I did not mention, I say a big thank you and God bless.

## ABSTRACT

Playfair is one of the best-known traditional ciphers but it is limited to different aspects such as: white space, numbers and other printable characters. The existing technique is based on the use of a  $5 \times 5$  matrix. This algorithm can only allow the text that contains uppercase alphabets only. But many algorithms have been proposed which allow extension of playfair to:  $6 \times 6$  matrix,  $8 \times 8$  matrix and  $7 \times 4$  matrix. This dissertation deals with some of the limitations such as lowercase, numbers, white space and repetition of alphabet in pairs. Proposed modification uses rhotrix which is a new paradigm of matrix theory for rectangular arrays. This work proposes an improved key matrix formation of the playfair cipher. The order of the rhotrix will be set to 6 to make it compatible with the  $[n \times (n-1)]$  mode of rhotrix representation which supports all 26 alphabets in both upper case letters (A-Z) as well as lower case letters (a-z), ten digits (0-9), special characters and the extended special characters. Experimental result produced better playfair cipher to incorporate randomness and to eliminate the chances of being attacked by frequency testing of character occurrence over an existing  $11 \times 11$  matrix. The system was developed in java because of platform independence and availability.

## TABLE OF CONTENTS

Title Page .....	i
Declaration.....	ii
Certification .....	iii
Acknowledgment .....	v
Abstract.....	vi
Table of Contents .....	vii
List of Figures .....	xi
List of Appendices .....	xii
Abbreviations, Definitions, Glossaries and Symbols .....	xiii
<b>CHAPTER ONE.....</b>	<b>1</b>
<b>INTRODUCTION.....</b>	<b>1</b>
<b>1.1 Background of the Study .....</b>	<b>1</b>
<b>1.2 Research Motivation.....</b>	<b>4</b>
<b>1.3 Research Problem.....</b>	<b>5</b>
<b>1.4 Aim and Objectives of the Study.....</b>	<b>6</b>
<b>1.5 Research Methodology .....</b>	<b>6</b>
<b>1.6 Organization of the Dissertation .....</b>	<b>7</b>
<b>CHAPTER TWO.....</b>	<b>8</b>
<b>LITERATURE REVIEW .....</b>	<b>8</b>
<b>2.1 Introduction .....</b>	<b>8</b>
<b>2.2 Origin of Cryptography.....</b>	<b>8</b>
<b>2.3 Cryptographic Algorithm.....</b>	<b>10</b>
<b>2.4 Classical Cryptography .....</b>	<b>13</b>
2.4.1 Substitution Cipher.....	13
2.4.2 Transposition Ciphers .....	14
2.4.3 Analysis of ciphers .....	14

<b>2.5</b>	<b>Modern Cryptography</b> .....	19
<b>2.6</b>	<b>Cryptanalytic attacks</b> .....	19
<b>2.7</b>	<b>Conversion of a Rhotrix to a ‘Coupled Matrix’</b> .....	21
<b>2.8</b>	<b>Purpose of Cryptography</b> .....	22
<b>2.9</b>	<b>Playfair Cipher</b> .....	23
2.9.1	Existing Playfair Cipher Algorithm .....	24
2.9.2	Analysis on Playfair Cipher.....	26
2.9.3	Limitation of Traditional Playfair Cipher.....	27
<b>2.10</b>	<b>Review of Related Literature</b> .....	27
<b>2.11</b>	<b>Literature Gap</b> .....	32
<b>CHAPTER THREE</b> .....		33
<b>DESIGN OF AN IMPROVED PLAYFAIR CRYPTOSYSTEM USING RHOTRIX</b> .....		33
<b>3.1</b>	<b>Introduction</b> .....	33
<b>3.2</b>	<b>Design Consideration</b> .....	33
<b>3.3</b>	<b>System Requirements</b> .....	34
3.3.1	Functional Requirements.....	34
<b>3.4</b>	<b>Transforming Rhotrix to Couple Matrix</b> .....	35
<b>3.5</b>	<b>Flow Chart</b> .....	37
<b>3.6</b>	<b>System Block Diagram</b> .....	39
<b>3.7</b>	<b>Algorithm for the Proposed System</b> .....	40
3.7.1	Proposed Key Formation System Algorithm.....	40
3.7.2	Algorithm for Encryption .....	41
<b>CHAPTER FOUR</b> .....		43
<b>RESULTS AND ANALYSIS</b> .....		43
<b>4.1</b>	<b>Introduction</b> .....	43
<b>4.2</b>	<b>Snapshots for the Implementation</b> .....	43
<b>4.3</b>	<b>Comparison of Methods and Results</b> .....	47

<b>4.4</b>	<b>Graphical Analysis</b> .....	50
4.4.1	English Alphabet Frequency.....	50
4.4.2	Analysis of Existing and Proposed System.....	51
<b>4.5</b>	<b>Statistical Analysis</b> .....	52
4.5.1	Character Length.....	53
4.5.2	Brute Force Attack.....	53
4.5.3	Space and Time Requirement.....	53
<b>CHAPTER FIVE</b> .....		54
<b>SUMMARY, CONCLUSION AND FUTURE WORK</b> .....		54
<b>5.1</b>	<b>Summary</b> .....	54
<b>5.2</b>	<b>Conclusion</b> .....	54
<b>5.3</b>	<b>Contribution to Knowledge</b> .....	55
<b>5.4</b>	<b>Future Work</b> .....	55
<b>REFERENCES</b> .....		56
<b>APPENDIX</b> .....		60

## LIST OF TABLES

Table 2.1: Probabilities of the Letter frequency of English Alphabets .....	16
Table 2.2: Traditional Playfair $5 \times 5$ matrix (Basu and Ray, 2012).....	24
Table 2.3: Traditional Playfair $5 \times 5$ matrix (Iqbal <i>et al.</i> , 2014).....	26
Table 2.4: Modified Version of Playfair Cipher Using $7 \times 4$ Matrix (Alam <i>et al.</i> , 2013).....	28
Table 2.5: Modify Playfair $6 \times 6$ Matrix (Chand and Bhattacharyya 2014) .....	29
Table 2.6: Modify Playfair $8 \times 8$ Matrix (Srivastava and Gupta 2011) .....	30
Table 2.7: Modify Playfair $10 \times 9$ Matrix (Bhattacharyya <i>et al.</i> , 2014) .....	31
Table 2.8: $11 \times 11$ Playfair matrix (Oduay and Baraa 2013) .....	32
Table 3.1: Proposed Key table generation.....	41

## LIST OF FIGURES

Figure 2.1: Secret Key Cryptography (Sangapu and Gomatam, 2014).....	11
Figure 2.2: Asymmetric Key Cryptography (Sangapu and Gomatam, 2014) .....	12
Figure 2.3: Process of Encryption and Decryption (Choudhary <i>et al.</i> , 2013) .....	13
Figure 3.1: Proposed system Flowchart. ....	38
Figure 3.2: System Block Diagram .....	39
Figure 4.1: Environment for Cryptographic Process .....	43
Figure 4.3: Encryption involving the key matrix based on the rhotrix model.....	45
Figure 4.4: Final Encryption phase .....	46
Figure 4.5: Decryption phase.....	47
Figure 4.6: Key generation and ciphertext of existing and proposed system.....	48
Figure 4.7: Test case showing the Key generation and ciphertext.....	49
Figure 4.8: Probability Distribution of the English Alphabet Characters .....	50
Figure 4.9: Graphical Analysis of Existing and Proposed System .....	51
Figure 4.10: Graphical Representation of Index of Coincidence.....	52

## LIST OF APPENDICES

Appendix A.....	60
Appendix B.....	61
Appendix C.....	62

## ABBREVIATIONS, DEFINITIONS, GLOSSARIES AND SYMBOLS

ACRONYM	DEFINITION
AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
CBC	Cipher Block Chaining
CFB	Cipher Feed Back
DES	Data Encryption Standard
IC	Index of Coincidence
JDK	Java Development Kit
RSA	Rivest–Shamir–Adleman

## CHAPTER ONE

### INTRODUCTION

#### 1.1 Background of the Study

The security of data transmission is an important problem in communication networks. A communication system is reliable as long as it maintains the integrity, availability, and privacy of data. Data integrity is the protection of data from unauthorized modification, resistance to penetration and undetected modification. Therefore, it is important to secure cryptosystem which provides encryption and decryption to the data. To achieve a secure cryptosystem, Cryptology is essential. Cryptology is the study of cryptography and cryptanalysis.

Cryptography is the study of information hiding and verification. It includes the protocols, algorithms and strategies to securely and consistently prevent or delay unauthorized access to sensitive information and enable verifiability of every component in a communication (Saini and Mandal, 2015).

Cryptanalysis is the art of 'attacking' cryptosystems in order to 'crack' them or at least discover their weaknesses (Stallings, 2011). When cryptanalysis reveals weaknesses in cryptosystems, cryptographers create more secure cryptosystems. Conversely, as cryptosystems become stronger, cryptanalysts try to discover more powerful methods of attacking them. Thus, cryptography and cryptanalysis are complementary.

Cryptography is divided into three main branches (Chirstof and Pelzi 2010) which are:

- a. **Symmetric Algorithms:** it refers to encryption and decryption methods in which both the sender and receiver share a secret key. All cryptography from ancient times until 1976 was exclusively based on symmetric methods. Symmetric

ciphers are still in widespread use, especially for data encryption and integrity check of messages.

- b. Asymmetric (or Public-Key) Algorithms: In public-key cryptography, a user possesses a secret key as in symmetric cryptography and a public key that may be freely distributed. Asymmetric algorithms can be used for applications such as digital signatures and key establishment, and also for classical data encryption.
- c. Cryptographic Protocols: crypto protocols deal with the application of cryptographic algorithms. Symmetric and asymmetric algorithms can be viewed as building blocks with which applications such as secure Internet communication can be realized. The Transport Layer Security (TLS) scheme, which is used in every Web browser, is an example of a cryptographic protocol.

Symmetric cryptography addresses the problem of secrecy protection by using the shared secret key to transform the message in such a way that it cannot be recovered anymore without this key. This process is called symmetric encryption. Based on the paradigm used to process the message, these ciphers are typically categorized into one of two classes: block ciphers and stream ciphers. The security of symmetric encryption algorithms can in general not be proved (the notable exception being the one-time pad). Instead, the trust in a cipher is merely based on the fact that no weaknesses have been found after a long and thorough evaluation phase (Canniere, 2007). Two types of ciphers are used in Symmetric Key Cryptography: Transposition cipher and Substitution cipher (Stallings, 2006).

In transposition cipher the characters in the plaintext are swapped to get the cipher text i.e. the characters retain their plaintext form but their position is changed. The plaintext is organized into two dimensional table and columns are interchanged according to a

predefined key. In substitution cipher one symbol of the plaintext is replaced by another symbol. Substitution ciphers has further two types. Monoalphabetic substitution cipher and poly alphabetic substitution cipher.

Monoalphabetic substitution cipher, is when a character in the plaintext is always changed to the same character in the cipher text. The well-known example of Mono alphabetic substitution cipher is the CAESAR cipher.

The polyalphabetic substitution cipher a single character in the plaintext is changed to many characters in the cipher text. The well-known example of poly alphabetic substitution cipher is VIGENERE cipher which changes a single character in the plaintext into many characters in the cipher text by considering the position of character in the plaintext.

This research work focuses on Playfair cipher which is a type of block cipher and also a substitution cipher, the Scheme was invented by British scientist Sir Charles Wheatstone in 1854, but was named after Lord Playfair who promoted the use of the cipher and hence it is called Playfair Cipher (Iqbal *et al.*, 2014). It was used by the British in the first and Second World War. It was also used by the Australians and Germans during World War II. Playfair is reasonably easy to use and was used to handle important but non-critical secrets. By the time the enemy cryptanalysts could break the message, the information would be useless to them. Between February 1941 and September 1945 the Government of New Zealand used it for communication between New Zealand, the Chatham Islands and the Pacific Islands. The technique encrypts pairs of letter, instead of single letters as in the simple substitution cipher. It was the first practical digraph substitution cipher, in which the ciphertext character replaces a particular plaintext character in the encryption which depend on an adjacent character in the plaintext. If the plaintext is viewed as a sequence of bits, then

substitution involves replacing plaintext bit patterns with ciphertext bit patterns (Stallings, 2011). The Playfair algorithm is based on the use of a  $5 \times 5$  matrix on the 26-letter characters of the English language constructed from the encryption key, with 2 of the 26 characters occupying a single position in the array. These two characters are  $i$  and  $j$ . Usually it is easy to distinguish from the context which of these two letters was intended in the plaintext. The encryption key for a Playfair cipher is a word, i.e., a finite sequence of characters taken from the set of plaintext characters. This keyword will determine the positioning of the characters in the encryption arrays.

Rhotrix is a new area of study relating to linear mathematical algebra, this concept of rhotrices was introduced by Ajibade (2003). As an extension of ideas on matrix-tertion and matrix-noitret, suggested by Atanassov and Shannon (1998). Ajibade defined rhotrix as mathematical arrays which are in some ways, between  $(2 \times 2)$  and  $(3 \times 3)$ -dimensional matrices. Extension in the size of R was considered possible. It is denoted by R and shown below.

$$R = \left\{ \begin{pmatrix} & a & \\ b & c & d \\ & e & \end{pmatrix} : a, b, c, d, e \in \mathfrak{R} \right\} \quad \dots \quad (1)$$

## 1.2 Research Motivation

The explosive growth in computer systems and their interconnection via network has increased the dependence of both organizations and individuals on the information stored and communicated using these systems which intern has led to a heightened awareness of the need to protect data and resources from intruders. (Jitendra *et al.*, 2013). Cryptography is the design of certain techniques for ensuring the secrecy and/or authenticity of information. The need of cryptographic algorithm is to avoid threat to integrity, confidentiality and availability. In every sector, collection of information or

transfer of data with a high level of security is needed. For this reason a strong encryption technique is required. This research focuses on play fair cipher algorithm which is very strong and also requires less memory and power. Although a considerable effort has been done in analysing Playfair ciphers of various sizes, this observation motivates the study carried out in this dissertation.

The Motivation for the research is based on the existing playfair cipher and some researchers' concepts which address some limitation of the playfair cipher in a unique but separate ways. The initial driving force behind the conception of the work was to extend the traditional playfair cipher which was based on the  $5 \times 5$  matrix and to equally use the concept of rhotrix which was introduced by Ajibade (2003) and the idea of conversion of rhotrix to a coupled matrix by Sani (2008) to obtain more security to the cipher.

### **1.3 Research Problem**

Millions of monetary transactions take place per second online. Also, daily business activities nowadays depend on software and the need for strong cryptographic algorithms or concepts cannot be overemphasized. The 21st century fraudsters or hackers are sophisticated in their malicious or criminal activities and therefore software security must be far ahead of them. (Ibikun *et al.*, 2013). As information is transferred from one user to another the information becomes highly vulnerable to all kind of threats caused by adversaries, the data communication between two entities can be secured if an encryption and decryption technique is used at two end points (Harinandan *et al.*, 2014). Therefore achieving faster communication in most confidential data is circulated through networks as electronic data. Cryptographic ciphers have an important role for providing security to these confidential data against unauthorized attacks (Ranjeet *et al.*, 2014). One of the well-known digraph substitution cipher is the Playfair

Cipher, which treats the plain text as single units and translates these units into cipher text. Though in the Playfair system, it is significantly hard to break with the frequency analysis used for simple monographs substitution ciphers (Safwat, 2014).

The drawback of the Playfair cipher is that the plain text consists of 25 uppercase letters only. One letter has to be omitted and cannot be reconstructed after decryption. Also lowercase letters, white space, numeric values and other printable characters cannot be handled by the traditional cipher. This means that complete sentences cannot be handled by this cipher. X is used as filler letter, and also as padding for odd numbers of letters in the message. Therefore could not cater for special characters thus limiting the possible combination for keys formation.

#### **1.4 Aim and Objectives of the Study**

The aim of this research is to develop a modified playfair cryptosystem using rhotrix. The objectives are to:

- a. introduce rhotrix as a coupled matrix to data structure of playfair
- b. support more characters, punctuation and some other printable characters.
- c. create more confusions to the cipher by taking advantage of a different form of matrix.
- d. implement the cryptosystem and evaluate for frequency analysis.

#### **1.5 Research Methodology**

The method used in this study is as follows:

- a. Rhotrix is used instead of matrix, which is a new paradigm of matrix theory of rectangular arrays.

- b. The size of the rhotrix is set to 6 to make it compatible with the  $[n \times (n-1)]$  mode of rhotrix representation.
- c. Repetitions of alphabets and strings with odd length are tackled using a filler character for efficient implementation.
- d. Playfair rules for encryption and decryption was adopted.
- e. Procedures from a to d are used to develop a secure cryptosystem, and would be implemented using java programming language.

## **1.6 Organization of the Dissertation**

The rest of this work discusses the implementation of a modified playfair cipher using rhotrix and is organized as follows: chapter 2 presents a literature and related works. Chapter 3 analyses this work by explaining tools used in the development of this work and analysis on how it was designed. Chapter 4 shows how this work was implemented and Chapter 5 gives a general conclusion on the work done, and its importance to cryptography.

## **CHAPTER TWO**

### **LITERATURE REVIEW**

#### **2.1 Introduction**

The purpose of this chapter is to review the relevant literature in the research of cryptography, modern cryptography, and some related modifications of playfair cipher that pertain to the present study. A description of the research that has been accomplished in the area of playfair ciphers was also provided.

#### **2.2 Origin of Cryptography**

Cryptography is derived from the Greek words: *kryptós*, "hidden", and *gráphein*, "to write" - or "hidden writing". People who study and develop cryptography are called cryptographers. Cryptography is the oldest and effective way to provide security to computer networks and data. The very first cryptographic techniques were developed over 200 years ago. Cryptography methods began with a person carving messages into wood or stone, which were then passed to the intended individual who had the necessary means to decipher the messages. This is a long way from how cryptography is being used today. Cryptography that used to be carved into materials is now being inserted into streams of binary code that passes over network wires, internet communication paths, and airwaves. Cryptography began around 2000 B.C, in Egypt when hieroglyphics were used to decorate tombs to tell the story of the life of the deceased. The practice was not as much to hide the messages themselves, but to make them seem more noble, ceremonial, and majestic. Around 400B.C, encrypting information system by writing a message on a sheet of papyrus, which was wrapped around a staff, was used. The message was only readable if it was around the correct staff, which allowed the letters to properly match up. This is referred to as the scytale

cipher when the papyrus was removed from the staff; the writing appeared as just a bunch of random characters (Wasnik *et al.*, 2013). In India, cryptography was apparently well known. It is recommended in the Kama Sutra as a technique by which lovers can communicate without being discovered. This may imply that cryptanalytic techniques were less than well developed in India 500 CE. Cryptography became important still later as a consequence of political competition and religious analysis. For instance, in Europe during and after the Renaissance, citizens of the various Italian states, including the Papacy, were responsible for substantial improvements in cryptographic practice. The Arab world religiously motivated textual analysis of the Koran led to the invention of the frequency analysis technique for breaking monoalphabetic substitution ciphers sometime around 1000 CE. Cryptography, cryptanalysis, and secret agent betrayal featured in the Babington plot during the reign of Queen Elizabeth I which led to the execution of Mary, Queen of Scots. And an encrypted message from the time of the Man in the Iron Mask (decrypted around 1900 by Étienne Bazeries) has shed some, regrettably non-definitive, light on the identity of that legendary, and unfortunate, prisoner. Cryptography, and its misuse, was involved in the plotting which led to the execution of Mata Hari and even more reprehensibly, if possible, in the travesty which led to Dreyfus' conviction and imprisonment, both in the early 20th century. Fortunately, cryptographers were also involved in setting Dreyfus free; Mata Hari, in contrast, was shot (Saini and Mandal, 2015).

In the past, messengers were used as the transmission mechanism, and encryption helped protect the message in case the messenger was captured. Today, the transmission mechanism has changed from human beings to packets carrying 0's and 1's passing through network cables or open airwaves. The messages are still encrypted in case an

intruder captures the transmission mechanism (the packets) as they travel along their paths.

In modern days, cryptography is no longer limited to secure sensitive military information but recognized as one of the major components of the security policy of any organization and considered industry standard for providing information security, trust, controlling access to resources, and electronic financial transactions (Abomhara *et al.*, 2010). The original data that is transmitted or stored is called plaintext, the one that can be readable and understandable either by a person or by a computer. Whereas the disguised data so-called ciphertext, which is unreadable, neither human nor machine can properly process it until it is decrypted. A system or product that provides encryption and decryption is called cryptosystem. Cryptosystem uses an encryption algorithms which determines how simple or complex the encryption process will be, the necessary software component, and the key (usually a long string of bits), which works with the algorithm to encrypt and decrypt the data.

### **2.3 Cryptographic Algorithm**

Cryptography is a process associated with scrambling plaintext (ordinary text, or cleartext) into ciphertext (a process called encryption), then back again (known as decryption). There are several ways to classify the various algorithms. The most common types are: symmetric key algorithm and public key algorithm. (Ayushi, 2010)

In symmetric key algorithm (secret key Cryptography), a single key is used for both encryption and decryption. As shown in figure 2.1, the sender uses the key (or some set of rules) to encrypt the plaintext and sends the ciphertext to the receiver. The receiver applies the same key to decrypt the message and recover the plaintext. Since a single key is used for both functions, secret key cryptography is also called symmetric

encryption. With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver.

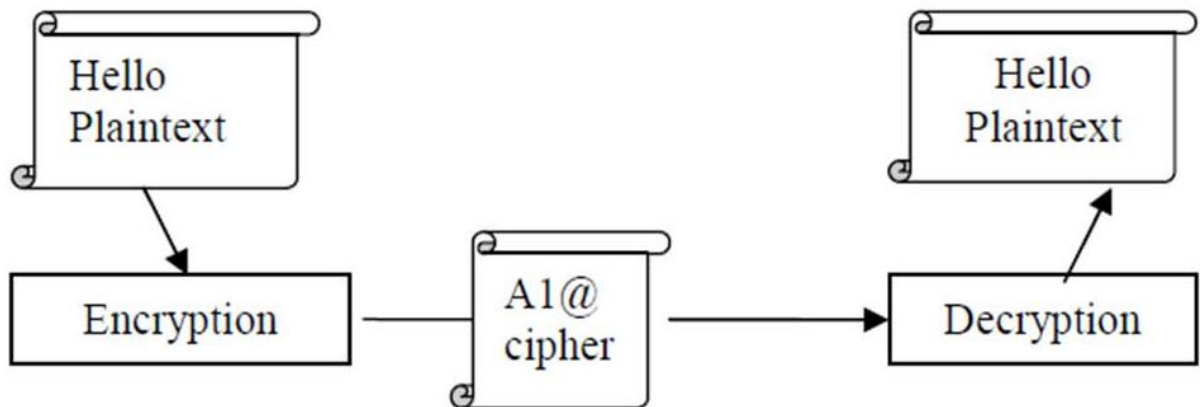


Figure 2.1: Secret Key Cryptography (Sangapu and Gomatam, 2014)

In Asymmetric key algorithm (Public key cryptography) it involves the use of key pairs: one private key and one public key. Both are required to encrypt and decrypt a message or transmission.

The private key, not to be confused with the key utilized in private key cryptography, is just that, it is not to be shared with anyone. The owner of the key is responsible for securing it in such a manner that it will not be lost or compromised. On the other hand, the Public key cryptography intends for public keys to be accessible to all users. In fact, this is what makes the system strong. If a person can access anyone public key easily, usually via some form of directory service, then the two parties can communicate securely and with little effort, i.e. without a prior key distribution arrangement

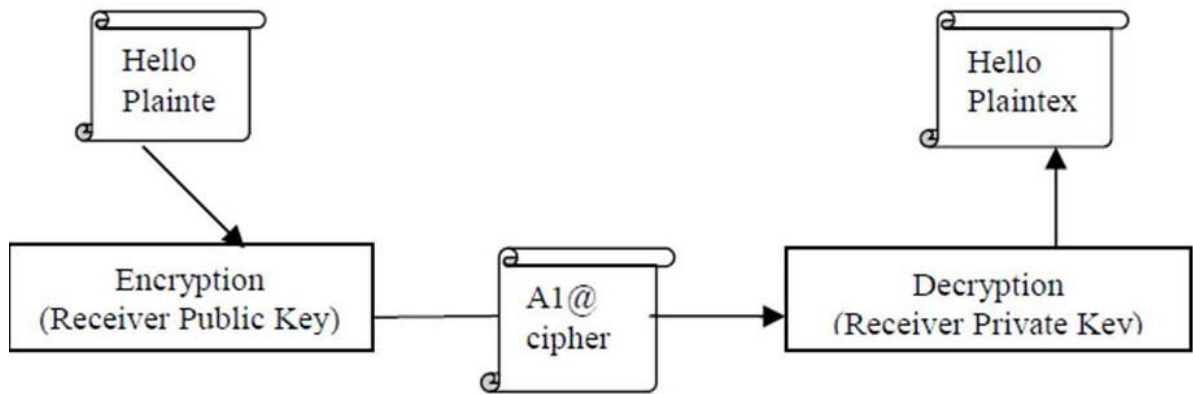


Figure 2.2: Asymmetric Key Cryptography (Sangapu and Gomatam, 2014)

### 2.3.1 Cryptography Technique

This technique is based on the encryption and decryption of plain-text to cipher-text which is safe to transmit message and preventing it from unauthorized access, by using a secret key in a specific algorithm. It uses the following ingredients:

- a. Plain Text: This is an intelligible piece of information i.e. original text that needs to be transferred safely to the receiver. It is the main input to the encryption algorithm.
- b. Secret Key: This is another input to the encryption and decryption algorithm, which is the main component used for converting the plain-text to cipher-text i.e. an unintelligible form which has the useful content hidden in a way.
- c. Encryption Algorithm: This is the actual process by which we are converting the plain-text into cipher text.
- d. Cipher Text: This is the output of the encryption process in which we are taking plain-text and secret key as input and processed by the encryption algorithm. The cipher-text can be understood as a scrambled piece of text which has useful information in secret form.

- e. Decryption Algorithm: This algorithm is the reverse of the encryption algorithm which takes in cipher-text and secret key as inputs and produces plain-text as the output.

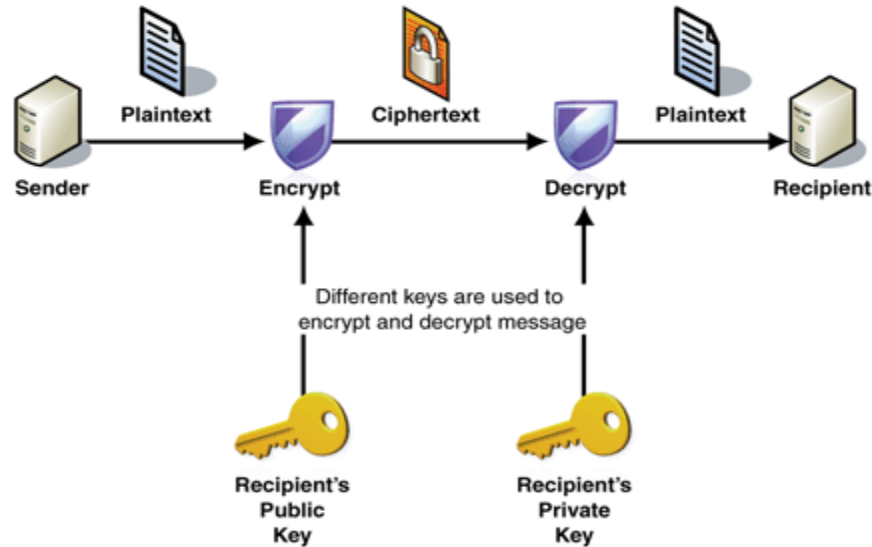


Figure 2.3: Process of Encryption and Decryption (Choudhary *et al.*, 2013)

## 2.4 Classical Cryptography

Classical cryptography is the oldest branch of cryptography which has been in existence for over 4000 years. They are ciphers that operate on an alphabet of letters (such as “A-Z”) and are implemented by hand or with simple mechanical devices. They are the earliest and most basic type of ciphers which makes them not very reliable with the recent development of technology. Classical ciphers were mostly used in the ancient days for military and diplomatic purposes. The two basic components of classical cryptography are substitution and transposition.

### 2.4.1 Substitution Cipher

Substitution cipher substitutes one symbol with another. If the symbols in the plaintext are alphabetic characters, it replaces one character with another. Substitution ciphers can be categorized as either mono-alphabetic or polyalphabetic ciphers.

In a mono-alphabetic cipher, a character (or a symbol) in the plaintext is always changed to the same character (or symbol) in the ciphertext regardless of its position in the text. For example, if the algorithm says that character A in the plaintext is changed to character D, every character A is changed to character D. In other words, the relationship between characters in the plaintext and the ciphertext is a one-to-one relationship.

In a polyalphabetic cipher, each occurrence of a character can have a different substitute. The relationship between a character in the plaintext and a character in the ciphertext is a one-to many relationships. For example, character A could be changed to D in the beginning of the text, but it could be changed to N at the middle. It is obvious that if the relationship between plaintext characters and ciphertext characters is one-to many, the key must indicate which of the many possible characters can be chosen for encryption.

#### **2.4.2 Transposition Ciphers**

In a transposition cipher, there is no substitution of characters; instead, their locations change. A character in the first position of the plaintext may appear in the tenth position of the ciphertext. A character in the eighth position may appear in the first position. In other words, a transposition cipher reorders the symbols in a block of symbols.

#### **2.4.3 Analysis of ciphers**

##### **a. The Friedman Test**

The Friedman Test is a statistical probabilistic method that can be used to determine the likelihood that the ciphertext message produced comes from a monoalphabetic or polyalphabetic cipher and for determining the length of the keyword if the cipher is polyalphabetic (Chris, 2006). Friedman test is sometimes called the Kappa Test. This technique of cryptanalysis was developed in 1925 by William Friedman. His method

was based upon the probability of randomly selecting two letters from an alphabet and having them be the same. Consider for example, an event of drawing two spades from a standard deck of cards of 52. Determining the probability of both cards drawn, there are 13 spades in the deck of 52. Therefore, the probability that the first card selected is a spade is 13/52.

Now, only 12 spades remain among the remaining 51 cards, so the probability that the second card drawn is 12/51.

Therefore, the total probability that both cards drawn are spades is:

$$(13/52) * (12/51) \approx 0.0588$$

Applying the above concept to determine the probability of drawing two of the same letters from a ciphertext, the probability of random selection for drawing two characters to be the same is considered.

Assuming  $n$  represents the number of letters in a given ciphertext and  $n(a)$  denotes the number of a's. The probability of selecting two a's would be:

$$\frac{n(a)}{n} \times \frac{n(a)-1}{n-1} \quad \dots (2.1)$$

Where  $n(a)$  is the number of a's and  $n$  is the number of letters in the given ciphertext

The probability of choosing two letters which are the same (i.e., two a's, two b's... two z's), would be represented as:

$$P_b(2a's, 2b's \dots 2z's) = \frac{n(a)}{n} \times \frac{n(a)-1}{n-1} + \frac{n(b)}{n-2} \times \frac{n(b)-1}{n-3} + \dots \frac{n(z)}{2} \times \frac{n(z)-1}{1} \quad \dots (2.2)$$

Where  $n(a)$  is the number of a's,  $n(b)$  is number of b's, and  $n(z)$  number of z's in the ciphertext.

The resultant probability is called the Index of Coincidence of the ciphertext, and is denoted by 'I'. The index of coincidence is a measure of how similar a frequency distribution is to the uniform distribution. The index of coincidence of a piece of text does not change if the text is enciphered with a substitution cipher. It is the probability that two randomly selected letters are the same, and is defined by the formula (Chris, 2006):

$$I.C = \frac{\sum_{i=A}^{i=Z} f_i(f_i - 1)}{N(N - 1)} \quad \dots (2.3)$$

Where  $f_i$  represents the count (frequency) of letter  $i$  (where  $i = A, B, C \dots Z$ ) in the ciphertext, and  $N$  is the total number of letters in the ciphertext.

The probability of the frequencies of the letters in the English language alphabet is represented in Table 2.1:

Table 2.1: Probabilities of the Letter frequency of English Alphabets

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M
Frequency	0.0	0.01	0.02	0.04	0.12	0.02	0.02	0.06	0.06	0.00	0.00	0.04	0.02
	817	5	78	25	7	23	02	09	97	15	77	03	41
Letter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Frequency	0.0	0.07	0.01	0.00	0.05	0.06	0.09	0.02	0.00	0.02	0.00	0.01	0.00
	675	51	93	1	99	33	06	76	98	36	15	97	07

According to table 2.1, the Index of Coincidence I approximately equals to 0.06560

### b. The Kasiski Test

Friedrich Kasiski was the first to publish a successful attack on the polyalphabetic cipher (Kester, 2013). Repeated fragments look up was suggested in the ciphertext which comprises of repetitions when the keyword encrypts the same fragment of text

several times. The presence of these repetitions is used to compile a list of the distances that separate the repetitions. This distance is likely to be the keyword length, which is used for further cryptanalysis. Once the length of the keyword is discovered, the message could be grouped in columns and then monoalphabetic substitution is performed on each column to determine the keyword. Kasiski observed the following (Katz and Lindell, 2008):

- a. That if a repeated substring in a plaintext is encrypted by the same substring in the keyword, then the ciphertext contains a repeated substring and the distance of the two occurrences is a multiple of the length of the keyword.
- b. Not every repeated string in the ciphertext arises in this way; but the probability of a repetition by chance is very small.

**c. Letter Frequency**

The letter frequency is a method used to check if a type of permutation cipher was used and thus if the letter distribution still matches the language of the plaintext. This is usually used to easily break down substitution ciphers, since its major weakness is that each plaintext character always maps to the same ciphertext character. This means that the statistical properties of the plaintext are preserved in the ciphertext (Christof and Pelzi, 2010). Initial random guesses are then made to compare the character with the highest frequency in the ciphertext with that of the highest probability distribution in the English text. The larger the text, the easier it is for the comparison. The letter with the highest probability distribution in a ciphertext is most likely to be an 'E', because from the probability distribution of English language characters, 'E' has the highest probability of occurrence, followed by 'T', then 'A', and so on. Once a little part of the ciphertext is matched with the probability of occurrence, the plaintext can be gotten.

Sorting out the letters of the alphabet according to their probability of occurrence gives: “E,T,A,O,I,N,S,R,H,L,D,C,U,M,F,P,G,W,Y,B,V,K,X,J,Q,Z” (Katz and Lindell, 2008)

Although, identifying the letters with the most frequency of abundance correctly depends on the amount of text at your disposal. The more the text, the easier it is to match each suspected character with its correspondence. Note that only the most frequent letters are identified, while the order of the less frequent ones can be statistically insignificant.

It is also possible to consider the frequency of the pairs of letters to ease cryptanalysis. According to the ciphertext, the existence of repeating pairs such as LX, LM, and NL can be used to find out the length of the key based on Kasiski’s principle. The most frequent pairs that exist in English language according to another cipher are given as follows:

TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, and OF. These pairing could be used in comparison with the existing pairs of the ciphertext to try and guess what they are.

Another method of cryptanalysis is by considering the frequency of the existence of trigrams in the English language. The most common trigrams that exist are given as follows:

THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, and DTH (Kester, 2013)

The ciphertext is studied to notice repeating pairs, thereby determining the length of the keyword. Once the length of the keyword is known, cryptanalysis takes place, which in turn compromises the encrypted text.

## **2.5 Modern Cryptography**

Keys are the most fundamental essential element in generating modern ciphertext. With the advent of the computer, ciphers need to be bit-oriented. This is so because the information to be encrypted is not just text; it can also consist of numbers, graphics, audio, and video data. It is convenient to convert these types of data into a stream of bits, encrypt the stream, and then send the encrypted stream. In addition, when text is treated at the bit level, each character is replaced by 8 (or 16) bits, which means the number of symbols becomes 8 (or 16). Mingling and mangling bits provides more security than mingling and mangling characters. Modern ciphers use a different strategy than the traditional ones. A modern symmetric cipher is a combination of simple ciphers. In other words, a modern cipher uses several simple ciphers to achieve its goal. Given a cipher, a key determines the mapping of the plaintext to the ciphertext. The key to a particular cryptosystem is such that when it is applied to the ciphertext, allows the encrypted message to be decrypted and when applied to a plaintext encrypts it. It is important to note that in the study of cryptography one talks about the lengths of keys in terms of bits. Messages, too, can be encrypted in 'n-bit blocks'. The longer a key is the more difficult it is to break the encrypted message using brute force, which is the most common method of breaking ciphers. This sort of attack involves running through possible combinations of keys and applying them to the cryptosystem until the message is decrypted.

## **2.6 Cryptanalytic attacks**

Cryptanalysis is a technique used for recovering encrypted messages from ciphertext with little or no knowledge of the key being used, while attacks are attempts by cryptanalysts on ciphertext, whether successful or not. Computer networks and distributed systems are vulnerable to a wide variety of threats that may be infused either

by intruders/illegal users or by legitimate users. Legitimate users are more powerful cryptanalysts than intruders, since they are insiders and possess information that is usually not available to intruders.

Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs (Stallings, 2011). There exist different types of cryptanalytic attacks, which include:

- a. *Ciphertext only Attacks*: this is a cryptanalytic attack in which the cryptanalyst has the ciphertext of one or more messages which have been encrypted with the same encryption technique, but doesn't have its key or the corresponding plaintext. This data is relatively easy to obtain in many scenarios, but a successful ciphertext-only attack is generally difficult, and requires a very large ciphertext sample.
- b. *Known plaintext Attacks*: this is a cryptanalytic attack in which the cryptanalyst knows both the plaintext and its corresponding ciphertext. The algorithm used is to decode the encoded messages and finding the corresponding key used, in order to easily decode future messages using the key. This type of attack can be applied on block ciphers.
- c. *Chosen Ciphertext Attacks*: in this type of attack, the cryptanalyst chooses a piece of the ciphertext and tries to find the corresponding decoded plaintext. This attack is mostly applicable to the public key cryptosystem such as Rivest Shamir Adleman (RSA).
- d. *Chosen Plaintext Attacks*: an attacker is able to obtain a chosen plaintext and its corresponding encrypted ciphertext. This identifies the key used to encode the message or an algorithm to decode future messages encoded with exactly the

same key. A good example of this attack is the differential cryptanalysis which can be applied against block ciphers (and in some cases also against hash functions). Some cryptosystems, particularly RSA, are vulnerable to chosen-plaintext attacks. Therefore, care must be taken to design the algorithm in order to prevent an attacker from having chosen plaintext encrypted.

- e. *Man in the middle Attacks*: this is a cryptanalytic attack in which the cryptanalyst intercepts the signal of a communication channel between two parties and then separately make a key exchange with each of them. A key that is already known to the cryptanalyst is used; the attacker decrypts messages that are sent and change its contents at will. Communication is assumed to be secure, not knowing that the cryptanalyst is seeing and modifying the messages at will. It is usually relevant for cryptographic communication and key exchange protocols. This type of attack can be prevented with the use of public key cryptosystem capable of providing digital signature such as RSA cryptosystem. Both parties in advance will have a common public key, then generate the shared message and attach a digital signature to it. The man-in-the-middle here cannot know the digital signature used without prior knowledge of the secret keys. But this is only possible if there is a secure way to secretly distribute the public key securely.

## **2.7 Conversion of a Rhotrix to a ‘Coupled Matrix’**

The method of converting a rhotrix to a special form of matrix called a ‘coupled matrix’ was defined by (Sani, 2008). This special matrix can be used to solve various problems involving  $n \times n$  and  $(n-1) \times (n-1)$  matrices simultaneously. Transposing any rhotrix or matrix is equivalent to rotating its columns through  $90^\circ$  in an anti-clockwise direction.

In the same way, if columns of a rhotrix are rotated through  $45^\circ$ , what they get is a special form of matrix with missing values.

For a R5 rhotrix for instance,

$$R_5^{T/2} = \left( \begin{array}{cccc} & & a_{11} & \\ & a_{21} & c_{11} & a_{12} \\ a_{31} & c_{21} & a_{22} & c_{12} & a_{13} \\ & a_{32} & c_{22} & a_{23} \\ & & a_{33} & \end{array} \right) = \begin{bmatrix} a_{11} & & a_{12} & a_{13} \\ & c_{11} & & c_{12} \\ a_{21} & & a_{22} & a_{23} \\ & c_{21} & & c_{22} \\ a_{31} & & a_{32} & a_{33} \end{bmatrix} \quad \dots(2.6)$$

where  $T/2$  indicates a rotation through  $45^\circ$  in an anti-clockwise direction and may be termed as ‘half transpose’. The special matrix in (2.6) is like coupling a  $3 \times 3$  matrix with a  $2 \times 2$  matrix, which is why, is called ‘a coupled matrix’. Therefore in general case

$$R_n^{T/2} = \langle a_{ij}, c_{lk} \rangle^{T/2} = [a_{ij}, c_{lk}] = [Ac]_n \quad \dots(2.7)$$

which is a coupled matrix. To multiply any two coupled matrices,  $[Ac]_n$  and  $[Bd]_n$ , the missing spaces is filled with zeros and multiply as usual for any two matrices. After the multiplication, the zeros is removed so as to end up with a coupled matrix. Thus for two coupled matrices, coupling  $t \times t$  matrices with  $(t-1) \times (t-1)$  matrices, Sani (2008) got

$$[Ac]_n [Bd]_n = [a_{i1j1}, c_{i1k1}] [b_{i2j2}, d_{i2k2}] = \left[ \sum_{i_2 j_2=1}^{t-1} (a_{i_1 j_1} b_{i_2 j_2}), \sum_{i_2 k_2=1}^{t-1} (c_{i_1 k_1} d_{i_2 k_2}) \right] \dots (2.8)$$

which is the same method of multiplication for rhotrices. That means, both the  $t \times t$  matrices and the  $(t-1) \times (t-1)$  matrices in the two coupled matrices are multiplied together using row–column multiplication.

## 2.8 Purpose of Cryptography

The main use of cryptography is to provide the following:

- a. *Confidentiality* is a service used to keep the content of information from all but those authorized to possess it. Secrecy is a term synonymous with confidentiality and privacy. There are numerous approaches to providing

confidentiality, ranging from physical protection to mathematical algorithms which render data unintelligible.

- b. *Data integrity* is a service which addresses the unauthorized alteration of data. To assure data integrity, one must have the ability to detect data manipulation by unauthorized parties. Data manipulation includes such things as insertion, deletion, and substitution
- c. *Authentication* is a service related to identification. This function applies to both entities and information itself. Two parties entering into a communication should identify each other. Information delivered over a channel should be authenticated as to origin, such as date of origin, data content, and time sent. For these reasons this aspect of cryptography is usually subdivided into two major classes: entity-authentication and data-origin authentication. Data origin authentication implicitly provides data integrity.
- d. *Non-repudiation* is a service which prevents an entity from denying previous commitments or actions. When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary.

## **2.9 Playfair Cipher**

Playfair cipher is manual symmetric encryption technique and was the first literal digraph substitution cipher. The first recorded description of the Playfair cipher was in a document signed by Wheatstone on 26 March 1854 (Iqbal *et al.*, 2014). However Lord Playfair promoted the use of this cipher and hence it is called Playfair Cipher. It was used by the British in the first and Second War World (Bhattacharyya *et al.*, 2014). It was also used by the Australians and Germans during World War II. Playfair is reasonably easy to use and was used to handle important but non-critical secrets. By the time the enemy cryptanalysts could break the message, the information would be

useless to them. Between February 1941 and September 1945 the Government of New Zealand used it for communication between New Zealand, the Chatham Islands and the Pacific Islands.

### 2.9.1 Existing Playfair Cipher Algorithm

The play fair cipher algorithm is based on the use of  $5 \times 5$  matrix of letters constructed using a keyword (i.e. the traditional cipher uses 25 uppercase alphabets with I=J or Q omitted). The matrix is constructed by filling the letters of keyword from left to right and top to bottom and the remaining cells are filled in alphabetic order ignoring the letters of keyword. Rules are also required to create the matrix and the used cipher.

Table 2.2 shows the construction of  $5 \times 5$  matrix using the keyword DUPLICATE

Table 2.2: Traditional Playfair  $5 \times 5$  matrix (Basu and Ray, 2012)

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

In table 2.2, the word DUPLICATE is the key supplied by the user. The key is entered into the first section of the  $5 \times 5$  array and then the alphabets are then entered but with no repetition. Since the letter A is in the key, it is not entered instead B is entered. In the instance of letter C, the key also contains C so, it is not entered, and the subsequent letters are then entered as long as they are not in the keyword.

The message is then broken up with digraphs or groups of 2 letters. In case of duplication of letters in a digraph one of the letters is used as padding and is placed between the letters. In case of odd number of characters the same padding is applied at

the end. Encryption in playfair is done based on substitution depending on the following three rules.

- a. Plaintext letters that fall on the same row matrix are each replaced by the letter on the right, with the first element of the row circularly following the last.
- b. In case of letters in the same column the letters to the bottom of each letter are taken. Wrapping happens in case any letter is in the last row.
- c. In case the letters are neither in the same row or column a rectangle is made with the letters and the letters at the opposite corners are taken.

The decryption process is done with the cipher text depending on the following rules

- a. Cipher text that fall on the same row matrix are replace with the letter on the left of the letter, with the last element of the row circularly following the first.
- b. Just in case both the letters are in the same column, replace them with the letter above them. If the letter is at the top, go back to the bottom of the column and use the letter to replace with top letter.
- c. If neither of the alphabets lies in the same column nor same row, imagine creating a rectangle form and write the corners alphabets.

For example balloon is the plaintext and duplicate as the secret keyword the ciphertext can be derived as follows.

First the plaintext is converted to uppercase and then broken up into digraphs using X as the padding character. The digraphs will be BA LX LO ON. For the first digraph B and A are in the same row. Using rule 'a' of the encryption process, the ciphertext gotten is CT. The next digraph LX is taken, they are neither in the same row or column. Hence using rule 'c' of the encryption process, the cipher gotten is PY. The next digraph is LO which is also neither in the same row or column. Hence using rule 'c' of the encryption

process, the cipher gotten is UR. The last digraph is ON which are in the same row, the cipher gotten is QO. Thus the cipher text is CTPYURQO

### 2.9.2 Analysis on Playfair Cipher

In Playfair cipher, the alphabets are arranged in a 5×5 key matrix based on secret key.

Key: SIMPLE

Table 2.3: Traditional Playfair 5 x 5 matrix (Iqbal *et al.*, 2014)

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

Though there are 26-alphabets in English language but Playfair cipher can handle only 25-alphabets. So, either Q is discarded or any one of i/j is used. To fill-in the key matrix table, the letters of the keyword (dropping any duplicate letters) are put in serially, and then the remaining spaces are filled with the rest of the letters of the alphabet in order. To encrypt a message, the message is broke into digraphs (groups of 2 letters) such that, for example, "HELLOWORLD" is to be treated as "HE LL OW OR LD", and then mapped them out on the key table. Then following 4 rules are applied, in order, to each pair of letters in the plaintext. If:

- a. Both letters are the same (or only one letter is left), add a *filler* after the first letter. Encrypt the new pair and continue.
- b. The letters appear on the same row of your table, they are to be replaced with the letters immediate right of them respectively.
- c. The letters appear on the same column of your table, they are to be replaced with the letters immediately below them respectively.

- d. The letters are not on the same row or column, replace them with the letters on the same row respectively but of the column of the other keeping the order of the pair intact.

### **2.9.3 Limitation of Traditional Playfair Cipher**

The original Playfair comprises of  $5 \times 5$  matrix in which the plain text can consist of 25 uppercase letters only, so it cannot encrypt lowercase letters. In this algorithm, the letters I & J are counted as one character. It is seen that the rules of encryption applies a pair of plaintext characters. So, it needs always even number of characters in plaintext message. In case, the message counts odd number of characters a spare letter X is added at the end of the plaintext message. Further repeating plaintext letters in the same pair are separated with a filler letter, such as X, so that the words COMMUNICATE would be treated as CO MX MU NI CA TE. One letter has to be omitted and cannot be reconstructed after decryption. Also white space, numbers and other printable characters cannot be handled by the traditional cipher. This means that complete sentences cannot be handled by this cipher.

### **2.10 Review of Related Literature**

The society depends on cryptography for a secure information communication, monetary and online transactions (Huang *et al.*, 2013). However, many threats emerge constantly that must be avoided, or curbed, without loss of assets.

In the work of Ravindra *et al.* (2011) the  $5 \times 5$  matrix was replaced by  $6 \times 6$  matrix. In this system all the uppercase alphabets as well as numbers can be handled. However, lowercase letters, white space and other printable characters cannot be handled.

Alam *et al.* (2013), modifies the original  $5 \times 5$  matrix playfair cipher to  $7 \times 4$  matrix playfair cipher in which any word with no repeating letter can be selected as a key

word, the remaining spaces are filled in order with the rest of alphabets, the second last and last cell is filled with the symbol “\*” and “#” respectively.

Table 2.4: Modified Version of Playfair Cipher Using 7×4 Matrix (Alam *et al.*, 2013)

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

The addition of “\*” and “#” in the matrix creates one-to-one correspondence between the plaintext and the ciphertext, which makes the encryption and decryption easy and unambiguous. However, lowercase letters, numbers, white space and other printable characters cannot be handled. Also the symbols “\*” and “#” at the time of decipherment was ignored.

Chand and Bhattacharyya (2014), develop a new concept which includes 6 × 6 play fair cipher matrix. This matrix consists of alphabets A to Z and numeric values 0 to 9. Four iteration steps of four different keywords were used to encrypt and decrypt text message successfully.

Table 2.5: Modify Playfair  $6 \times 6$  Matrix (Chand and Bhattacharyya 2014)

M	O	N	A	R	C
H	Y	B	D	E	F
G	I	J	K	L	P
Q	S	T	U	V	W
X	Z	0	1	2	3
4	5	6	7	8	9

This extended play fair algorithm was based on the use of four  $6 \times 6$  matrices of letters constructed using corresponding four keywords. The matrix is constructed by filling in the letters of the keyword from left to right and from top to bottom, and the filling in the remainder of the matrix with the remaining letters in alphabetic order and digits in ascending order form 0 to 9. However, lowercase letters, white space and other printable characters cannot be handled.

Basu and Ray (2012), assumed a rectangular matrix having 10 columns and 9 rows which can support almost all the printable characters including white space. The  $10 \times 9$  matrix contains almost all the printable characters. This includes lowercase and uppercase alphabets, punctuation marks, numbers and special characters. The order of placement of different groups of characters can also be done so that the matrix formed by using the same secret keyword depends on the order of placement. This means that the ciphertext will also depend on the order of placement of different groups of characters. The matrix with the secret keyword as Duplicate and a particular placement order is given.

Srivastava and Gupta (2011), 8 x 8 matrix was used instead of the traditional 5x5 matrix for creating the cipher text. The 8 x 8 matrix contains uppercase alphabets, numbers, punctuation mark and some special characters.

Table 2.6: Modify Playfair 8×8 Matrix (Srivastava and Gupta 2011)

S	H	I	V	@	A	K	T
B	C	D	E	F	G	J	L
M	N	O	P	Q	R	U	W
X	Y	Z	0	1	2	3	4
4	6	7	8	9	!	#	\$
%	^	&	*	(	)	-	+
=	{	}	[	]	\		;
:	'	.	<	>	/	,	?

The cipher text is converted to their ASCII codes in decimal and then to corresponding binary values of 7 bits. Finally a Linear Shift Register is applied to get the final cipher text. However, lowercase letter cannot be handled by the algorithm proposed Srivastava and Gupta (2011).

Bhattacharyya *et al.*, (2014), extended play fair algorithm based on the use of a  $10 \times 9$  matrix of letters constructed using a keyword “MONARCHY”. The 10 x 9 matrix contains almost all the printable characters. This includes lowercase and uppercase alphabets, punctuation marks, numbers and special characters. The matrix is constructed by filling in the letters, numbers or special characters of the keyword from left to right and from top to bottom, and the filling in the remainder of the matrix with the remaining letters in alphabetic order and digits in ascending order from 0 to 9 and special characters. The upper case alphabets are placed first then the lower case

alphabets following the digits 0 to 9 can be placed next cells of the lower case alphabet z in an ascending order. And finally the special characters which are arranged in an order. However, repeating plaintext letters in the same pair are separated with X as filler letter, therefore causing ambiguity to the system and also string with odd length of text was not handled.

Table 2.7: Modify Playfair 10×9 Matrix (Bhattacharyya *et al.*, 2014)

M	O	N	A	R	C	H	Y	B	D
E	F	G	I	J	K	L	P	Q	S
T	U	V	W	X	Z	a	b	c	d
e	F	g	h	I	J	k	l	n	m
o	P	q	r	S	T	u	v	w	x
y	Z	0	1	2	3	4	5	6	7
8	9	~	,	.	/	;	“	\	
<	>	?	:	{	}	-	=	!	@
#	\$	%	^	&	*	(	)	_	+

Oduay and Baraa (2013) extends the play fair algorithm based on the use of  $11 \times 11$  matrix of letters constructed using a keyword “{NAWROZow@Duhk!}”. It contains almost all the printable characters. This includes lowercase and uppercase alphabets, punctuation marks, numbers and special characters. The matrix is constructed by filling in the letters, numbers or special characters of the keyword from left to right and from top to bottom, and the filling in the remainder of the matrix with the remaining letters in alphabetic order and digits in ascending order form 0 to 9 and special characters. In this system, not only the alphabet in both cases (upper and lower) encrypts but also the numerals, special characters and an extended special characters. At encryption time, the

symbol “■” is being used to provide space between two words. Moreover, “#” is used as filler character in order to separate two alphabets if they are repeated in pair. “#” was also used to put at the end if the number of plaintext characters is an odd number. However, during the decryption time symbol “#” was suppressed.

Table 2.8: 11×11 Playfair matrix (Oduay and Baraa 2013)

{	N	A	W	R	O	Z	o	W	@	D
u	H	k	!	}	B	C	E	F	G	H
I	J	K	L	M	P	Q	S	T	U	V
X	Y	a	b	c	d	e	f	G	i	j
l	M	n	p	q	r	s	t	V	x	y
z	0	1	2	3	4	5	6	7	8	9
#	\$	%	^	&	*	(	)	_	=	+
[	]	;	‘	:	“	\	,	.	/	<
>	?	£	¥	α	β	π	σ	μ	τ	∞
±	≥	≤	÷	Ç	â	ä	à	Å	ç	ê
ë	È	ï	î	ì	Ä	Å	É	Æ	Æ	■

## 2.11 Literature Gap

Most existing works extend the playfair cipher by increasing the size of the matrix, but maintaining the key generation structure from right to left and bottom up. However, the proposed work extends the traditional playfair cipher and also modifies the key generation using rhotrix.

## CHAPTER THREE

### DESIGN OF AN IMPROVED PLAYFAIR CRYPTOSYSTEM USING RHOTRIX

#### 3.1 Introduction

This chapter starts with the method involved in carrying out the research. This involves the research design, algorithm of the proposed system, the system block diagram and finally flow chart of the system, which serves as the plan for system implementation.

#### 3.2 Design Consideration

The design process is based on the following main considerations:

- a. Large set of character support, thereby allowing encryption / decryption of most type of text files.
- b. Use of binary file mode (equivalent 8-bit ASCII code) to encrypt / decrypt any type of plaintext.
- c. Rhotrix will be used instead of matrix, which is a new paradigm of matrix theory of rectangular arrays.
- d. The size of the rhotrix will be set to  $11 \times 11$  to make it compatible with the  $[n \times (n-1)]$  mode of rhotrix representation.
- e.  $11 \times 11$  diamond structure covers 121 characters, which actually allow an extended support for Unicode values.
- f. Further to incorporate randomness and to eliminate the chances of being attacked by frequency testing of character occurrence, the rhotrix is to be rotated / rearranged in a Half-Transpose manner.
- g. Simple time efficient operations are to be used in each step to reduce the time complexity to  $O(n)$  [ $n$ =size of the plaintext].

- h. Use of simple operations to make the proposed algorithm an efficient lightweight cipher.

### **3.3 System Requirements**

The system was developed in java because of platform independence and availability of the language as open source. The following represents the hardware and software requirements of the system

- a. Personal Computer
- b. A minimum of 512MB RAM, Pentium 2 processor and 400GB hard disk
- c. Uninterruptible Power Supply (UPS)
- d. JDK 6.0 and above
- e. Any Java IDE (eclipse, NetBeans, e.t.c)
- f. Windows 7 operating system

#### **3.3.1 Functional Requirements**

Functional requirements are requirements that give the description of what the system should do. The System should:

- a. Represent the key matrix using rhotrix.
- b. Support up to 121 different characters.
- c. Encrypt the plaintext using playfair algorithm.
- d. Decrypt the cipher text using playfair algorithm.

#### **3.3.2 Non-Functional Requirements**

These are the requirements that give a detailed specification of how the system supposed to operate/ behave. It is also the system's quality characteristics or quality attributes. The non-functional requirements of the end product are:

1. Availability: As a secured cryptic application, it should always be available, accessible and usable as long as it is needed.
2. Security: It should allow only signed-in users to interact with it, and it should effectively protect users' data from eavesdropping and other security attacks.
3. Performance: the tool is expected to be at least 95% efficient on all Operating Systems.
4. Accuracy: The tool works in such a way that, users should be able to accurately and effectively encrypt, decrypt, send and receive messages.

### 3.4 Transforming Rhotrix to Couple Matrix

The method of converting a rhotrix to a special matrix called "coupled matrix" was described in section 2.8. This idea was used to solve systems of  $n \times n$  and  $(n-1) \times (n-1)$  matrix problems simultaneously. Given two matrices of size  $n \times n$  and  $(n-1) \times (n-1)$  representation, it can be converted to the Rhomboidal structure of a rhotrix as follows:

1. Given  $n=6$ ; resulted in two Matrices  $A[6 \times 6]$  and  $B[5 \times 5]$

$$A \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix} \text{ and } B \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} \end{bmatrix} \dots (3.1)$$



3. Now, fill up the empty spaces with 0s for a standard matrix

$$\begin{bmatrix}
 \mathbf{a}_{11} & \mathbf{0} & \mathbf{a}_{12} & \mathbf{0} & \mathbf{a}_{13} & \mathbf{0} & \mathbf{a}_{14} & \mathbf{0} & \mathbf{a}_{15} & \mathbf{0} & \mathbf{a}_{16} \\
 \mathbf{0} & \mathbf{b}_{11} & \mathbf{0} & \mathbf{b}_{12} & \mathbf{0} & \mathbf{b}_{13} & \mathbf{0} & \mathbf{b}_{14} & \mathbf{0} & \mathbf{b}_{15} & \mathbf{0} \\
 \mathbf{a}_{21} & \mathbf{0} & \mathbf{a}_{22} & \mathbf{0} & \mathbf{a}_{23} & \mathbf{0} & \mathbf{a}_{24} & \mathbf{0} & \mathbf{a}_{25} & \mathbf{0} & \mathbf{a}_{26} \\
 \mathbf{0} & \mathbf{b}_{21} & \mathbf{0} & \mathbf{b}_{22} & \mathbf{0} & \mathbf{b}_{23} & \mathbf{0} & \mathbf{b}_{24} & \mathbf{0} & \mathbf{b}_{25} & \mathbf{0} \\
 \mathbf{a}_{31} & \mathbf{0} & \mathbf{a}_{32} & \mathbf{0} & \mathbf{a}_{33} & \mathbf{0} & \mathbf{a}_{34} & \mathbf{0} & \mathbf{a}_{35} & \mathbf{0} & \mathbf{a}_{36} \\
 \mathbf{0} & \mathbf{b}_{31} & \mathbf{0} & \mathbf{b}_{32} & \mathbf{0} & \mathbf{b}_{33} & \mathbf{0} & \mathbf{b}_{34} & \mathbf{0} & \mathbf{b}_{35} & \mathbf{0} \\
 \mathbf{a}_{41} & \mathbf{0} & \mathbf{a}_{42} & \mathbf{0} & \mathbf{a}_{43} & \mathbf{0} & \mathbf{a}_{44} & \mathbf{0} & \mathbf{a}_{45} & \mathbf{0} & \mathbf{a}_{46} \\
 \mathbf{0} & \mathbf{b}_{41} & \mathbf{0} & \mathbf{b}_{42} & \mathbf{0} & \mathbf{b}_{43} & \mathbf{0} & \mathbf{b}_{44} & \mathbf{0} & \mathbf{b}_{45} & \mathbf{0} \\
 \mathbf{a}_{51} & \mathbf{0} & \mathbf{a}_{52} & \mathbf{0} & \mathbf{a}_{53} & \mathbf{0} & \mathbf{a}_{54} & \mathbf{0} & \mathbf{a}_{55} & \mathbf{0} & \mathbf{a}_{56} \\
 \mathbf{0} & \mathbf{b}_{51} & \mathbf{0} & \mathbf{b}_{52} & \mathbf{0} & \mathbf{b}_{53} & \mathbf{0} & \mathbf{b}_{54} & \mathbf{0} & \mathbf{b}_{55} & \mathbf{0} \\
 \mathbf{a}_{61} & \mathbf{0} & \mathbf{a}_{62} & \mathbf{0} & \mathbf{a}_{63} & \mathbf{0} & \mathbf{a}_{64} & \mathbf{0} & \mathbf{a}_{65} & \mathbf{0} & \mathbf{a}_{66}
 \end{bmatrix} \dots (3.4)$$

The equation 3.4 is the improved rhotrix-matrix structure that will be used in key generation of the modified playfair cipher.

### 3.5 Flow Chart

The system flow chart for the proposed modified playfair cryptosystem using rhotrix starts with a plaintext and a keyword entered by the user. The key rhotrix is generated based on 11×11 rhotrix concatenated with both the key and filler characters according to insertion of rows and columns in even or odd rows or columns. The encryption process apply the conventional playfair rules on the plaintext using the generated key-rhotrix to get the cipher text. Encryption process stops when final ciphertext of the plaintext is displayed. The flowchart is further illustrated in figure 3.1

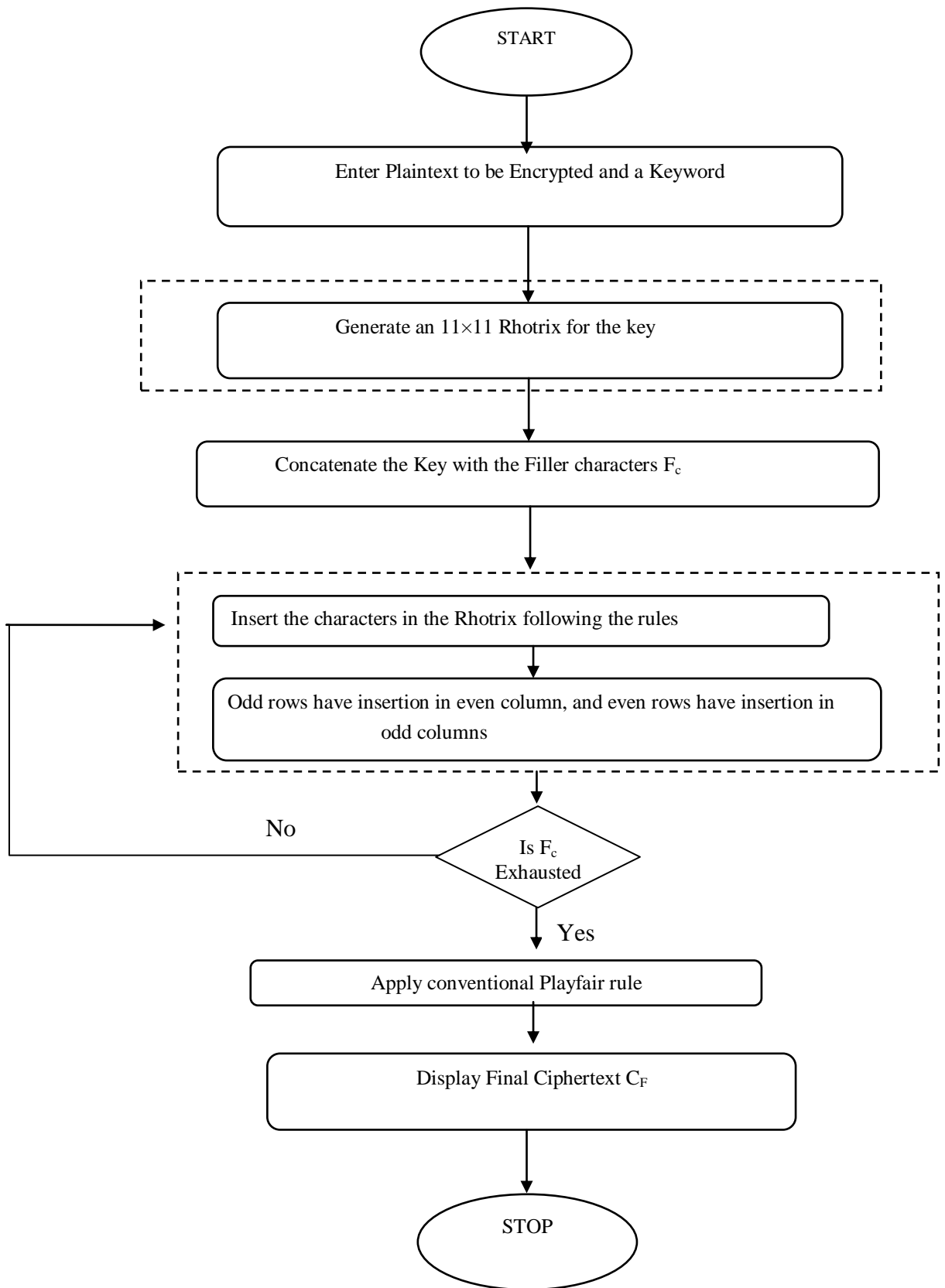


Figure 3.1: Proposed system Flowchart.

### 3.6 System Block Diagram

The system block diagram is used to model the algorithmic process of the system. The block diagram for the modified Playfair cryptosystem is illustrated in Figure 3.3:

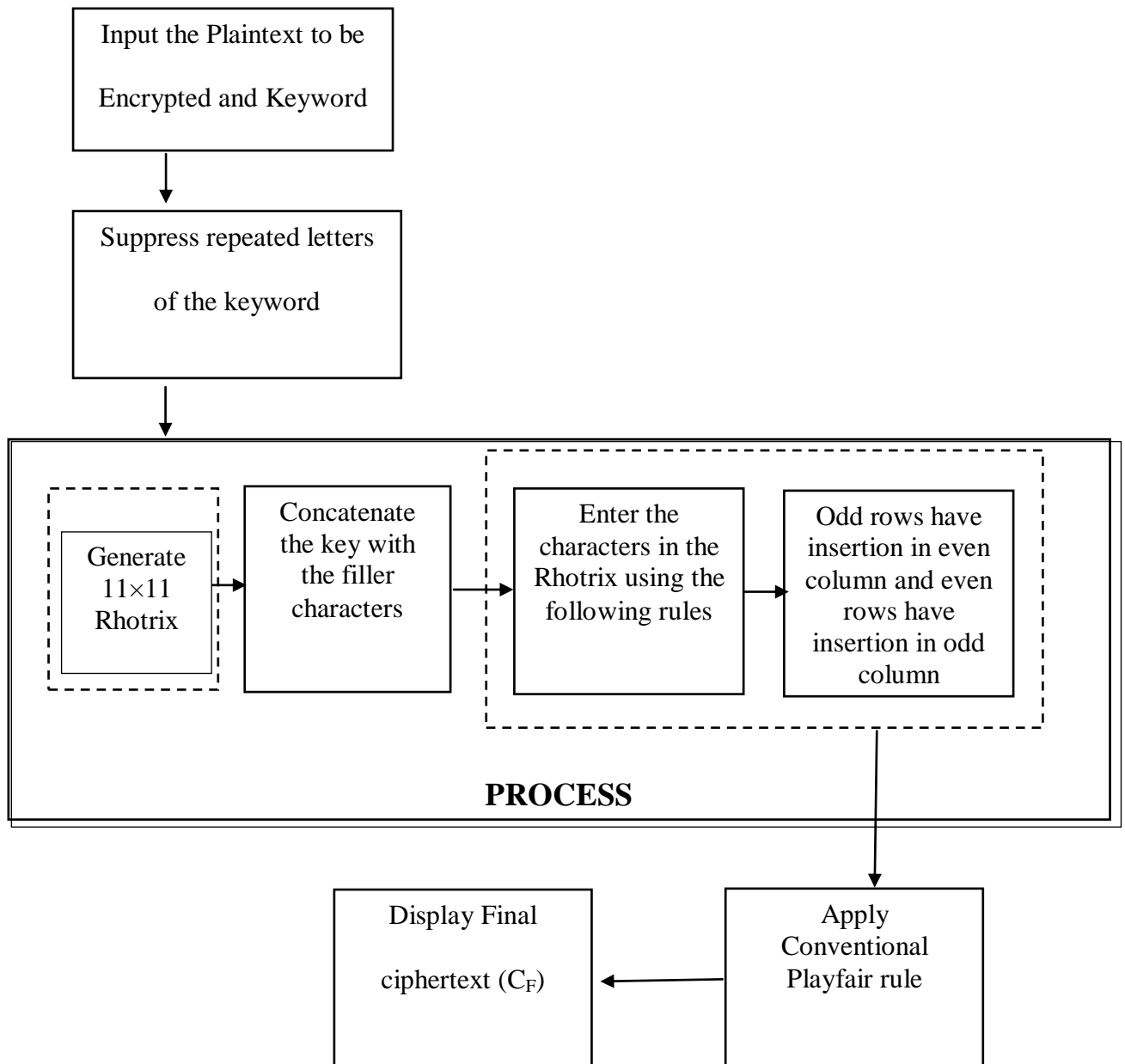


Figure 3.2: System Block Diagram

From Figure 3.3, the processes that were merged to modify the key generation of playfair cipher are encircled with dotted lines.

### 3.7 Algorithm for the Proposed System

This algorithm consists of two phases- key rhotrix formation and the encryption phase.

#### 3.7.1 Proposed Key Formation System Algorithm

Step One: Read the input Key  $K_f$ .

Step Two: Prepare Filler characters  $F_c$

Step Three: Create the Key table  $K$  of  $11 \times 11$  rhotrix

Step Four: For each character in the key  $K_f$  and is not in Key Table  $K$ , do

- a) if row is even, insert character at odd columns of  $K$
- b) else if row is odd, insert character at even columns of  $K$

Step Five: Continue to fill up the key table

- a) if not end of  $K$ , then add unused filler character from  $F_c$  into  $K$  according to STEP 4(a) and (b)
- b) if end of  $K$ , then add unused characters  $F_c$  in empty spaces from beginning starting from left to right, top to bottom
- c) End the key table

Step Six: Return  $K$  as key rhotrix.

#### For Example:

The keyword “a.rukayyat07@gmail.com” is used to form the key generation Process for the new system.

Table 3.1: Proposed Key table generation

3	a	4	.	5	r	6	u	8	k	9
y	,	t	?	0	;	7	:	@	\	g
"	m	'	i	#	l	\$	c	&	o	*
q	(	ë	)	A	+	B	-	C	/	D
>	E	£	F	¥	G	<b>β</b>	H	<b>π</b>	I	<b>σ</b>
J	<b>μ</b>	K	<b>τ</b>	L	∞	M	±	N	≥	O
≤	P	÷	Q	Ç	R	â	S	ä	T	à
U	å	V	ç	W	ê	X	è	Y	ï	Z
î	b	ì	d	Ä	e	Å	f	É	h	æ
j	■	n	%	p	!	q	{	s	}	v
^	w	~	x	Ö	z	Ø	l	Ñ	2	η

Table 3.1 is constructed by filling in the letters of the keyword

“a.rukayyat07@gmail.com”, according to proposed key formation algorithm and the remainder of the table is filled with the filler characters

“qëABCDEF GHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789.,?:\“@#&\*()+-/>£¥**αβπσμτ**∞±≥÷ Ç â ä à å ç ê è ì î ï Ä Å É æ ■ % ! { } ^ ~ Ö Ø Ñ η”.

The upper case alphabets is filled first then the lower case alphabets, digits in ascending order form 0 to 9 and finally special characters .

### 3.7.2 Algorithm for Encryption

The procedure for the Encryption design of the proposed system is carefully described as follows:

Step One: Read the plaintext file  $P_f$ .

Step Two: Generate Key Table  $K$  using algorithm in section 3.7.1

Step Three: Take a character from the Plaintext

- a) Compare character with its neighbour
- b) When there is a repetition or text with odd length, add “*ø*” (Special filler character) in-between them and at the end of plaintext respectively.

Step Four: Formulate the plaintext into Digraph.

Step Five: Take a Digraph

- a) Get the position of each character from the key rhatrix  $K$
- b) If both characters are from the same row, shift columns to right
- c) If both characters are from the same column, shift rows down
- d) otherwise, letters in Digraph that form rectangle, swap first column with second column

Step six: Perform table look-up to get the encoded Di-graph.

Step Seven: Repeat Step 5 and 6 until there is no Di-graph remaining in the plaintext.

Step Eight: Display final ciphertext.

## CHAPTER FOUR

### RESULTS AND ANALYSIS

#### 4.1 Introduction

The proposed modified Playfair algorithm using rhotrix was implemented on java platform and numbers of tests are considered to observe the encryption efficiency in terms of certain parameters like letter frequency, index of coincidence, character length, and time. An 11×11 matrix Playfair cipher was used for comparison with respect to the mentioned parameters. Some experimental results are covered in this chapter.

#### 4.2 Snapshots for the Implementation

The snapshot for the entire encryption process is as shown in Figure 4.1:

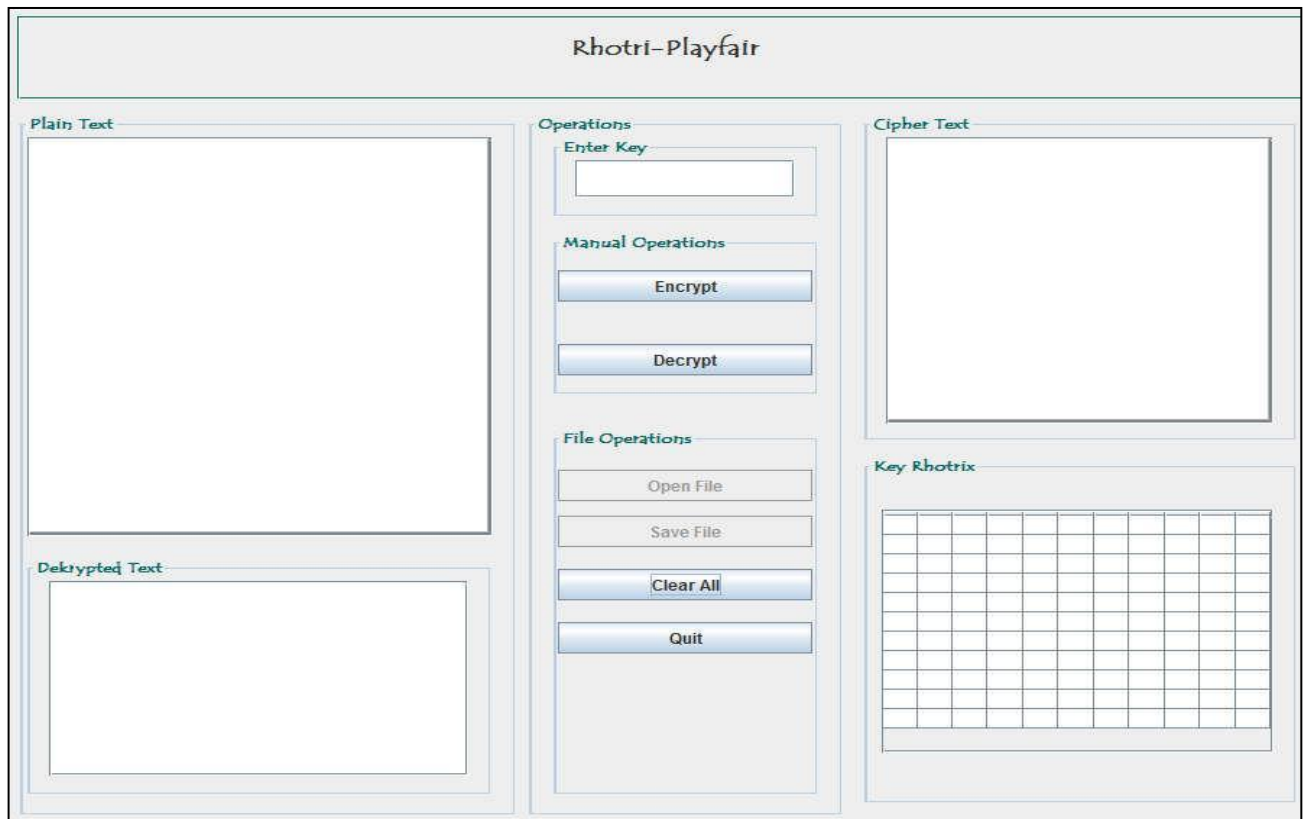


Figure 4.1: Environment for Cryptographic Process

The Software system that implemented the new system is as depicted in figure 4.1. It has three panels; the left, middle and right panels. The left panel has two sub-panels; the upper panel is where the user enters the plaintext to be encrypted. The lower panel is where the deciphered text are displayed when the decryption process is completed. The middle panel allows the user to enter the key to be used for the encryption in the upper panel, the middle panel handles the operations for encryption and decryption, and the lower panel handles file and other related operations. The right panel displays the cipher-text on the top panel when the encryption is completed, and the lower panel displays the matrix generated when eventually the cryptographic process starts.

The first step of the encryption is shown in figure 4.2.

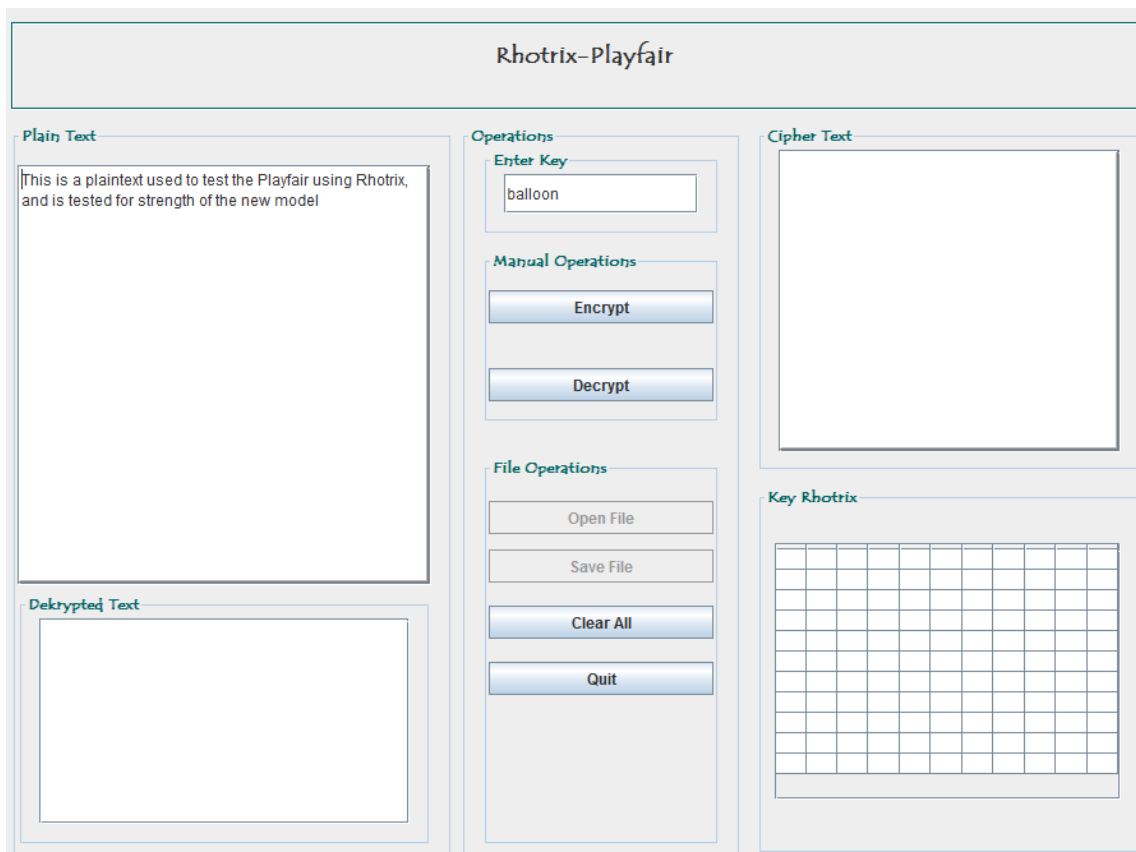


Figure 4.2: Encryption first step

The first step requires the user to enter the plaintext and the key to be used for the cryptographic process, and the “Encrypt” button is pressed. The key entered is then

used to generate the key matrix based on the new rhotrix model. This new model enhances the diffusion and confusion in the key generated in two ways; one: the use of an 11×11 rhotrix that contains up to 121 characters including alphabets, numbers, and punctuations. This increases the support for natural encryption/decryption as most of the characters used in messaging are included. Two: rhotrix is different from matrix representation, and hence makes it more difficult for attacker to break the cipher. This is shown in Figure 4.3.

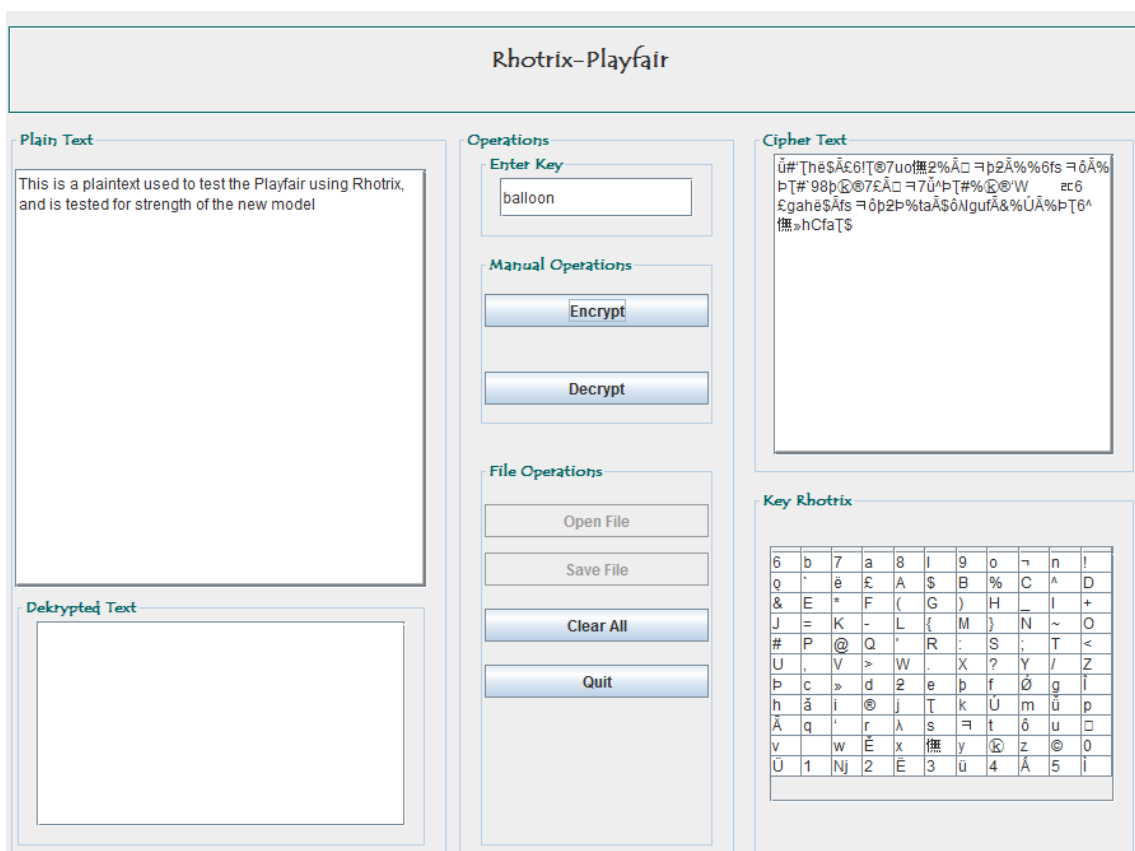


Figure 4.3: Encryption involving the key matrix based on the rhotrix model

The encryption process ends by applying the three known playfair rules on the plaintext using the generated key-matrix to get the cipher text as shown in figure 4.4.

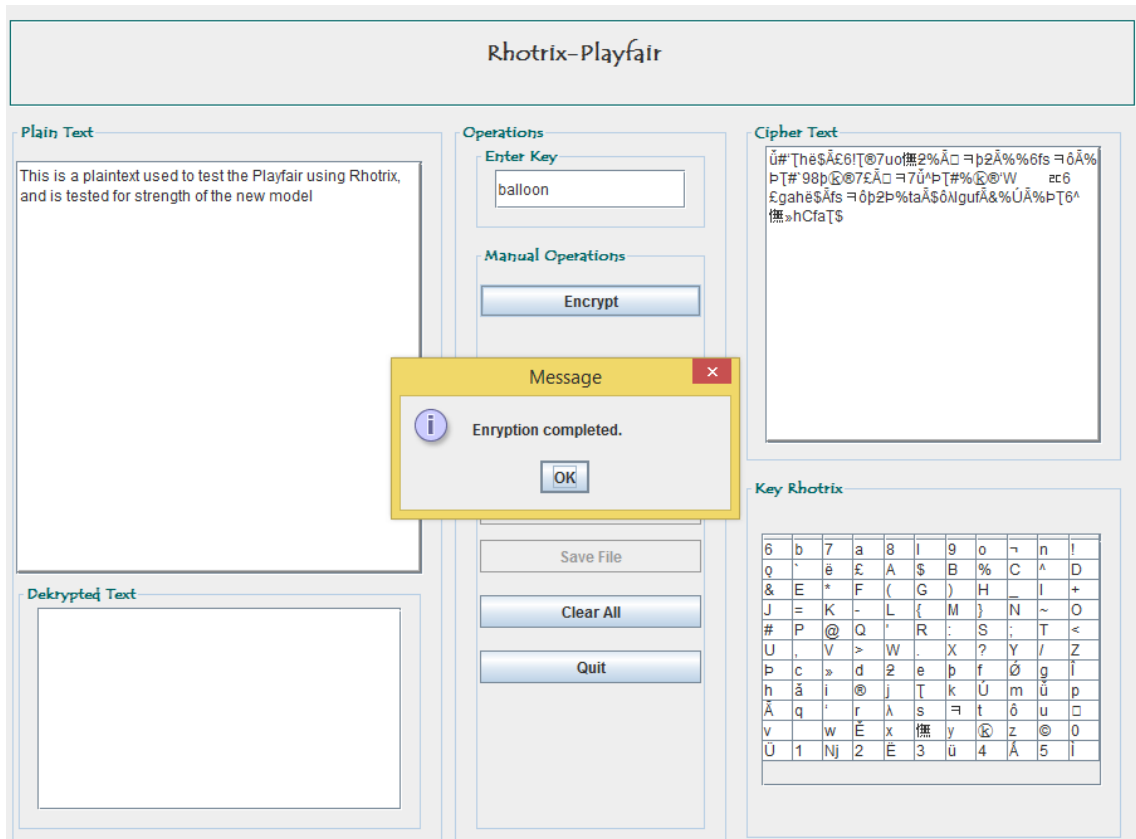


Figure 4.4: Final Encryption phase

The decryption process involves reversing the process in the encryption process to get the plain text from the cipher text as shown in figure 4.5

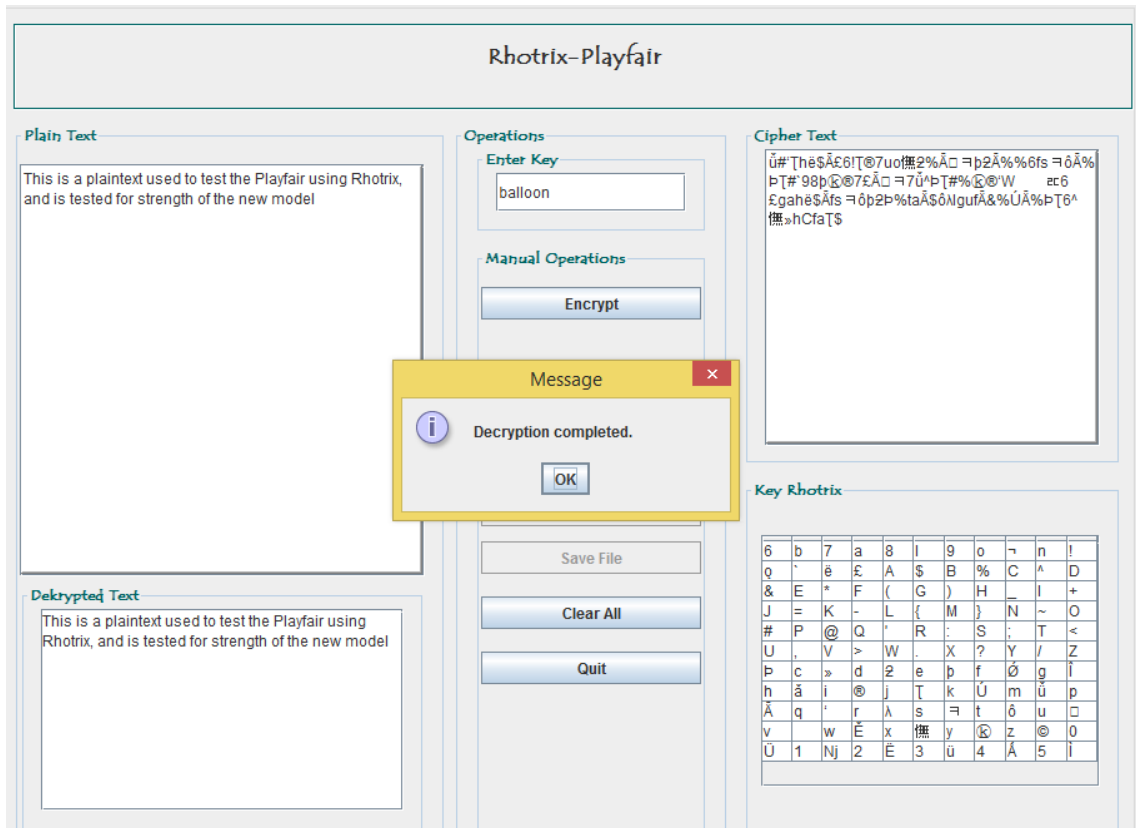


Figure 4.5: Decryption phase

### 4.3 Comparison of Methods and Results

In analysing the efficiency of the proposed cryptosystem, it is necessary to implement a test case using a given plaintext and analyse the output of the old and new system.

Suppose the plaintext is: *“This is a plaintext used to test Playfair using Rhotrix”*

Keyword used: *“balloon”*

The new system uses an 11×11 rhotrix in generating its key. A similar conventional playfair using an 11×11 matrix was implemented in order to compare the two. In most literatures covered Ouday and Baraa (2013); Basu and Ray (2012); Chand and Bhattacharyya (2014); and Amandeep *et al.* (2012), the size of the matrix is what mostly differentiate one playfair implementation to another. Whereas, in this research,

the proposed system was not just changing the size of the matrix, but rather changing the underlined structure from matrix to rhotrix.

THE TEST CASE

<b>The Chosen Key:</b> balloon																																																																																																																																																																																																																																																			
<b>The Plaintext:</b> This is a plaintext used to test the Playfair using Rhotrix, and is tested for strength of the new model																																																																																																																																																																																																																																																			
<b>Keys Generated Using 11×11 Matrix</b>	<b>Keys generated Using 11×11 Rhotrix</b>																																																																																																																																																																																																																																																		
<table border="1"> <tr><td>b</td><td>a</td><td>l</td><td>o</td><td>n</td><td>q</td><td>ë</td><td>A</td><td>B</td><td>C</td><td>D</td></tr> <tr><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td></tr> <tr><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td></tr> <tr><td>c</td><td>d</td><td>e</td><td>f</td><td>g</td><td>h</td><td>i</td><td>j</td><td>k</td><td>m</td><td>p</td></tr> <tr><td>q</td><td>r</td><td>s</td><td>t</td><td>u</td><td>v</td><td>w</td><td>x</td><td>y</td><td>z</td><td>0</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>~</td><td>!</td></tr> <tr><td>'</td><td>£</td><td>\$</td><td>%</td><td>^</td><td>&amp;</td><td>*</td><td>(</td><td>)</td><td>_</td><td>+</td></tr> <tr><td>=</td><td>-</td><td>{</td><td>}</td><td>~</td><td>#</td><td>@</td><td>'</td><td>:</td><td>:</td><td>&lt;</td></tr> <tr><td>,</td><td>&gt;</td><td>.</td><td>?</td><td>/</td><td>»</td><td>»</td><td>»</td><td>»</td><td>»</td><td>»</td></tr> <tr><td>ä</td><td>@</td><td>T</td><td>U</td><td>ü</td><td>Ä</td><td>'</td><td>λ</td><td>≡</td><td>□</td><td>□</td></tr> <tr><td>È</td><td>無</td><td>Ⓚ</td><td>□</td><td>Ü</td><td>Nj</td><td>È</td><td>ü</td><td>Ä</td><td>i</td><td></td></tr> </table>	b	a	l	o	n	q	ë	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	c	d	e	f	g	h	i	j	k	m	p	q	r	s	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7	8	9	~	!	'	£	\$	%	^	&	*	(	)	_	+	=	-	{	}	~	#	@	'	:	:	<	,	>	.	?	/	»	»	»	»	»	»	ä	@	T	U	ü	Ä	'	λ	≡	□	□	È	無	Ⓚ	□	Ü	Nj	È	ü	Ä	i		<table border="1"> <tr><td>6</td><td>b</td><td>7</td><td>a</td><td>8</td><td>l</td><td>9</td><td>o</td><td>~</td><td>n</td><td>!</td></tr> <tr><td>q</td><td>'</td><td>ë</td><td>£</td><td>A</td><td>\$</td><td>B</td><td>%</td><td>C</td><td>^</td><td>D</td></tr> <tr><td>&amp;</td><td>E</td><td>*</td><td>F</td><td>(</td><td>G</td><td>)</td><td>H</td><td>_</td><td>I</td><td>+</td></tr> <tr><td>J</td><td>=</td><td>K</td><td>-</td><td>L</td><td>{</td><td>M</td><td>}</td><td>N</td><td>~</td><td>O</td></tr> <tr><td>#</td><td>P</td><td>@</td><td>Q</td><td>'</td><td>R</td><td>:</td><td>S</td><td>:</td><td>T</td><td>&lt;</td></tr> <tr><td>U</td><td>,</td><td>V</td><td>&gt;</td><td>W</td><td>.</td><td>X</td><td>?</td><td>Y</td><td>/</td><td>Z</td></tr> <tr><td>p</td><td>c</td><td>»</td><td>d</td><td>2</td><td>e</td><td>p</td><td>f</td><td>Ø</td><td>g</td><td>ï</td></tr> <tr><td>h</td><td>ä</td><td>i</td><td>@</td><td>j</td><td>T</td><td>k</td><td>U</td><td>m</td><td>ü</td><td>p</td></tr> <tr><td>Ä</td><td>q</td><td>'</td><td>r</td><td>λ</td><td>s</td><td>≡</td><td>t</td><td>□</td><td>u</td><td>□</td></tr> <tr><td>v</td><td></td><td>w</td><td>È</td><td>x</td><td>無</td><td>y</td><td>Ⓚ</td><td>z</td><td>□</td><td>0</td></tr> <tr><td>Ü</td><td>1</td><td>Nj</td><td>2</td><td>È</td><td>3</td><td>ü</td><td>4</td><td>Ä</td><td>5</td><td>i</td></tr> </table>	6	b	7	a	8	l	9	o	~	n	!	q	'	ë	£	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	p	c	»	d	2	e	p	f	Ø	g	ï	h	ä	i	@	j	T	k	U	m	ü	p	Ä	q	'	r	λ	s	≡	t	□	u	□	v		w	È	x	無	y	Ⓚ	z	□	0	Ü	1	Nj	2	È	3	ü	4	Ä	5	i
b	a	l	o	n	q	ë	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																																																																																																																																																																																																																																									
c	d	e	f	g	h	i	j	k	m	p																																																																																																																																																																																																																																									
q	r	s	t	u	v	w	x	y	z	0																																																																																																																																																																																																																																									
1	2	3	4	5	6	7	8	9	~	!																																																																																																																																																																																																																																									
'	£	\$	%	^	&	*	(	)	_	+																																																																																																																																																																																																																																									
=	-	{	}	~	#	@	'	:	:	<																																																																																																																																																																																																																																									
,	>	.	?	/	»	»	»	»	»	»																																																																																																																																																																																																																																									
ä	@	T	U	ü	Ä	'	λ	≡	□	□																																																																																																																																																																																																																																									
È	無	Ⓚ	□	Ü	Nj	È	ü	Ä	i																																																																																																																																																																																																																																										
6	b	7	a	8	l	9	o	~	n	!																																																																																																																																																																																																																																									
q	'	ë	£	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																																																																																																																																																																																																																																									
p	c	»	d	2	e	p	f	Ø	g	ï																																																																																																																																																																																																																																									
h	ä	i	@	j	T	k	U	m	ü	p																																																																																																																																																																																																																																									
Ä	q	'	r	λ	s	≡	t	□	u	□																																																																																																																																																																																																																																									
v		w	È	x	無	y	Ⓚ	z	□	0																																																																																																																																																																																																																																									
Ü	1	Nj	2	È	3	ü	4	Ä	5	i																																																																																																																																																																																																																																									
gUwehëlvleDedëuosjovvtfevonëfstuvoifUbolktdeä vvtëgnheUH4dw2qëlgahëlvfstufehotavluslgufvJHt voifëqsihCfasG	ü#‘T hë\$Ä£6!T@7uo無2%Ä□ ⇒ p2Ä%%6fs ⇒ □ Ä%P T#’98pⓀ@7£Ä□ ⇒ 7ü^P T#%Ⓚ@‘W 卄 6£ga hë\$Äfs ⇒ □ p2P%taÄ\$□ λlgufÄ&%ÜÄ%P T6^無 »hCfa T\$																																																																																																																																																																																																																																																		

Figure 4.6: Key generation and ciphertext of existing and proposed system

Another test case showing the key generation and ciphertext of the existing and proposed system is shown in figure 4.7

<b>The Chosen Key:</b> Abubakar23@gmail.com																																																																																																																																																																																																																																																			
<b>The Plaintext:</b> The left panel has two sub-panels; the upper panel is where the user enters the plaintext to be encrypted. The lower panel is where the deciphered text are displayed when the decryption process is completed. The middle panel allows the user to enter the key to be used for the encryption in the upper panel, the middle panel handles the operations for encryption and decryption, and the lower panel handles file and other related operations.																																																																																																																																																																																																																																																			
<b>Keys Generated Using 11×11 Matrix</b>	<b>Keys generated Using 11×11 Rhoatrix</b>																																																																																																																																																																																																																																																		
<table border="1"> <tr><td>A</td><td>b</td><td>u</td><td>a</td><td>k</td><td>r</td><td>2</td><td>3</td><td>@</td><td>g</td><td>m</td></tr> <tr><td>i</td><td>l</td><td>.</td><td>c</td><td>o</td><td>q</td><td>ë</td><td>B</td><td>C</td><td>D</td><td>E</td></tr> <tr><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td></tr> <tr><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td><td>d</td></tr> <tr><td>e</td><td>f</td><td>h</td><td>j</td><td>n</td><td>p</td><td>q</td><td>s</td><td>t</td><td>v</td><td>w</td></tr> <tr><td>x</td><td>y</td><td>z</td><td>0</td><td>1</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td>~</td><td>!</td><td>`</td><td>£</td><td>\$</td><td>%</td><td>^</td><td>&amp;</td><td>*</td><td>(</td><td>)</td></tr> <tr><td>=</td><td>+</td><td>=</td><td>-</td><td>{</td><td>}</td><td>~</td><td>#</td><td>'</td><td>:</td><td>;</td></tr> <tr><td>&lt;</td><td>,</td><td>&gt;</td><td>?</td><td>/</td><td>p</td><td>&gt;</td><td>2</td><td>p</td><td>Ø</td><td>l</td></tr> <tr><td>ä</td><td>®</td><td>ŧ</td><td>Ü</td><td>ü</td><td>Ä</td><td>'</td><td>λ</td><td>ø</td><td>ô</td><td>□</td></tr> <tr><td>É</td><td>撫</td><td>Ⓚ</td><td>©</td><td>Ü</td><td>Nj</td><td>È</td><td>ü</td><td>Á</td><td>l</td><td></td></tr> </table>	A	b	u	a	k	r	2	3	@	g	m	i	l	.	c	o	q	ë	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	d	e	f	h	j	n	p	q	s	t	v	w	x	y	z	0	1	4	5	6	7	8	9	~	!	`	£	\$	%	^	&	*	(	)	=	+	=	-	{	}	~	#	'	:	;	<	,	>	?	/	p	>	2	p	Ø	l	ä	®	ŧ	Ü	ü	Ä	'	λ	ø	ô	□	É	撫	Ⓚ	©	Ü	Nj	È	ü	Á	l		<table border="1"> <tr><td>4</td><td>A</td><td>5</td><td>b</td><td>6</td><td>u</td><td>7</td><td>a</td><td>8</td><td>k</td><td>9</td></tr> <tr><td>r</td><td>~</td><td>2</td><td>!</td><td>3</td><td>`</td><td>@</td><td>£</td><td>g</td><td>\$</td><td>m</td></tr> <tr><td>%</td><td>i</td><td>^</td><td>l</td><td>&amp;</td><td>.</td><td>*</td><td>c</td><td>(</td><td>o</td><td>)</td></tr> <tr><td>q</td><td>_</td><td>ë</td><td>+</td><td>B</td><td>=</td><td>C</td><td>-</td><td>D</td><td>{</td><td>E</td></tr> <tr><td>}</td><td>F</td><td>~</td><td>G</td><td>#</td><td>H</td><td>'</td><td>l</td><td>:</td><td>J</td><td>;</td></tr> <tr><td>K</td><td>&lt;</td><td>L</td><td>,</td><td>M</td><td>&gt;</td><td>N</td><td>?</td><td>O</td><td>/</td><td>P</td></tr> <tr><td>p</td><td>Q</td><td>&gt;</td><td>R</td><td>2</td><td>S</td><td>p</td><td>T</td><td>Ø</td><td>U</td><td>l</td></tr> <tr><td>V</td><td>ä</td><td>W</td><td>@</td><td>X</td><td>ŧ</td><td>Y</td><td>Ü</td><td>Z</td><td>ü</td><td>d</td></tr> <tr><td>Ä</td><td>e</td><td>'</td><td>f</td><td>λ</td><td>h</td><td>≡</td><td>j</td><td>ô</td><td>n</td><td>□</td></tr> <tr><td>p</td><td></td><td>q</td><td>É</td><td>s</td><td>撫</td><td>t</td><td>Ⓚ</td><td>v</td><td>©</td><td>w</td></tr> <tr><td>Ü</td><td>x</td><td>Nj</td><td>y</td><td>È</td><td>z</td><td>ü</td><td>0</td><td>Á</td><td>1</td><td>l</td></tr> </table>	4	A	5	b	6	u	7	a	8	k	9	r	~	2	!	3	`	@	£	g	\$	m	%	i	^	l	&	.	*	c	(	o	)	q	_	ë	+	B	=	C	-	D	{	E	}	F	~	G	#	H	'	l	:	J	;	K	<	L	,	M	>	N	?	O	/	P	p	Q	>	R	2	S	p	T	Ø	U	l	V	ä	W	@	X	ŧ	Y	Ü	Z	ü	d	Ä	e	'	f	λ	h	≡	j	ô	n	□	p		q	É	s	撫	t	Ⓚ	v	©	w	Ü	x	Nj	y	È	z	ü	0	Á	1	l
A	b	u	a	k	r	2	3	@	g	m																																																																																																																																																																																																																																									
i	l	.	c	o	q	ë	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	d																																																																																																																																																																																																																																									
e	f	h	j	n	p	q	s	t	v	w																																																																																																																																																																																																																																									
x	y	z	0	1	4	5	6	7	8	9																																																																																																																																																																																																																																									
~	!	`	£	\$	%	^	&	*	(	)																																																																																																																																																																																																																																									
=	+	=	-	{	}	~	#	'	:	;																																																																																																																																																																																																																																									
<	,	>	?	/	p	>	2	p	Ø	l																																																																																																																																																																																																																																									
ä	®	ŧ	Ü	ü	Ä	'	λ	ø	ô	□																																																																																																																																																																																																																																									
É	撫	Ⓚ	©	Ü	Nj	È	ü	Á	l																																																																																																																																																																																																																																										
4	A	5	b	6	u	7	a	8	k	9																																																																																																																																																																																																																																									
r	~	2	!	3	`	@	£	g	\$	m																																																																																																																																																																																																																																									
%	i	^	l	&	.	*	c	(	o	)																																																																																																																																																																																																																																									
q	_	ë	+	B	=	C	-	D	{	E																																																																																																																																																																																																																																									
}	F	~	G	#	H	'	l	:	J	;																																																																																																																																																																																																																																									
K	<	L	,	M	>	N	?	O	/	P																																																																																																																																																																																																																																									
p	Q	>	R	2	S	p	T	Ø	U	l																																																																																																																																																																																																																																									
V	ä	W	@	X	ŧ	Y	Ü	Z	ü	d																																																																																																																																																																																																																																									
Ä	e	'	f	λ	h	≡	j	ô	n	□																																																																																																																																																																																																																																									
p		q	É	s	撫	t	Ⓚ	v	©	w																																																																																																																																																																																																																																									
Ü	x	Nj	y	È	z	ü	0	Á	1	l																																																																																																																																																																																																																																									
<p>jSipfihvK4jkifp.j3pCEnpBauj}jkif#wpCjfr.qfQKrp  pf.ëeBpEjfpApCjfr.tfQKfpvfp3pCjfk4bceovfe7pC  qëfApioj4bqvQwcëjSip.qQKrpjf.ëeBpEjfpApCjfv  EijejfpAEVvfe7rcpAVEeBQf0bQwpEjfovpjipw  QaQf4CeJ1K4qkjiBpeBëokEqffvQwcëjSipEAERi  prjpf.ëcb.qpCjfr.tfQKcnpivAppCjfrxpfCQëfAr.t  fEVlnQKvjipfpaQf4CeJ1ëlopvjiphrqfQKrpjfg@pCj  frEQEERiprjpf.ëujUwfiBpvjipnQApj@lOptplkQpio  j4bqvlQopjkEVwQaQf4CeJ1lBjkEVvjip.qQKrpjf.ë  ujUwfiBplefircUwëqvjApKQifj@QwëQf2kCeJ1B  h</p>	<p>S_ÄfiË }ÜjkifÄ=(K6pC)©pB76(KQjkif#wpC }  '4=Ä % }4(K'+'% &amp;pE } 'Ä-pC } '4=λ % }  ' } } p3pC } ' }ÜbceQ } %fAÄ_ojÜ!  % (ä= %jS_Ä&amp;)' }4(K'+'% &amp;pE } 'Ä  -pC } 'VEij % } 'Ä-EV } %ü4-  Ä-VE %&amp;%Ë0bä }pE } 'Ä }撫_Ä }äf%ËÜ* %  { }Ü%\$jiBp %&amp;%-  \$)%Ë % } ä= %jS_Ä)-)®_Ä4(K'+'%cb&amp;)p撫p  C } '4=λ % } *©Ä_© } -ÄpC } '4{xfpC{ %fA4=  λ %EVln% } }撫_Ä' }f%ËÜ* % {©%_ {Ä }撫_  Ä撫4Ä % }4(K'+RpC } 'rEä))®_Ä4(K'+'%u  jü }fiBp }撫_Ä©%-Ä(K7^ )©λÄ+\$%Ä_ojÜ! %  (K^){ÄjkEV }äf%ËÜ* % {©+KjkEV }撫_Ä&amp;)'  % }4(K'+'%ujü }fiBplefi4-  ü } { }撫-Ä }%if(K7ä } {Ä %4f* % {©&amp;撫</p>																																																																																																																																																																																																																																																		

Figure 4.7: Test case showing the Key generation and ciphertext

## 4.4 Graphical Analysis

The ciphertext was further analysed graphically for easy comparison with the various existing methods. This would clearly show the frequencies of characters and how they are dispersed.

### 4.4.1 English Alphabet Frequency

One of the ways attackers use in breaking ciphers is to study the frequencies of the characters in the cipher text. The number of occurrences could give insight on the key used based on the known probability distribution frequencies of English alphabets as stated by Katz and Lindell (2008). This is displayed in the figure 4.7

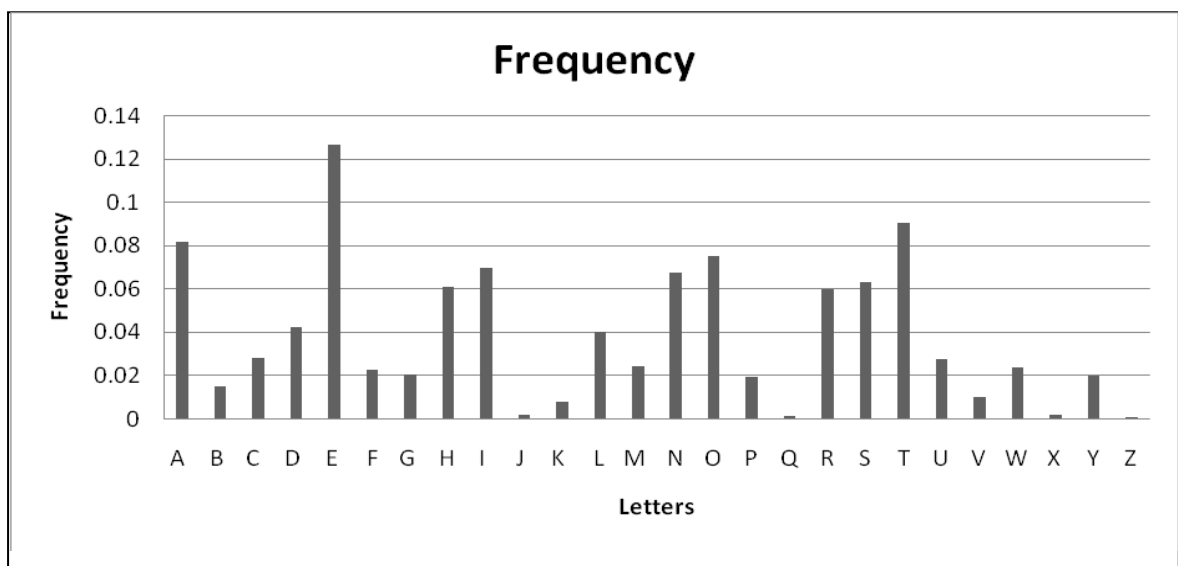


Figure 4.8: Probability Distribution of the English Alphabet Characters

Figure 4.8 shows the most frequent letters distribution of English, such as: letter E is the most frequent letter (about 13%), T is the second most frequent letter (about 9%), and A is the third most frequent letter (about 8%). Appendix A lists the letter frequency distribution of English.

#### 4.4.2 Analysis of Existing and Proposed System

Encryption was performed on the selected test case using the conventional  $11 \times 11$  Playfair cipher using matrix and rhotrix, and a graph of the frequencies of the letters as they appear in the cipher text was shown in figure 4.8.

The Plaintext: This is a plaintext used to test the Playfair using Rhotrix, and is tested for strength of the new model

The cipher text of the existing system is as follows:

gUwehëlvlēDedëuosjovvtfevonëfstuvoifUbolktdeävnvtëgnheUH4dw;zëlgahëlvfstufëhota  
vluslgufvJHtvoifëqsihCfasG

The cipher text of the proposed system is as follows:

'⇒ ühë^λ%6fűÚ7uo Ó&λĚ ÍŎλ&&6fs Ꞥ λ&2ű'\$!-ÎÛÚ7%λĚ 7 \*2ű'&ÛÚ ⇒  
;z6%gahë^λfs Ꞥ ÍŎ2&ta^ Ꞥ □ lgufλ(&Āλ&2ű6\* phCfaű^

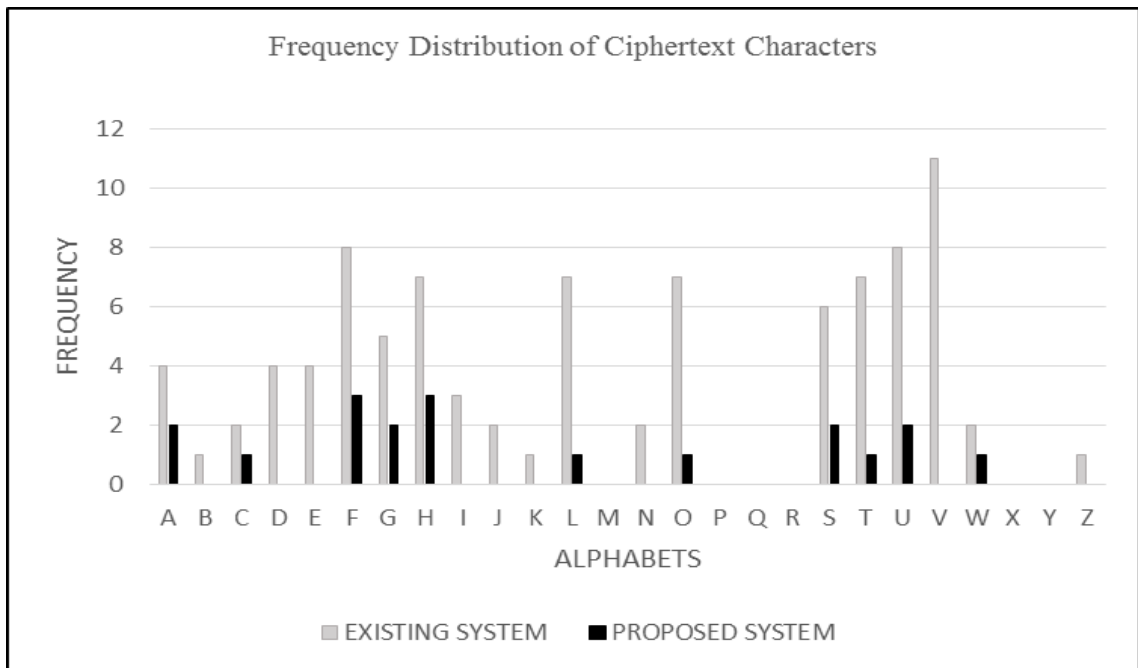


Figure 4.9: Graphical Analysis of Existing and Proposed System

From figure 4.9 the proposed system has lower repetition of English characters in the ciphertext when compared with existing system. Thus, the proposed system has lesser chance of frequency tracking and more safe than existing system.

#### 4.5 Statistical Analysis

Statistical analysis was performed on the ciphertext to compare the performances. The index of coincidence was calculated and compared. To determine the index of coincidence, the probabilities of the frequency of the ciphertext characters are used. This is determined using equation 2.3.

Table 4. 1: Result obtained from calculating index of Coincidence

	Existing System	Proposed System
Index of Coincidences	0.058528	0.002389

From table 4.1, it can be seen that the proposed system has a lower index of coincidence when compare to the existing system. Thus, the lesser the possibility of cryptanalyst to break the encrypted text. The graphically representation is shown in figure 4.9:

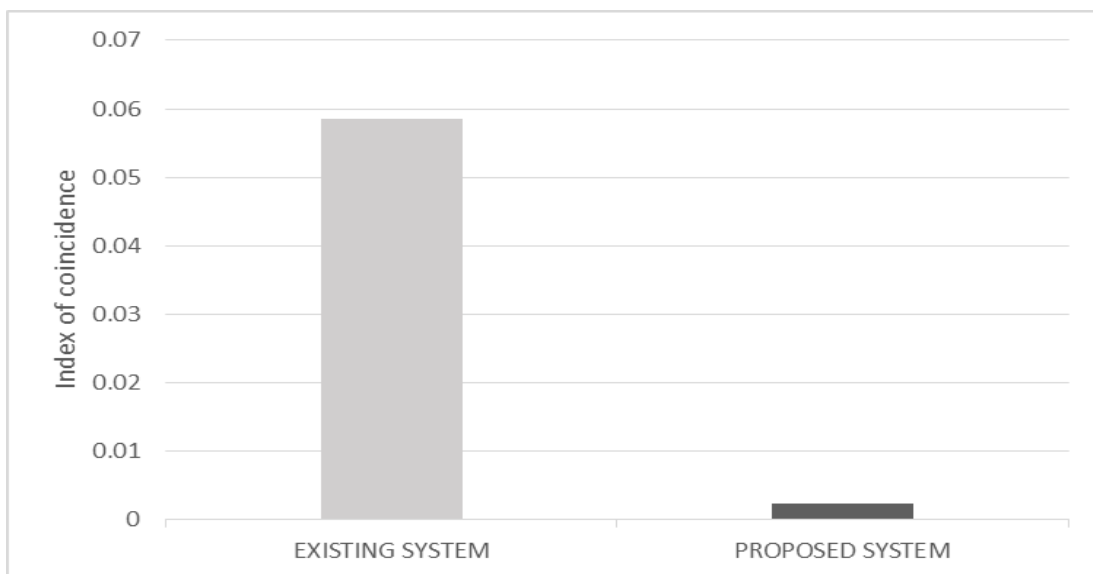


Figure 4.10: Graphical Representation of Index of Coincidence

#### 4.5.1 Character Length

Character length should be very large in order to increase the complexity and security of algorithms. The proposed system was able to increase the number of input characters from 26 to 256 which include Characters that exist within the range of 0 to 31 which are called control characters, while from 32 to 127 are called printable characters, and from 128 to 255 are the extended ASCII codes. The control characters are unprintable characters that are used to control peripherals such as printers, and as such are displayed as blank spaces whenever they exist within the ciphertext, but are known only to the computer internally. This makes information hidden and more secure because the existence of the control characters within the ciphertext gives false information to an adversary trying to get access to unauthorised information.

#### 4.5.2 Brute Force Attack

The proposed system uses a key rhotrix of size  $11 \times 11$  and rotates the rhotrix in half-transpose manner. So the attacker has to find from a  $121!$  ( $=8.094E200$ ) options, for each byte of data, which is quite a huge alternative to search for the proper key/plaintext. So the proposed system is safe from brute force attack.

#### 4.5.3 Space and Time Requirement

The space requirement of this modified Playfair algorithm is also very low. It uses an  $11 \times 11$  rhotrix and a buffer of 121-nibble is required. In addition two more bytes are required as file read/write buffer and rhotrix manipulation buffer. The running time of the algorithm at its worst case scenario is  $O(n^2)$ , where  $n$  is the size of the rhotrix.

## CHAPTER FIVE

### SUMMARY, CONCLUSION AND FUTURE WORK

#### 5.1 Summary

This research was able to improve the playfair cryptosystem using rhotrix by increasing the characters set of the traditional playfair cipher. The playfair square table was redesigned to accommodate more ASCII characters without ignoring the symbol ‘#’ during decipherment, and also providing a new scheme for its key generation. The use of extended character set based on ASCII in the system rapidly increases the security of ciphertext, therefore making the use of brute force and frequency analysis attack approach difficult to break the system. Thus, the obtained result increases more confusion to the cipher.

#### 5.2 Conclusion

The aim of the research which was to improve security to the playfair cipher to increase its confusion using rhotrix was achieved. It can also be concluded that from the set objectives, the following was also achieved:

- a. A stronger cryptosystem to improve the weakness of the playfair cipher has been achieved
- b. A new key generation scheme was adopted with the use of rhotrix.
- c. The ciphertext generated is highly dependent on the plaintext characters to be encrypted.
- d. At encryption time, the symbol ‘q’ was used to provide space between two words, and ‘ë’ is used as filler character for repeated separate two alphabet and odd length of strings

### **5.3 Contribution to Knowledge**

- a. Introduction of rhotrix in formation of key generations in the playfair cipher.
- b. The proposed method includes the encryption of whitespaces and punctuations along with the character set, at decryption time ‘#’ symbol was inclusive, thus making the cryptanalysis more difficult than previously developed methods.

### **5.4 Future Work**

The modified cryptosystem can be extended and also test for frequency testing of extended characters and other cryptanalysis such as: adaptive chosen-plaintext attack and chosen-ciphertext attack.

## REFERENCES

- Abomhara, M., Omar Z. and Othman, O. K. (2010). An Overview of Video Encryption Techniques. *International Journal of Computer Theory and Engineering*, 2(1), 103-110.
- Alam, A, Khalid, S. and Salam, M. (2013). A Modified Version of Playfair Cipher Using  $7 \times 4$  Matrix, *International Journal of Computer Theory and Engineering*. 5(4), 626-628.
- Ajibade, A. O. (2003), "The Concept of Rhotrix in Mathematical Enrichment," *International Journal of Mathematical Education in Science and Technology*, 34 (2), 175-179.
- Amandeep, K., Harsh, K.V. and Ravindra, K. S. (2012). 3D ( $4 \times 4 \times 4$ ) – Playfair Cipher. *International Journal of Computer Applications*, 51(2), 36-38.
- Ayushi, A. (2010). A Symmetric Key Cryptographic Algorithm. *International Journal of Computer Applications*, 1(15), 1-4.
- Atanassov, K. T. and Shannon, A. G., (1998), Matrix-Tertions and Matrix-Noitrets: Exercises in Mathematical Enrichment. *International Journal of Math Education in Science and Technology*, 29, 898-903
- Basu, S. and Ray, U. K. (2012). Modified Playfair Cipher using Rectangular Matrix *International Journal of Computer Applications*, 46(9), 28-30.
- Bhattacharyya, S., Chakraborty, S. and Chand, N. (2014). A Modified Encryption Technique using Playfair Cipher  $10 \times 9$  Matrix with Six Iteration Steps, *International Journal of Engineering Science and Innovative Technology (IJCATM)*, 3 (2), 307-312.
- Canniere, C. (2007). Analysis and Design of symmetric encryption algorithms.

- (DoctoralDissertation, Katholieke Universiteit Leuven, Belgium). Retrieved November 11, 2014 from <https://www.cosic.esat.kuleuven.be/publications/thesis-139.pdf>
- Chand, N. and Bhattacharyya, S. (2014). A Novel Approach for Encryption of Text Messages Using Playfair Cipher  $6 \times 6$  Matrix with Four Iteration Steps, *International Journal of Engineering Science and Innovative Technology (IJCATM)*, 3 (1), 478-484.
- Chris C., (2006). *The Friedman Test*. Retrieved on 15/07/2015 from [www.nku.edu/~christensen](http://www.nku.edu/~christensen)
- Chirstof, P. and Pelzi J. (2010) *Understanding Cryptography*. [Springer-Verlag, Berlin Heidelberg]. DOI: 10.1007/978-3-642-04101-3
- Choudhary, J., Gupta, R. K. and Singh, S. (2013). A Survey of Existing Playfair Ciphers. *International Journal of Engineering and Advanced Technology (IJEAT)*, 2(4), 658-659.
- Harinandan, T., Arnab, S., Akash, G. and Swashata, G. (2014). Novel Modified Playfair Cipher using a Square Matrix. *International Journal of Computer Applications*, 101(12), 16-21.
- Huang, X., Wang, C., Huang, W. and Li, J. (2013), the Nonlinear Filter Boolean Function of LILI 128 Stream Cipher Generator, 4 (2), 165-168.
- Ibukun, E., Olawande, D. and Nicholas, O. (2013). Improving Security Using Refined 16 X 16 Playfair Cipher for Enhanced Advanced Encryption Standard (AES), *Covenant Journal of Informatics and Communication Technology (CJICT)*, 1(2), 79-88.
- Iqbal, Z., Bhumika, G., Kamal, Kr. G. and Prachi, G. (2014). Enhanced the Security of Playfair Technique using Excess 3 Code (XS3) and Ceasar Cipher. *International*

*Journal of Computer Applications*, 103(13), 16-20.

Katz, J. and Lindell, Y. (2008). Introduction to modern cryptography. Chapman and Hall Taylor & Francis Group 6000 Broken Sound Parkway NW, pp. 32-50.

Kester, Q.-A. (2013). A Hybrid Cryptosystem based on Vigenere cipher and Columnar Transposition cipher. *International Journal of Advanced Technology and Engineering Research*, 3 (1), 141-147.

Jitendra, C., Ravindra, K. G., and Shailendra, S. (2013). A Generalized Version of PlayFair Cipher. *An international journal of advanced computer technology*, 2 (6), 176-179.

Ouday, N. A. H. and Baraa, W. S. (2013). 11 × 11 Playfair Cipher based on a Cascade of LFSRs. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 12(1), 29-35.

Ranjeet, M., Vivek, S., Jibi, A. and Rajni, M. (2014). Analysis and Comparison of symmetric key Cryptographic algorithms based on various File features. *International Journal of Network Security & Its Applications (IJNSA)*, 6(4), 43-52. DOI: 10.5121/ijnsa.2014.6404.

Ravindra, B. K., Uday, S. K., Vinay, B. and Aditya, P. K. (2011). An Extension to Traditional Playfair Cryptographic Method, *International Journal of Computer Applications*, 17 (5), 34-36.

Safwat H. (2014). A Novel Implementation of an Extended 8x8 Playfair Cipher Using Interweaving on DNA-encoded Data. *International Journal of Electrical and Computer Engineering (IJECE)*, 4(1), 93-100.

Saini, N., and Mandal, S. (2015). Review paper on cryptography. *International Journal of Research (IJR)*, 2(5), 45-49.

Sangapu, V. A., and Gomatam, V. A. (2014). Recent Advancements on Symmetric

Cryptography Techniques. *Global Journal of Computer Science and Technology*, 14(2), 20-30.

Sani, B. (2008) Conversion of a Rhotrix to a Coupled Matrix. *International Journal of Mathematical Education in Science and Technology* 39(2), 244-249. DOI 10.1080/00207390701500197

Srivastava, S.S., and Gupta, N. (2011). A Novel Approach to Security using Extended Playfair Cipher, *International Journal of Computer Applications*, 20 (6). DOI 10.1.1.206.4168 Retrieved September 11, 2015 from <http://research.ijcaonline.org/volume20/number6/10.1.1.206.4168.pdf>

Stallings, W. (5th Ed). (2011), *Cryptography and Network Security Principle and Practice*. Pearson Education: New York, pp. 10-22.

Stallings, W., and Brown, L. (2nd Ed). (2011) *Computer Security – Principles and Practice*. Pearson Education: New York, pp.39-62.

Stallings, W. (4th Ed). (2006), *Cryptography and Network Security-Principles and Practice*. Pearson Education:New York, pp.3

Wasnik, T.P., Vishal, S. P., Sushant, A. P., Sachin, R. D. and Gaurav J. S. (2013).

Cryptography as an instrument to network security.*International Journal of Application Innovation in Engineering & Management (IJAIEM)*, 2(3), 72-80.

## APPENDIX A

Letter Frequency of English Language Characters (Christof and Pelzi, 2010)

<b>Letter</b>	<b>Frequency</b>
<b>A</b>	0.0817
<b>B</b>	0.015
<b>C</b>	0.0278
<b>D</b>	0.0425
<b>E</b>	0.127
<b>F</b>	0.0223
<b>G</b>	0.0202
<b>H</b>	0.0609
<b>I</b>	0.0697
<b>J</b>	0.0015
<b>K</b>	0.0077
<b>L</b>	0.0403
<b>M</b>	0.0241
<b>N</b>	0.0675
<b>O</b>	0.0751
<b>P</b>	0.0193
<b>Q</b>	0.001
<b>R</b>	0.0599
<b>S</b>	0.0633
<b>T</b>	0.0906
<b>U</b>	0.0276
<b>V</b>	0.0098
<b>W</b>	0.0236
<b>X</b>	0.0015
<b>Y</b>	0.0197
<b>Z</b>	0.0007

## APPENDIX B

Frequency Distribution of Ciphertext Characters based on Conventional Playfair Cipher  
using  $11 \times 11$  matrix

<b>Characters</b>	<b>Frequency</b>	<b>Probability</b>	<b>Percentage</b>	<b>IC</b>
<b>A</b>	4	0.0434782609	4.35%	0.001433
<b>B</b>	1	0.0108695652	1.09%	0
<b>C</b>	2	0.0217391304	2.17%	0.000239
<b>D</b>	4	0.0434782609	4.35%	0.001433
<b>E</b>	4	0.0434782609	4.35%	0.001433
<b>F</b>	8	0.0869565217	8.70%	0.006689
<b>G</b>	5	0.0543478261	5.43%	0.002389
<b>H</b>	7	0.0760869565	7.61%	0.005017
<b>I</b>	3	0.0326086957	3.26%	0.000717
<b>J</b>	2	0.0217391304	2.17%	0.000239
<b>K</b>	1	0.0108695652	1.09%	0
<b>L</b>	7	0.0760869565	7.61%	0.005017
<b>M</b>	0	0	0.00%	0
<b>N</b>	2	0.0217391304	2.17%	0.000239
<b>O</b>	7	0.0760869565	7.61%	0.005017
<b>P</b>	0	0	0.00%	0
<b>Q</b>	0	0	0.00%	0
<b>R</b>	0	0	0.00%	0
<b>S</b>	6	0	0.00%	0.003583
<b>T</b>	7	0.0760869565	7.61%	0.005017
<b>U</b>	8	0.0869565217	8.70%	0.006689
<b>V</b>	11	0.1195652174	11.96%	0.013139
<b>W</b>	2	0.0217391304	2.17%	0.000239
<b>X</b>	0	0	0.00%	0
<b>Y</b>	0	0	0.00%	0
<b>Z</b>	1	0.0108695652	1.09%	0
<b>total</b>	92	1	100.00%	0.058528

## APPENDIX C

### Probability Distribution of Ciphertext Characters using Proposed System

<b>Characters</b>	<b>Frequency</b>	<b>Probability</b>	<b>Percentage</b>	<b>IC</b>
<b>A</b>	2	0.1052631579	10.53%	0.000239
<b>B</b>	0	0	0.00%	0
<b>C</b>	1	0.0526315789	5.26%	0
<b>D</b>	0	0	0.00%	0
<b>E</b>	0	0	0.00%	0
<b>F</b>	3	0.1578947368	15.79%	0.000717
<b>G</b>	2	0.1052631579	10.53%	0.000239
<b>H</b>	3	0.1578947368	15.79%	0.000717
<b>I</b>	0	0	0.00%	0
<b>J</b>	0	0	0.00%	0
<b>K</b>	0	0	0.00%	0
<b>L</b>	1	0.0526315789	5.26%	0
<b>M</b>	0	0	0.00%	0
<b>N</b>	0	0	0.00%	0
<b>O</b>	1	0.0526315789	5.26%	0
<b>P</b>	0	0	0.00%	0
<b>Q</b>	0	0	0.00%	0
<b>R</b>	0	0	0.00%	0
<b>S</b>	2	0.1052631579	10.53%	0.000239
<b>T</b>	1	0.0526315789	5.26%	0
<b>U</b>	2	0.1052631579	10.53%	0.000239
<b>V</b>	0	0	0.00%	0
<b>W</b>	1	0.0526315789	5.26%	0
<b>X</b>	0	0	0.00%	0
<b>Y</b>	0	0	0.00%	0
<b>Z</b>	0	0	0.00%	0
<b>total</b>	19	1	100.00%	0.002389