

Thesis No: CSER

## A TECHNIQUE FOR DNA CRYPTOGRAPHY BASED ON DYNAMIC MECHANISMS

By

**Md. Rafiul Biswas**



Department of Computer Science and Engineering  
Khulna University of Engineering & Technology  
Khulna 9203, Bangladesh  
September, 2018

# **A Technique for DNA Cryptography Based on Dynamic Mechanisms**

By

**Md. Rafiul Biswas**

Roll No: 1707506

A Thesis submitted in partial fulfillment of the requirements for the degree of  
Master of Science in Engineering in Computer Science and Engineering



Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

September, 2018

## Declaration

This is to certify that the thesis work entitled “**A Technique for DNA Cryptography Based on Dynamic Mechanisms**” has been carried out by Md. Rafiul Biswas in the Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh. The above thesis work or any part of this work has not been submitted anywhere for the award of any degree or diploma.

*RAZ*  
26.09.18

---

Signature of Supervisor

**Dr. Kazi Md. Rokibul Alam**

Professor,

Dept. of Computer Science and

Engineering,

Khulna University of Engineering &

Technology

*Rafiul*  
26.09.18

---

Signature of Candidate

**Md. Rafiul Biswas**

Roll: 1707506

Dept. of Computer Science and

Engineering,


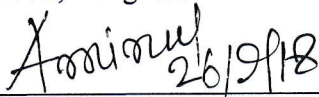
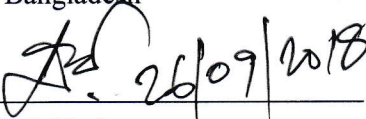
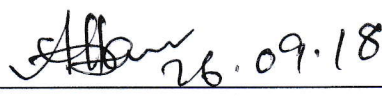
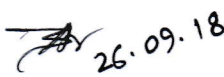
Khulna University of Engineering &

Technology

## Approval

This is to certify that the thesis work submitted by Md. Rafiul Biswas entitled “**A Technique for DNA Cryptography Based on Dynamic Mechanisms**” has been approved by the board of examiners for the partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh in September, 2018.

### BOARD OF EXAMINERS

1.   
26.09.18  
\_\_\_\_\_ Chairman  
Dr. Kazi Md. Rokibul Alam (Supervisor)  
Professor  
Dept. of Computer Science and Engineering  
Khulna University of Engineering & Technology,  
Khulna, Bangladesh
2.   
26/9/18  
\_\_\_\_\_ Member  
Dr. Muhammad Aminul Haque Akhand  
Professor & Head  
Dept. of Computer Science and Engineering  
Khulna University of Engineering & Technology,  
Khulna, Bangladesh
3.   
26/09/2018  
\_\_\_\_\_ Member  
Dr. M. M. A Hashem  
Professor  
Dept. of Computer Science and Engineering  
Khulna University of Engineering & Technology,  
Khulna, Bangladesh
4.   
26.09.18  
\_\_\_\_\_ Member  
Dr. K. M. Azharul Hasan  
Professor  
Dept. of Computer Science and Engineering  
Khulna University of Engineering & Technology  
Khulna, Bangladesh
5.   
26.09.18  
\_\_\_\_\_ Member  
Dr. Mohammad Abdur Rouf (External)  
Professor  
Dept. of Computer Science and Engineering  
Dhaka University of Engineering & Technology  
Gazipur, Bangladesh

Gazipur, Bangladesh

## **Acknowledgement**

At first, I would like to thank Almighty for showering all his blessings on me whenever I needed. It is my immense pleasure to express my indebtedness and deep sense of gratitude to my supervisor Dr. Kazi Md. Rokibul Alam, Professor, Department of Computer Science and Engineering (CSE), Khulna University of Engineering & Technology (KUET) for his continuous encouragement, constant guidance and keen supervision throughout of this study. I am especially grateful to him for giving me his valuable time whenever I need and always providing continuous support in my effort.

I am extremely grateful to Information & Communication Technology (ICT) Division of Ministry of Posts, Telecommunications and Information Technology, Bangladesh to provide me scholarship to carry on my M. thesis in the M. Sc. Engg. program.

I am especially grateful to all the faculty members of the Department of CSE, KUET to have their privilege of intensive, in-depth interaction and suggestions for the successful completion of my master degree.

At last I am grateful to my parents, family member and friends for their patience, support and encouragement during this period.

September 2018

Author

## **Abstract**

This thesis proposes a dynamic technique for DNA cryptography based on dynamic sequence table and dynamic DNA encoding mechanisms along with an asymmetric cryptosystem. The main focusing area of the thesis is to deploy a completely dynamic approach which includes generating dynamic sequence table to produce ciphertext, dividing the ciphertext into a number of chunks, applying public key to encrypt each chunk and finally dynamic DNA encoding is used to merge the chunk of each ciphertext. Firstly, the plaintext is transformed into its corresponding DNA bases. Secondly, ASCII characters are randomly assigned to each sequence of DNA bases in the dynamic sequence table. Herein, the position of the DNA bases changes dynamically through iteration process. The generated ciphertext is divided into a fixed sized of chunks and to encrypt each chunk an symmetric cryptosystem is employed. Dynamic DNA encoding is formed using adequate random strings and efficient mathematical series and it is used to set how much random strings will be taken to merge all the ciphertext of chunks together. The usage of dynamic sequence table, public key cryptography, random string and dynamic DNA encoding altogether enhance the secrecy of data. According to the National Institute of Standards and Technology (NIST), an empirical test is performed to analyze the randomness of the ciphertext. Finally the encryption and decryption time is compared between the proposed technique and other existing techniques.

## Contents

	<b>PAGE</b>
Title Page	i
Declaration	ii
Approval	iii
Acknowledgement	iv
Abstract	v
Contents	vi
List of Table	ix
List of Figures	x
Nomenclature	xi
<b>CHAPTER I</b>	<b>Introduction</b> 1
	1.1 Background 1
	1.2 Motivation 2
	1.3 Overview of the Field 3
	1.4 Problem Statement 4
	1.5 Objectives of the Thesis 4
	1.6 Contributions 5
	1.7 Organization of the Thesis 5
<b>CHAPTER II</b>	<b>Literature Review</b> 6
	2.1 Introduction 6
	2.2 DNA Cryptography using OTP 6
	2.3 DNA Steganography 7
	2.4 Hybrid DNA Cryptography 8
	2.5 DNA Cryptography with Machine Learning 9
	2.6 Summary from the Related Works 10
<b>CHAPTER III</b>	<b>Theoretical Consideration</b> 11
	3.1 Introduction 11
	3.2 DNA cryptography with symmetric cryptosystem 11
	3.3 DNA cryptography with asymmetric cryptosystem 12
	3.4 Required Cryptographic Tools 12
	3.4.1 RSA cryptosystem 12
	3.4.2 ElGamal cryptosystem 14
	3.4.3 Paillier cryptosystem 15

	<b>Proposed Dynamic DNA Cryptographic</b>	
<b>CHAPTER IV</b>	<b>Technique</b>	17
	4.1 Introduction	17
	4.2 Generation of dynamic DNA sequence table	17
	4.3 Formation of Dynamic DNA Encoding	19
	4.4 Encryption Process	20
	4.5 Decryption Process	21
<b>CHAPTER V</b>	<b>Experimental Analysis</b>	23
	5.1 Experimental Setup	23
	5.2 Experimental Results	23
	5.2.1 Output of Dynamic Sequence Table	23
	5.2.2 Output of Dynamic DNA Encoding	23
	5.2.3 Output of Encryption Process	24
	5.2.4 Output of Decryption Process	24
	5.3 Comparisons and Discussions	26
	5.3.1 Comparisons with respect to data size	26
	5.3.2 Comparisons with respect to time	29
	5.3.3 Comparison of time among asymmetric cryptosystems based proposed technique	31
	5.3.4 Comparisons with other related techniques	32
<b>CHAPTER VI</b>	<b>Security and Statistical Analysis</b>	35
	6.1 Introduction	35
	6.2 Ciphertext Strength Analysis	35
	6.3 NIST Statistical Test	37
	6.4 Random Number Generation Tests	38
	6.4.1 The Frequency Test	38
	6.4.2 Frequency Test within a Block	39
	6.4.3 Runs Test	40
	6.4.4 Test for the Longest Run of Ones in a Block	40
	6.4.5 Rank Test	41
	6.4.6 Discrete Fourier Transform (Spectral) Test	41
	6.4.7 Non-overlapping Template Matching Test	42
	6.4.8 Overlapping Template Matching Test	42
	6.4.9 Maurer's "Universal Statistical" Test	43
	6.4.10 Linear Complexity Test	43
	6.4.11 Serial Test	44



6.4.12	Approximate Entropy Test	44
6.4.13	Cumulative Sums Test	45
6.4.14	Random Excursions Test	45
6.4.15	Random Excursions Variant Test	46
6.5	Analysis the results of the proposed technique	46
6.5.1	The Frequency Test	46
6.5.2.	Frequency Test within a Block	47
6.5.3	The Runs Test	47
6.5.4	Tests for the Longest-Run-of-Ones	48
	Block	
6.5.5	Rank Test	48
6.5.6	Discrete Fourier Transform	49
6.5.7	The Non-overlapping Template Test	49
6.5.8	The Overlapping Template Test	51
6.5.9	Maurer's "Universal Statistical" Test	51
6.5.10	The Linear Complexity Test	52
6.5.11	The Serial Test	52
6.5.12	The Approximate Entropy Test	53
6.5.13	The Cumulative Sums (Cusums) Test	53
6.5.14	The Random Excursions Test	54
6.5.15	The Random Excursions Variant Test	54
6.6	Discussion from the Analysis	55
<b>CHAPTER VII</b>	<b>Conclusions</b>	57
	7.1 Concluding Discussion	57
	7.2 Future work	57
<b>REFERENCES</b>		58
<b>APPENDICES</b>		62

## LIST OF TABLES

<b>Table Number</b>	<b>Caption of Tables</b>	<b>Page Number</b>
5.1	Output of Dynamic Sequence Table	24
5.2	Output of Encryption Process	24
5.3	Output of Decryption Process	25
5.4	Dataset for RSA based proposed dynamic DNA cryptographic technique	26
5.5	Dataset for ElGamal based proposed dynamic DNA cryptographic technique	26
5.6	Dataset for Paillier based proposed dynamic DNA cryptographic technique	27
6.1	Frequency Test	46
6.2	Block Frequency Test	47
6.3	The Runs Test	47
6.4	Longest Runs Test	48
6.5	Rank Test	48
6.6	FFT	49
6.7	Non overlapping test	49
6.8	Overlapping Template	51
6.9	Universal test	51
6.10	Linear Complexity Test	52
6.11	Serial Test	52
6.12	Entropy Test	53
6.13	Cumulative Test	53
6.14	Random excursions test	54
6.15	Random excursions variant test	54
6.16	Final Analysis report	56
A.1	Dynamic Sequence Table	62

## LIST OF FIGURES

Figure Number	Caption of the Figure	Page Number
1.1	DNA Structure	4
3.1	Symmetric Cryptosystem	11
3.2	Asymmetric Cryptosystem	12
4.1	Generation of dynamic sequence table	18
4.2	Formation of Dynamic DNA Encoding	19
4.3	Encryption Process	21
4.4	Decryption Process	22
5.1	Comparison between plaintext and ciphertext length (a) RSA; (b) ElGamal; (c) Paillier.	28
5.2	Comparisons of length of ciphertext among RSA, ElGamal, Paillier	29
5.3	Comparisons between encryption time and decryption time for different data size. (a) RSA; (b) ElGamal; (c) Paillier	31
5.4	Comparisons of encryption time among RSA, ElGamal and Paillier based proposed technique	32
5.5	Comparisons of decryption time among RSA, ElGamal and Paillier based proposed technique	32
5.6	Comparisons of encryption time with other related techniques	33
5.7	Comparisons of decryption time with other related techniques	33

## Nomenclature

AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
DES	Data Encryption Standard
3DES	Tripple DES
DNA	Deoxyribo Nucleic Acid
FFT	Fourier Transform Test
NIST	National Institute of Standards and Technology
OTP	One Time Pad
PCR	Polymerase Chain Reaction
RSA	Rivest–Shamir–Adleman

# CHAPTER I

## Introduction

### 1.1 Background

With the continuous improvement of technology, information security has become a vital issue in our digital life. From the primitive age of science man always tried to hide important message from enemy. Cryptography refers to encapsulation of secret information through encrypting into ciphertext so that third party can't understand the original message. DNA cryptography is comparatively a recent topics of research which refers to hide plaintext through the use of DNA sequence. DNA cryptography concept was firstly introduced by Adleman et. al. [1] to solve the Hamiltonian path problem which is referred to the Traveling Salesman Problem.

The recent review shows that average DNA cryptographic techniques are based on the symmetric cryptosystems. Symmetric cryptosystems lack the security of key because same key is used in encryption and decryption process. Besides, transmission of key among the participants needs a secure transmission channel. Whereas the asymmetric cryptosystems overcome the limitations of symmetric cryptosystems by providing more security and expediency. In asymmetric cryptosystems the public key is exposed among the involved participants and the private key is not shared. However, DNA cryptosystems breaks the Number Theory Research Unit (NTRU) system in polynomial time [2]. Recently the Data Encryption Standard (DES) and some other traditional cryptographic techniques secrecy have been breached using DNA computing

The technique proposed in [3] is a new tile-colony algorithm that can utilize the DNA hybridization process as an effective source for the random key construction. It uses a physical process like as thermal noise for generating random number.

DNA self-assembly property [4] shows how to break the RSA public key cryptosystem. The paper proposed in [5] presents that Diffie-Hellman is possible to break by using the properties of the DNA self-assembly. Thereby, hybrid cryptographic technique which is composed of DNA mechanism, asymmetric cryptosystems and mathematical equation, is

proposed to overcome the limitations of existing technique as well as ensure better security.

The main aspect of the thesis is to deploy a complete dynamic mechanism. Plaintext is converted to binary sequence. Then dynamic sequence table is generated for all the ASCII characters. To form dynamic sequence table, firstly each DNA sequence are allocated to each ASCII character randomly. Later, the positions of DNA sequence rearrange according to a mathematical series. Number of iterations are performed to increase confusion in the ciphertext. The generated ciphertext are split into fixed sized chunks to encrypt each chunk an asymmetric cryptosystem is employed. Then dynamic DNA encoding *i.e.* composed random strings and mathematical series, is used to fix up the required number of random strings to merge the chunks altogether. Thereby, multi-stage technique enhances the level of secrecy of data which is appropriate for any application of cryptography.

## **1.2 Motivation**

Since ancient time, human being has devised schemes to conceal communications, whatever they are. Data security is very important, and often must be maintained at any cost and by any means. From the political level to that of individuals, information is tremendously valuable. Businessmen try to secure their trade, unlawful business representatives hide information from the law and they must decipher the message. Phone surveillance has created a veritable psychosis among political figures and businessmen who accused special services of such practices. We come in contact with cryptology every day for various applications like electronic (e-) voting system, e-payment system, e-tender system, computerized lotteries, computerized contests, e-auction system etc.

DNA cryptography is a promising technique to ensure the secrecy as well as to speed up the transmission of data over computer networks. It is more advantageous than traditional cryptography because it provides two fold securities whereas the traditional cryptography provides only one fold security. To provide two fold securities, DNA cryptography combines computational problems as well as biological problems [6] where the first fold is ensured by an existing cryptosystem and the second fold is provided by the features of DNA characteristics. The binding capabilities of DNA bases (A-T, C-G) offer the opportunity of creating self-assembly structures that are an excellent means of ensuring

secrecy [7]. Thus DNA bases are used as the information carrier to make the transmitted data more secure over the public communication channel.

Unlike traditional cryptography, DNA cryptography relies on DNA characteristics along with cryptographic techniques to ensure security. Valuable properties of this technique are: self-assembling criteria of DNA molecules [8]. Biotic Pseudo DNA cryptography method is based upon the genetic information on biological systems [9]. Traditional cryptosystems are only based on mathematical techniques that weaken the robustness of their encryption process [10]. For example, recent studies show that cryptosystems like RSA, DES, AES, MD5 etc. are not secure enough [11]. Because it is expected that by proposing unbreakable cryptosystem, DNA cryptography could ensure the data security for the next generation.

### **1.3 Overview of the Field**

The foremost role of cryptography is to hide the data from attackers. It has two main terms plaintext and ciphertext. The original message which is send by the user that known as plaintext and the text which adds keys to the original message is called as ciphertext. Encryption techniques is categorized as symmetric cryptosystems and asymmetric cryptosystems. In symmetric cryptosystems, common key is used for sender and receiver side. Some of the symmetric cryptosystems are AES, DES, and 3DES. In asymmetric cryptosystems, the public key of the first user (user1) used in sender side and the private key of the second user (user2) used in receiver side. A few of the asymmetric cryptosystems are RSA, Diffie-Hellman, ECC, and digital signature algorithm. With smallest number of rounds, AES encrypts the message with the key length of 128 bits, 192 bits, and 256 bits. DNA Cryptography is the means to hide the original data in terms of the DNA sequence. The characteristics of DNA paves an idealistic approach that ensures security for the data transmitted from sender to receiver. It can tackle the ever-changing data breaches and enhances security.

Deoxyribo Nucleic Acid (DNA) carries the genetic information in the form of nucleotides (molecules that form the structural units of DNA) which enhances the growth and development of every living organism. The DNA comprises of four bases namely Adenine (A), Thymine (T), Cytosine (C), and Guanine (G). These nucleotides occurs repeatedly to form a long DNA sequence as shown in Fig. 1.1



Figure 1.1: DNA Structure

In general, the DNA sequences are used to represent or encode the original data and the properties and DNA nucleotides are used a security enhancing feature which also helps to perform encryption and decryption of DNA sequence representing data.

#### 1.4 Problem Statement

Existing cryptographic technique has the following limitations.

- Traditional cryptosystems offer only single fold security.
- Their secrecy are breached while the underlying computational techniques are revealed.
- Existing DNA cryptographic techniques extensively deploy symmetric cryptosystems.
- Often they treat DNA bases statically.

#### 1.5 Objectives of the Thesis

To cope with the situation, this thesis proposes a new DNA cryptographic technique through exploiting dynamic mechanisms *i.e.* dynamic sequence table [12] and dynamic DNA encoding [13] along with an asymmetric cryptosystem. Dynamic sequence table is generated through following steps: initially DNA sequence are assigned to each ASCII character randomly and the positions of DNA sequence changes through iteration number. The ciphertext produced from the dynamic sequence table are split into a number of fixed sized chunks and to encrypt each chunk public key cryptography is employed. To merge the chunk altogether dynamic DNA encoding is used. Dynamic DNA encoding is formed



as follows: sufficient random strings are generated and clever mathematical series determine the required number of random strings. Therefore, multi-stage security is ensured here to increase the level of secrecy. To reach the goal following issues will be considered.

- To ensure more security over traditional asymmetric cryptosystems.
- To adopt dynamic mechanisms for DNA bases to enrich secrecy
- To exploit a modern cryptosystem to maintain robustness
- To propose a generous technique to apply on any form and platform of data

## 1.6 Contributions

To ensure dynamism, the proposed technique newly develops two mechanisms

- Dynamic sequence table
- Dynamic DNA encoding

## 1.7 Organization of the Thesis

To ensure secrecy of data the dynamic mechanisms is proposed. The thesis is organized as below.

- **Chapter I** briefly explains the introduction of the thesis and motivation of works. Overview of the field and the objectives of thesis are also discussed elaborately here.
- **Chapter II** represents the existing works in the related field and focuses on the advantages and drawbacks of existing works.
- **Chapter III** discusses the theoretical background of the thesis.
- **Chapter IV** explains the dynamic DNA cryptographic technique. The procedure of ensuring multistage security of data through dynamic mechanisms is described here elaborately. Every step of encryption and decryption also discussed in this chapter.
- **Chapter V** explains the experimental analysis. This chapter also explains the experimental setup and the analyzing results that compares with existing techniques performance. Security, statistical analysis, results are also discussed this chapter.
- **Chapter VI** provides an analysis report on security of the proposed technique.
- **Chapter VII** concludes this thesis together with the outline of probable future directions of research opened by this work.

## CHAPTER II

### Literature Review

#### 2.1 Introduction

Due to advancement in the data communication technology, it emphasizes the need to ensure more secrecy of data. Threats on data to break security has become a matter of concern nowadays. A number of cryptographic techniques have been proposed. DNA cryptography is comparatively a new concept in the field of security. Analyzing the existing works, the related works can be majorly categorized in the following types.

- DNA cryptography with one time pad (OTP)
- DNA steganography
- Hybrid DNA cryptography
- DNA cryptography with machine learning

#### 2.2 DNA Cryptography using OTP

One time pad (OTP) is a principle of key generation applied to the stream ciphering method which offers total privacy. The OTP encryption scheme has proved to be unbreakable in theory, but difficult to realize in practical applications. Because OTP encryption specially requires the absolute randomness of the key, its development has suffered from dense constraints. DNA cryptography is a new and promising technology in the field of information security. DNA chromosomes storing capabilities can be used as one-time pad structures with pseudo-random number generation and indexing in order to encrypt the plaintext messages.

The technique proposed in [14] presents a feasible solution to the OTP symmetric key generation and transmission problem with DNA at the molecular level. Through recombinant DNA technology, by using only sender-receiver known restriction enzymes to combine the secure key represented by DNA sequence and the T vector and generate the DNA bio-hiding secure key and then place the recombinant plasmid in implanted bacteria for secure key transmission. The designed bio experiments and simulation results show that the security of

the transmission of the key is further improved and the environmental requirements of key transmission are reduced

In [15] the author proposed pseudo DNA cryptography technique provides three level of security because they used biological process (DNA complementary rule), arithmetic operation (XOR operation), and OTP scheme (random key is shared between sender and recipient) in this technique.

The technique proposed in [16] is based on DNA secret writing. Plaintext is converted to binary value to perform one time pad operation (OTP) where the length for each bit  $OTP > 10 \times n$ . Ciphertext is stored in microdot. On the other hand, Polymerase chain reaction (PCR) technology is used in decryption process.

A two stage encryption algorithm based on DNA sequence has proposed in [17]. In the first stage an encryption of plain text is done by generating a random key. The plain text is again encrypted to produce the cipher text in the second stage. Moreover, this encryption algorithm is based on a symmetric key cryptography system, where we provide a shared key to encrypt as well as decrypt the intended message.

The technique proposed in [18] is based on DNA OTP encryption to encrypt a huge number of short messages. The disadvantages of this technique are, difficulties to prepare a huge number of DNA OTP in which data can be easily separated and read out.

### **2.3 DNA Steganography**

The technique proposed in [19] considers DNA based steganographic technique. DNA based play fair algorithm and substitution technique are applied to encrypt the plaintext. Play fair matrix is generated using secret key and the chains of amino-acid is formed through binary coding. In addition, it added two bit binary with amino-acid to increase confusion.

The technique proposed in [20] is based on DNA steganography. DNA based keys are distributed through secure medium. Herein, the ciphertext is not transmitted openly but the code blocks and the chemical reactions of DNA must be transferred securely.

The paper proposed in [21] presents an implementation of steganography using DNA molecules. They used DNA sequence using a randomly generated single-substitution key. To retrieve the message, the intended recipient must know the sequences of two primers that anneal to target regions present on the message strand. Polymerase chain reaction (PCR) and sequencing are used to retrieve the encoded sequence, which is decoded into the original plaintext via the single substitution key.

In this paper [22] the modification of the DNA insertion algorithm is used because of its low cracking probability. The confidential information like secret messages and document images are hidden inside the DNA sequence and the performance is measured by calculating the cracking probability, BPN, payload and capacity.

In this paper [23] image and DNA are the two covers, which are used to secure the data. The DNA insertion algorithm is used to hide the data in the DNA sequence and it results a fake DNA sequence. The capacity, payload, BPN and Cracking Probability are calculated for the fake DNA sequence to ensure the security. The fake DNA is hidden in a cover image using LSB and F5 algorithm.

This paper [24] presents a novel DNA based image steganography algorithm. DNA based security systems have drawn the attention of many researchers for more than a decade and lots of work have been done. Still it is in its evolutionary phase. The proposed work contributes to such a motive to improve the overall security of age old image steganography technique. It provides a two layer security to the secret data by wrapping it inside two layers of cover media, one being a theoretical DNA strand, and the other is a cover image.

The technique proposed in [25] is based on three levels where in the first level a shift key is required to generate data string S1, in second level, S1 is converted into DNA sequence and in third level, DNA sequence is converted into cipher text using look-up table. The limitations of the technique is that the input plaintext is applicable only for ASCII value.

## **2.4 Hybrid DNA Cryptography**

The technique proposed in [26] discusses the development of DNA computing various field. They focus on the vuranblity of traditional cryptosystems like DES, RSA and NTU. To

ensure better security they provide a review merging the traditional cryptosystems with DNA technique.

The technique proposed in [27] is based on cryptography on DNA storage. At first plaintext is encrypted through random private key. Then the ciphertext are swapped parallel to form small clusters.

The technique proposed in [28] considers hybrid cryptography according to RGB colors. Herein, the plaintext is represented as matrix form. Strong key is generated to encrypt and the key is encapsulated using DNA steganography. DNA bases and amino acids are applied to enhance the level of security of ciphertext.

The technique proposed in [29] offers multi-level security using round key selection. Firstly round key is selected to encrypt the message. Secondly, the ciphertext is transferred to  $16 \times 16$  matrix format. At last, shift operation is performed on ciphertext to make it more secure.

## **2.5 DNA Cryptography with Machine Learning**

The technique proposed in [30] is based on DNA computing on a lower level. Here, key is generated using the theory of natural selection namely Genetic Algorithm with Needleman-Wunsch (NW) algorithm. Afterwards the encryption and decryption process are implemented using different biological operations like transcription, translation, DNA sequencing and deep learning are performed.

The technique proposed in [31] considers hybridized model combination of DNA sequence and Genetic Algorithm (GA). Here, GA is used to minimize the correlation of adjacent pixels of image. Then XOR operation are performed on pixel value to produce DNA sub strings which is used as key to encrypt image.

The technique proposed in [32] chaotic sequence of desired length is generated by using the logistic map function whose initial value is calculated using the secret key. A number of DNA masks are generated and these masks along with the chaotic sequences are used to encrypt the digital image. Finally genetic algorithm is employed to get the best mask for encryption. The proposed method can resist various types of attacks and produce high entropy and very low correlation between pixels.

The study [33] has proposed a symmetric key encryption scheme effectively utilizing the randomness offered by sequence of nitrogenous bases which forms the DNA. The proposed scheme has a feistel structure with multiple rounds. The optimization power offered by genetic algorithm is exploited to select the strong and best fit keys for each round of the encryption algorithm.

## **2.6 Summary from the Related Works**

Most of the DNA cryptographic techniques consider the encryption process as static. There are a few techniques where the ciphertext is represented in matrix form and shift operation is performed to increase security. But in these cases, symmetric cryptosystem is used which provide less security than asymmetric cryptosystem. Also the key size is increased and ciphertext size is larger.

Hence, this thesis proposes a technique which provides multi-level security and in each level it considers dynamicity. Firstly, the plaintext is transferred according to dynamic sequence table. Secondly, it is divided into a number of fixed sized chunk and each chunk is encrypted by any asymmetric cryptosystems like RSA or ElGamal or Paillier. Finally to merge each chunk dynamic DNA encoding is used. NIST tests to check the randomness of key and ciphertext are performed. The technique can applied any application of information security.

## CHAPTER III

### Theoretical Consideration

#### 3.1 Introduction

The theoretical consideration of DNA cryptography to implement a technique is based on symmetric and asymmetric cryptosystems. They are described below.

#### 3.2 DNA cryptography with symmetric cryptosystem

Symmetric cryptosystem uses the same key to encrypt and decrypt data. It is also called private key cryptography. Symmetric key encryption algorithms process plaintext with the secret key to create encrypted data called ciphertext. These cryptosystems are extremely fast and well suited for encrypting large quantities of data. This is the most traditional type of cryptography, in which the key information is common for both sender and receiver. They are vulnerable when transmitting the key, some examples are: DES, RC2, 3DES, Password Based Encryption (PBE) algorithms are derived from symmetric algorithms; such algorithms use a random bytes and a number of iterations to generate a key. The security of these kind of algorithm depends on three main vectors called cryptography algorithm, key size and way to share the key. These cryptosystems are combined with DNA one time pad and DNA XOR one time pad to derive a new cryptosystem with the combination of symmetric cryptosystem and DNA. Fig 3.1 depicts symmetric cryptosystem.

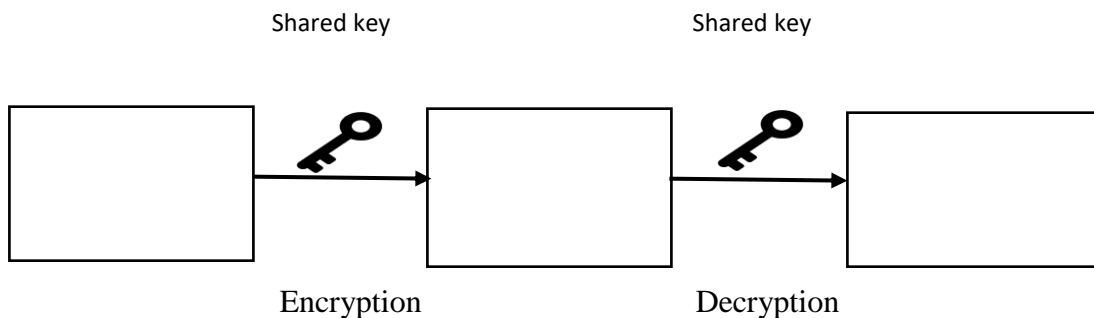


Figure 3.1: Symmetric cryptosystem

### 3.3 DNA cryptography with asymmetric cryptosystem

This cryptography approach is called public key cryptography. According to his approach, encryption and decryption is performed by two different keys. As the encryption process begins, instead of generating single key, two keys are generated called public key and private key. The generator A keeps the private key with himself and distribute the public key to all users that can send information to it. Now, as some user B want to send information to the user A. In such case, user B will use the public key of User A to perform the encoding process. Here the cryptography will be performed using public key of receiver. Now after the encoded process, the cipher information is transferred to the receiver A. As receiver receive this information, the decoding process is performed using private key of User A. This decoding process is able to get the information back in its original form. Complete cryptography process is shown in Fig. 3.2

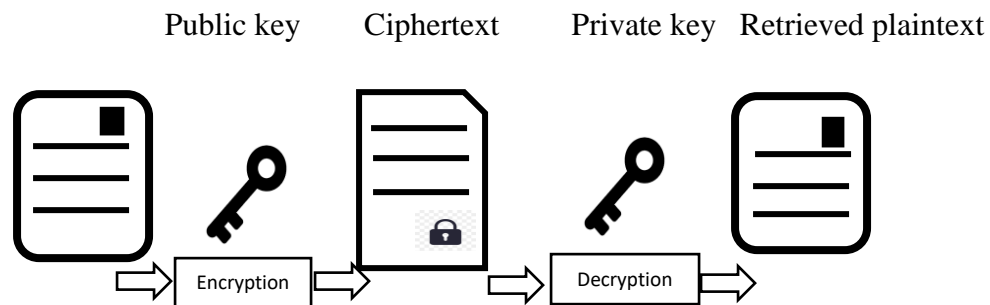


Figure 3.2: Asymmetric cryptosystem

### 3.4 Required Cryptographic Tools

The proposed technique exploits asymmetric cryptosystems with DNA cryptographic technique. Most popular asymmetric cryptosystems are RSA, ElGamal and Paillier. The proposed technique used DNA cryptography with RSA, ElGamal and Paillier cryptosystems.

#### 3.4.1 RSA cryptosystem

As opposed to traditional, symmetric encryption systems, RSA works with two different keys: a public and a private key. Both work complementary to each other, which means that a message encrypted with one of them can only be decrypted by its counterpart. Since the



private key cannot be calculated from the public key, the latter is generally available to the public.

Those properties enable asymmetric cryptosystems to be used in a wide array of functions, such as digital signatures. In the process of signing a document, a fingerprint encrypted with RSA, is attached to the file, and enables the receiver to verify both the sender and the integrity of the document. The security of RSA itself is mainly based on the mathematical problem of integer factorization. A message that is about to be encrypted is treated as one large number. When encrypting the message, it is raised to the power of the key, and divided with remainder by a fixed product of two primes. By repeating the process with the other key, the plaintext can be retrieved again. The best currently known method to break the encryption requires factorizing the product used in the division. Currently, it is not possible to calculate these factors for numbers greater than 768 bits. So that modern cryptosystems use a minimum key length of 3072 bits.

This algorithm offers the public key (n, e) for encryption and the private key (n, d) for decryption (d is secret).

#### **RSA Key Generation algorithm:**

Step 1: Generate two large prime numbers, p and q

Step 2:  $n = p * q$

Step 3:  $\phi = (p-1) * (q-1)$

Step 4: Choose a number e, coprime to phi

Step 5: Find d, such that  $d * e \text{ mod } \phi = 1$

Step 6: Publish e and n as the public key. Keep d and n as the secret key.

After key generation we can easily encrypt the plaintext to ciphertext and decrypt ciphertext to plaintext using following equation.

#### **Encryption:**

Ciphertext,  $C = P^e \text{ Mod } n$

**Decryption:**

Original Message,  $P = C^d \text{ Mod } n$

**3.4.2 ElGamal cryptosystem**

The ElGamal cryptosystem is usually used in a hybrid cryptosystem. I.e., the message itself is encrypted using a symmetric cryptosystem and ElGamal is then used to encrypt the key used for the symmetric cryptosystem. Because asymmetric cryptosystems like ElGamal are usually slower than symmetric ones for the same level of security, so it is faster to encrypt the symmetric key (which most of the time is quite small if compared to the size of the message) with ElGamal and the message (which can be arbitrarily large) with a symmetric cipher.

The ElGamal Algorithm provides an alternative to the RSA for public key encryption. 1) Security of the RSA depends on the (presumed) difficulty of factoring large integers. 2) Security of the ElGamal algorithm depends on the (presumed) difficulty of computing discrete logs in a large prime modulus. ElGamal has the disadvantage that the ciphertext is twice as long as the plaintext. It has the advantage the same plaintext gives a different ciphertext (with near certainty) each time it is encrypted.

ElGamal encryption/decryption algorithm is based on the difficulty of discrete logarithm problem where it is strait forward to raise numbers to large powers but it is much harder to do the inverse computation of the discrete logarithm. The ElGamal algorithm depends on certain parameters which are affecting the performance, speed and security of the algorithm. Here, the importance of these parameters and the role it takes in the security and complexity of the system are analyzed, particularly the effect of changing the length of the modulo number and the private key number are investigated.

**ElGamal Key Generation algorithm:**

Step 1: Generate a prime number  $g$ .

Step 2: Generate another prime number  $n$  such that  $n < g$

Step 3: Choose a random number  $x$  where  $x < n$

Step 4: Calculate  $h = g^x \text{ Mod } n$

Step 5: Publish  $n$ ,  $g$  and  $h$  as the public key. Keep  $x$  as the secret key.

### **Encryption:**

Step 1: Choose a random number  $y$ , where  $y < n$

Step 2: Calculate,  $s = h^y \text{ Mod } n$

Step 3: Calculate,  $c1 = g^y \text{ Mod } n$  and  $c2 = M * s \text{ Mod } n$

### **Decryption:**

Step 1: Calculate,  $s = c1^x \text{ Mod } n$

Step 2: Calculate  $M = c2 * s^{-1} \text{ Mod } n$

### **3.4.3 Paillier cryptosystem**

The Paillier cryptosystem is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing  $n^{\text{th}}$  residue classes is believed to be computationally difficult. Public-key cryptography becomes an important research topic in recent years. The number of convincingly secure asymmetric cryptosystems is rather small. RSA and ElGamal are representatives of two different types of asymmetric cryptosystem classes.

The scheme is an additive homomorphic cryptosystem, this means that, given only the public-key and the encryption of  $m1$  and  $m2$ , one can compute the encryption of  $m1 + m2$ . Homomorphic encryption methods provide a way to out-source computations to the cloud while protecting the confidentiality of the data. In order to deal with the large and growing data sets that are being processed nowadays, good encryption performance is an important step for practicality of homomorphic encryption methods.

### **Paillier Key Generation:**

Step 1: Choose two prime numbers  $p$ ,  $q$

Step 2: Compute  $n = p * q$  and  $\lambda = \text{lcm}(p-1, q-1)$

Step 3: Calculate  $g = n + 1$

Step 4: Calculate  $\mu = L(g^\lambda \text{ Mod } n^2)$  where  $L(u) = \frac{u-1}{n}$

Step 5: The public (encryption) key is  $(n, g)$ . The private (decryption) key is  $(\lambda, \mu)$ .

**Encryption:**

Step 1: Choose a random number  $r$ , where  $r < n$

Step 2: Calculate,  $c = g^m . r^n \text{ Mod } n^2$

**Decryption:**

Step 1: Calculate message,  $m = m = L(c^\lambda \text{ Mod } n^2) * \mu \text{ mod } n$

## CHAPTER IV

### Proposed Dynamic DNA Cryptographic Technique

#### 4.1 Introduction

This paper combines the concepts of dynamic DNA mechanisms along with an asymmetric cryptosystem. The key idea consists of four stages. They are described below.

- Generation of dynamic sequence table
- Formation of dynamic DNA encoding
- Encryption process
- Decryption process

#### 4.2 Generation of dynamic DNA sequence table

Dynamic sequence table assigns DNA base sequences for 256 ASCII characters dynamically. To generate the dynamic sequence table, at first 256 ASCII characters are mapped randomly with 256 distinct DNA base sequences alike to [24]. Where each DNA base sequence consists of four DNA bases (*i.e.* A, T, G, and C) which are assigned as ‘A’ = 00, ‘T’ = 01, ‘G’ = 10 and ‘C’ = 11. Prior to data transmission, no one except the sender and the receiver only share the initial status of this table *i.e.* Table A.1 in Appendix A. Now the sender finitely iterates the positions of DNA base sequences according to a mathematical series *e.g.* the Fibonacci series. In addition, only the involved parties know the series in advance. Here, only the positions of DNA base sequences are re-arranged at every iteration. But, positions of ASCII characters remain unchanged. The sender executes the process as follows. Fig. 4.1 depicts its generation process.

*Step 1:* Generates the Fibonacci series as  $F_{(i)} = F_{(i-1)} + F_{(i-2)}$  where  $F_{(0)} = 1$ , and  $F_{(1)} = 1$ .

*Step 2:* Picks an arbitrary value from the series. Use this value for the first iteration.

*Step 3:* Scans the first character of the plaintext; pick the corresponding ASCII value of it.

*Step 4:* Checks the value of Step 3; it is either odd or even. If odd, sets the variable ‘Bool’ as Bool = 1. Otherwise sets Bool = 0.

*Step 5:* If Bool = 1, re-arranges all DNA base sequences forwardly according to (Item Number + the value of Fibonacci series) % 256. If Bool = 0, re-arranges all DNA base sequences backwardly according to the value of a variable ‘Result’. Where, Result = (Item Number – the value of Fibonacci series) % 256. If Result < = 0, sets Result = Result + 256.

*Step 6:* Uses the next consecutive values (either in ascending or descending order) of the Fibonacci series for the next iterations.

*Step 7:* Encrypts ‘the initial arbitrary value chosen from the series’, ‘the required iteration numbers’, and ‘the value of Bool’ and sends them to the receiver after the encryption process (will be discussed in section III (c)).

Prior to decryption of the ciphertext, the receiver needs to decrypt the parameters of Step 7 to iterate the table equal to the sender.

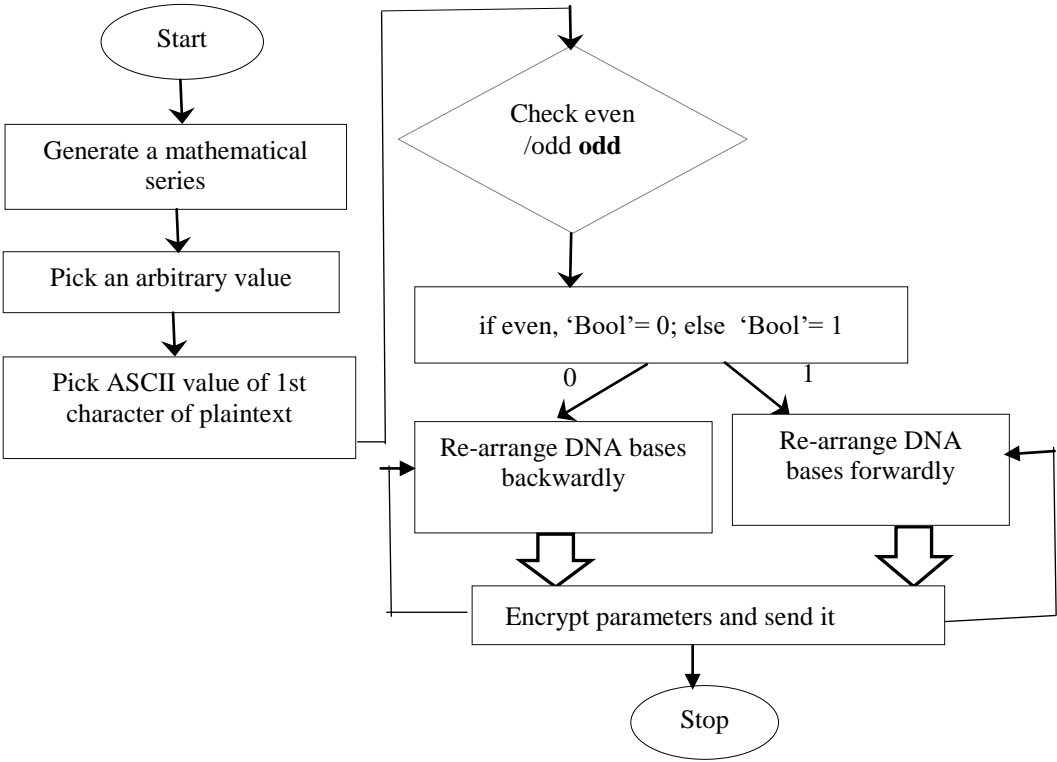


Figure 4.1: Generation of dynamic sequence table

### 4.3 Formation of Dynamic DNA Encoding

The objective of dynamic DNA encoding is to increase the level of secrecy of ciphertext. For this purpose, the ciphertext of each chunk is merged using it. The value of DNA encoding which is used to merge any two ciphertext of chunks changes dynamically due to the use of Fibonacci series and random strings. This formation procedure is as follows. Fig. 4.2 depicts its block diagram.

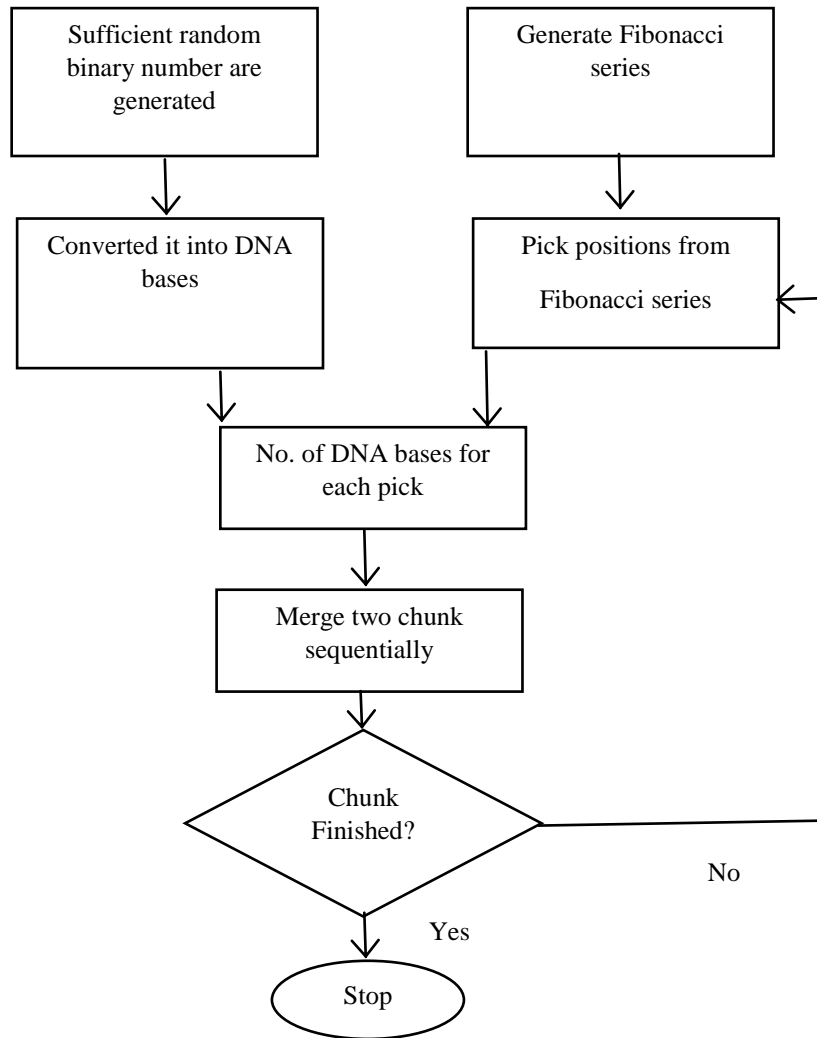


Figure 4.2: Formation of Dynamic DNA Encoding.

Step 1: At first sufficient number of random binary strings are generated.

Step 2: Random binary strings are converted into DNA base as 00 = A, 01 = C, 10 = G and 11 = T.

Step 3: Now the ciphertext of each chunk is merged using Fibonacci series where the series is: 1, 1, 2, 3, 5, 8, 13....., From this series for example, values from the 4th positions are used to pick DNA bases. Thus for the value 3, first three DNA bases are used to merge the ciphertext of chunk 1 and the ciphertext of chunk 2.

Step 4: Here if Fibonacci series is used in ascending order, while the number of chunk increases the required number of DNA bases also increases.

Instead of 4th position while another position is selected from the Fibonacci series, DNA bases used for merging the ciphertext of chunks are also changed. Thus for different position of Fibonacci series, dynamic DNA encoding produces different DNA bases.

#### **4.4 Encryption Process**

Initially both the sender and the receiver possess one copy of dynamic sequence table and the receiver generate private key for decryption and distribute for the sender public key to encrypt the message. Fig 4.3 shows its encryption process.

*Step 1:* At first convert each plaintext into 3 digit ASCII value.

*Step 2:* Transform the ASCII value into its corresponding binary value.

*Step 3:* Convert the binary value into DNA bases as 00=A, T=01, G=10, C=11.

*Step 4:* Place the DNA bases in the dynamic sequence table. The DNA bases change their positions through a number of iterations as well as the positions of each ASCII character of also changes. After each encryption process the dynamic sequence is table is wiped out.

*Step 5:* Now for each DNA base sequence each ASCII character is obtained from the dynamic sequence table. Here, the ASCII character is obtained in integer form.

*Step 6:* Divide this integer valued ciphertext into a number of fixed sized chunks.

*Step 7:* Now encrypt each chunk using the public key of the asymmetric cryptosystem.

*Step 8:* Transform each chunk into DNA bases by using step 2 and step 3.



*Step 9:* Merge each chunk of DNA bases using dynamic DNA encoding. This is the final ciphertext.

*Step 10:* No. of iteration, dimension of fixed sized chunk and the clue of dynamic DNA encoding are sent to the receiver using encrypted medium.

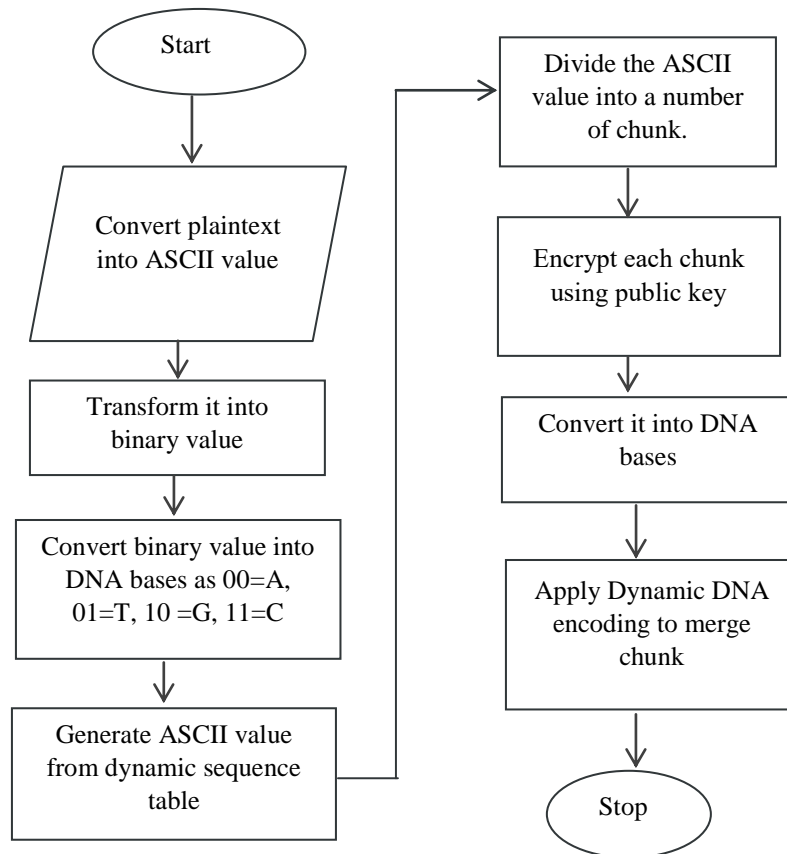


Figure 4.3: Flowchart of Encryption Process

#### 4.5 Decryption Process

Although it is the reverse process of the encryption process, the receiver executes it as follows. Fig. 4.4 depicts the decryption process.

*Step 1:* The receiver regenerate the dynamic sequence table according to the iteration number.

*Step 2:* Remove the dynamic DNA encoding portion from the ciphertext. Thus original chunk of ciphertext is found.

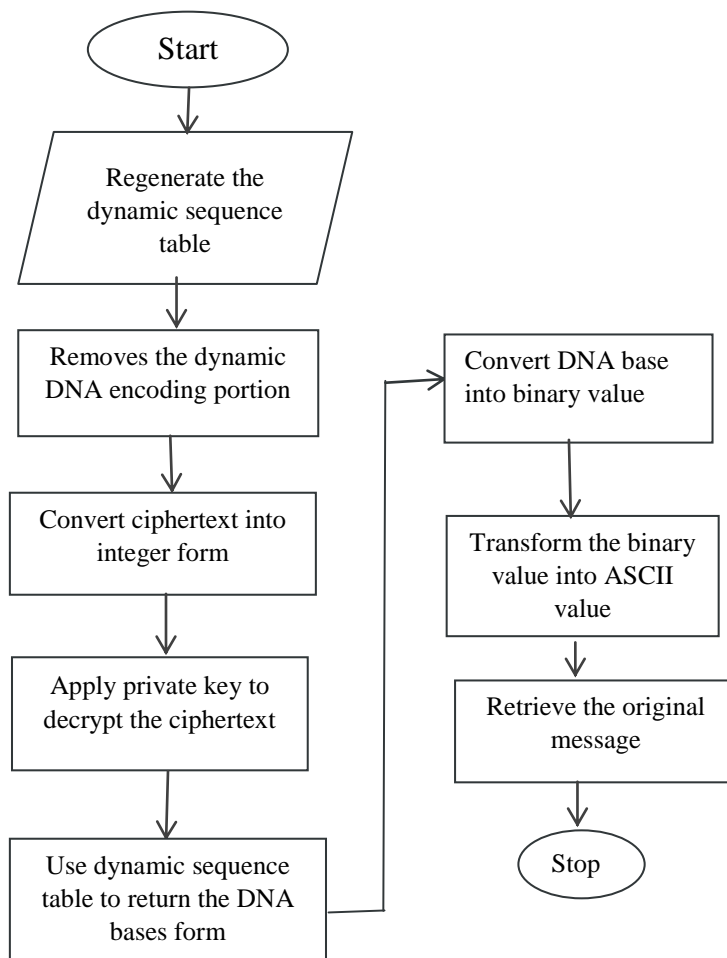


Figure 4.4: Flowchart of Decryption Process

*Step 3:* The chunk of each ciphertext is converted into integer form. Here firstly it is converted into binary form as A=00, T=01, G=10, C=11 then to integer form.

*Step 4:* Apply private key on each chunk to decrypt the ciphertext.

*Step 5:* Merge all chunk together.

*Step 6:* Use dynamic sequence table to retrieve the DNA bases form.

*Step 7:* Convert the DNA bases into binary form as A=00, T=01, C=11, G=10.

*Step 8:* Convert the binary value into ASCII value.

*Step 9:* Retrieve the original message.

## CHAPTER V

### Experimental Analysis

#### 5.1 Experimental Setup

The prototype of the proposed technique has been developed under the environment of Intel(R) Core™ i5-2430M 2.50 GHz 64 bit processor with 8 GBytes of RAM running on Windows 8.1 operating system. It has been developed on Eclipse IDE and java used as the primary language to implement the storage structure. Also in every encryption and decryption operations of exploited asymmetric cryptosystems, 2048 bit key is used. Moreover, for experiment, three asymmetric cryptosystems i.e. RSA, ElGamal and Paillier have been chosen. In addition to test the randomness of ciphertext NIST test analysis has been performed.

#### 5.2 Experimental Results

Experimental results consists of four stages. They are described below.

- Output of dynamic sequence table
- Output of Dynamic DNA encoding
- Output of encryption process
- Output of decryption process.

##### 5.2.1 Output of Dynamic Sequence Table

For the stage of dynamic sequence table of section 4.2, the output is presented in Table 5.1.

##### 5.2.2 Output of Dynamic DNA Encoding

*Step 1:* Generate sufficient random binary strings as: 01001110100101110111101

*Step 2:* Convert it into DNA bases as: CACTGCATGATTC

*Step 3:* Generate Fibonacci series: 1, 1, 2, 3, 5, 8, 13..... Use it from the 4th position.

*Step 4:* To merge the ciphertext of chunk 1 and chunk 2, use the value of 4th position of Fibonacci series i.e. three (3). Use it to pick first three DNA bases i.e. CAC. Similarly to

merge chunk 2 and chunk 3, select the value of 5th position of Fibonacci series i.e. five (5). Use it to pick next five DNA bases i.e. TGCAT. Thus dynamic DNA encoding consecutively merges the ciphertext of chunks.

**Table 5.1.** Output of Dynamic Sequence Table

Step	Operation		
Step 1	Fibonacci series: 1, 1, 2, 3, 5, 8, 13, 21, 34. . . . .		
Step 2	Pick the value 2 from the series for the 1st iteration		
Step 3	1st character of the plaintext is W ( ASCII value = 87)		
Step 4	Set 'Bool'=1 as 87 is an odd value		
Step 5	After 1st iteration, Table 1 becomes as: (A portion is shown)		
	Item Number	DNA bases	ASCII Character
	1	GGGG	x
	2	CACA	Y
	-----	----	----
256	AGGC	M	
Step 6	Next value of the series is 3 and it is used for 2nd iteration		
Step 7	Encrypt values: '2', '1', '1' for the receiver		

### 5.2.3 Output of Encryption Process

For the stage of encryption process of section 4.4, the output is presented in Table 5.2.

### 5.2.4 Output of Decryption Process

For the stage of decryption process of section 4.5, the output is presented in Table 5.3.

**Table 5.2.** Output of Encryption Process

Step	Operation	
Step 1	Plaintext	Welcome to CSE, KUET
	ASCII value	087101108099111109101032116111032067083069044032075085069084
Step 2	Binary value*	000010000111100000011111111111101111110111110010010010001010010
Step 3	DNA Base*	AAGATTAAGTCCCCTACTCGCC
Step 4	Uses dynamic sequence table <i>i.e.</i> Step 5 of Table II	
	ASCII Character*	dHG%FsD@bj&VM

Step	Operation	
Step 5	Integer value*	100072071037070115068064098106
Step 6	# of chunk = 3, # of data in every chunk = 20	
	Chunk 1* = 0000010007; Chunk 2* = 5068064098; -----	
Step 7	Cipher (chunk 1*)	8512165540962839461615101501422
	Cipher (chunk 2*)	0174264412207390691103777222850
Step 8	DNA bases (chunk 1*)	CTTATTGCCTTACGATATTGCGACCGACCCTTAAAC GACCGAAG
	DNA bases (chunk 2*)	TATAAAGGTTATCGACCGATAAAAAAAAAACCTTATTGA TTGAAAG
Step 9	Using dynamic DNA encoding merges ciphertext of chunks.	
	Final Ciphertext*	CTTATTGATTGATATAAAGGTTCCGAGAGAACCTTA C
Step 10	Encrypt values: '20', and others, already shown in Step 7 of Table II and Step 3 of Table III.	

\* Only a portion of data is shown

**Table 5.3.** Output of Decryption Process

Step	Operation	
Step 1	Retrieves values of parameters <i>i.e.</i> '20', '2', '1', '1', '5'.	
Step 2	Removes DNA bases of dynamic DNA encoding	
	Cipher (chunk 1*)	CTTATTGCCTTACGATATTGCGACG
	Cipher (chunk 2*)	TATAAAGGTTATCGACCGATAAGT
Step 3	Binary (chunk 1*)	110101000101101101010011101010010
	Binary (chunk 2*)	010001000000101001010001111000111
	Integer (chunk 1*)	851216554096283946161510150142274
	Integer (chunk 2*)	017426441220739069110377722285096
Step 4	Decrypted (chunk 1*)	00000100078974357435835
	Decrypted (chunk 2*)	506806409834894395789347
Step 5	Merged chunk*	100078974357435835506806409834894
Step 6	ASCII form*	dHG%FsD@bj&VM
Step 7	Uses dynamic sequence table <i>i.e.</i> Step 5 of Table II	
	DNA base*	GACCTGGAGTGTCTACTCGCCTC
Step 8	Binary value	11011110000001000001111111111101
Step 9	3-digit ASCII value*	087101108099111109101032116111032067
Step 10	Retrieved Plaintext	Welcome to CSE, KUET

### 5.3 Comparisons and Discussions

The prototype of the proposed technique has been implemented using RSA, ElGamal and Paillier cryptosystems. Several tests have been performed comparing with other related DNA cryptographic techniques. A comparative results based on data size and required time have been shown in Fig. 5.1, Fig. 5.2 and Fig. 5.3.

#### 5.3.1 Comparisons with respect to data size

The prototype of the technique has been tested for different data size employing RSA, ElGamal and Paillier cryptosystems respectively shown in Table 5.4, Table 5.5 and Table 5.6. From the figure for all the data size it has been observed that for RSA the size of the ciphertext is almost 12 times greater than the size of the plaintext. Whereas for ElGamal the size of the ciphertext is almost 24 times greater than the size of the plaintext. Besides for Paillier, (on average) the size of the ciphertext is almost 16 times greater than the size of the plaintext.

Table 5.4: Dataset for RSA based proposed dynamic DNA cryptographic technique

Number of test	Plaintext length(No.of Character)	Ciphertext length(No.of Character)	Encryption Time(ms)	Decryption Time(ms)
Test 1	18954	210389	1428	3541
Test 2	27632	313346	2479	5776
Test 3	34219	403913	3551	7354

Table 5.5: Dataset for ElGamal based proposed dynamic DNA cryptographic technique

Number of test	Plaintext length(No.of Character)	Ciphertext length(No.of Character)	Encryption Time(ms)	Decryption Time(ms)
Test 1	18954	451863	4569	2352
Test 2	27632	655431	9734	4576
Test 3	34219	796960	13021	6901

Table 5.6: Dataset for Paillier based proposed dynamic DNA cryptographic technique

Number of test	Plaintext length(No.of Character)	Ciphertext length(No.of Character)	Encryption Time(ms)	Decryption Time(ms)
Test 1	18954	246402	2697	1992
Test 2	27632	497376	3763	2781
Test 3	34219	670161	4201	3125

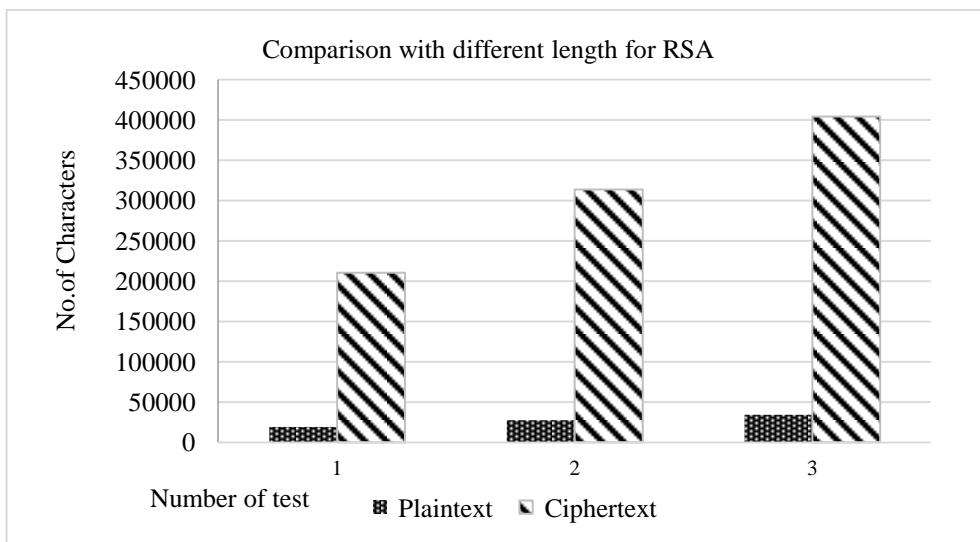


Figure 5.1:  
(a)

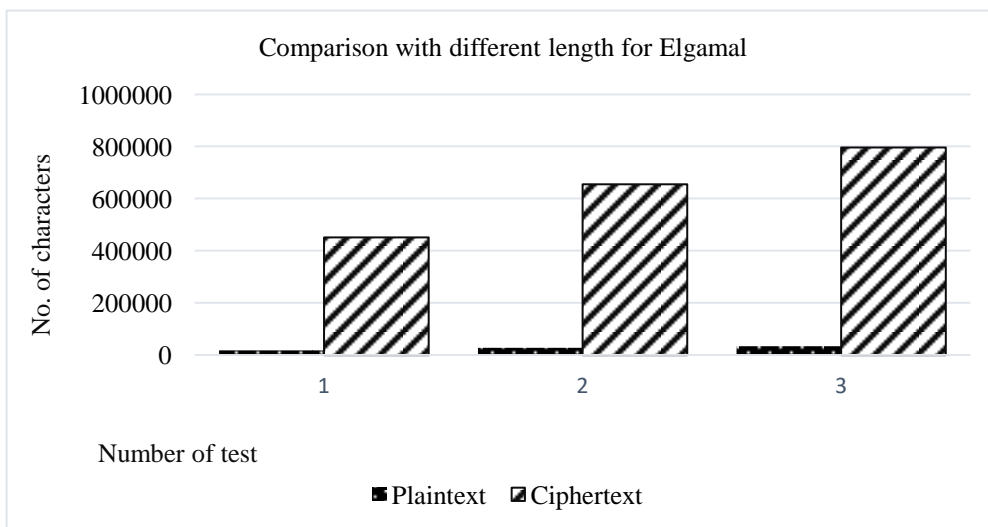


Figure 5.1:

(b)

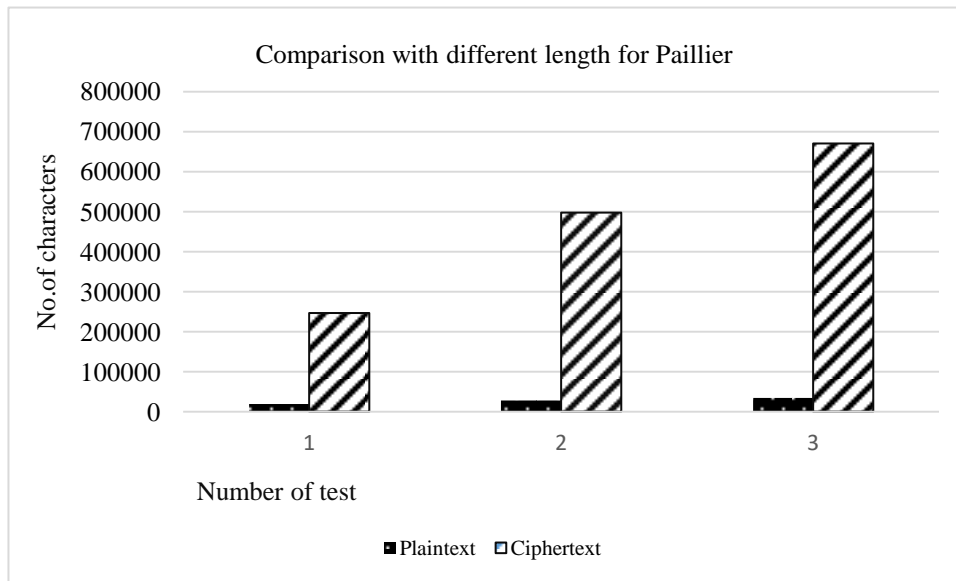


Figure 5.1: (c)

Figure 5.1 Comparison between length of plaintext and length of ciphertext (a) RSA; (b) ElGamal; (c) Paillier.

In the above Fig. 5.1, it has been observed that the length of the ciphertext for RSA is almost eleven (11) times greater than the plaintext size. Here 2048 bit key is used while employing RSA.

While for ElGamal, it has been observed that the size of ciphertext is about twenty (23) times greater than the size of plaintext. Because ElGamal exploits two modular exponent operation while RSA needs one modular operation. As Paillier is probabilistic, the ciphertext size varies from 12 times to 18 times.

From the above Fig. 5.1 it can be concluded to a decision that for all the three different data sets and different cryptosystems, the size of the ciphertext becomes sufficiently larger than the size of the plaintext. Actually this is usual for cryptographic operations especially while an asymmetric cryptosystem is exploited. The underlying reason is, the exponentiation operation (s) involved in the encryption stage of the asymmetric cryptosystem makes the data size of the ciphertext so bigger.



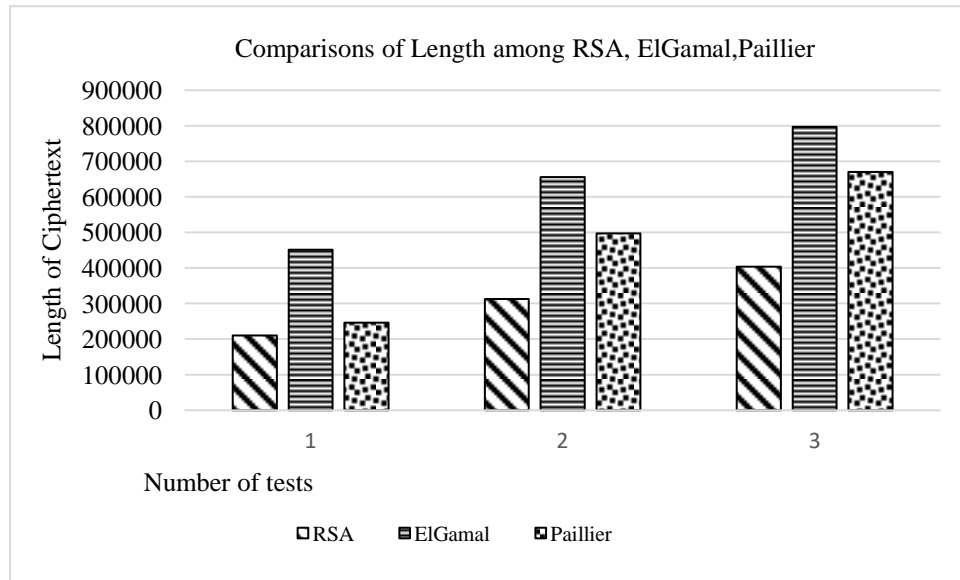


Figure 5.2: Comparisons of length of ciphertext among RSA, ElGamal, Paillier

Fig. 5.2 dissertates the comparative results with respect to the length of ciphertext among RSA, ElGamal and Paillier cryptosystems. It has been observed that in all case the length of ciphertext of ElGamal is greater than RSA and Paillier.

The proposed technique mainly focuses on security rather than space. Here, dynamic sequence table and dynamic DNA encoding are used with asymmetric cryptosystems to ensure more security than the traditional asymmetric cryptosystem.

### 5.3.2 Comparisons with respect to time

A comparative analysis between encryption time and decryption time was performed on the no. of character i.e. plaintext 18954, 27632, 34219 respectively. It is presented on Fig. 5.3. It has been observed that the required decryption time is greater than encryption time for RSA. It is because both RSA encryption and decryption involve modular exponentiation, but whereas the public encryption exponent is normally small and fixed (usually either 3 or  $2^{16}+1 = 65,537$ ), the secret decryption exponent is usually almost as long as the modulus. Thus, doubling the modulus size makes encryption take twice as long, but makes decryption take four times as long.

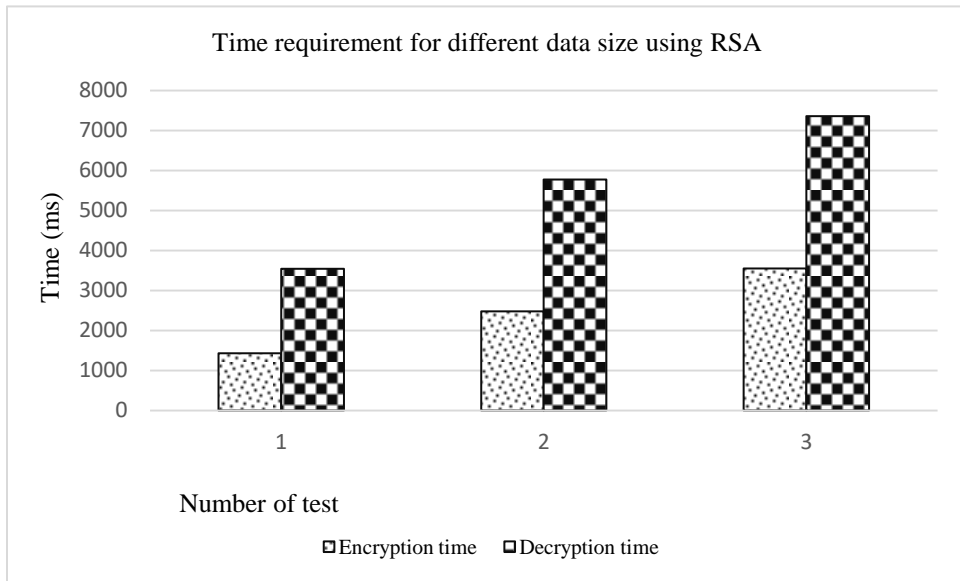


Figure 5.3 (a)

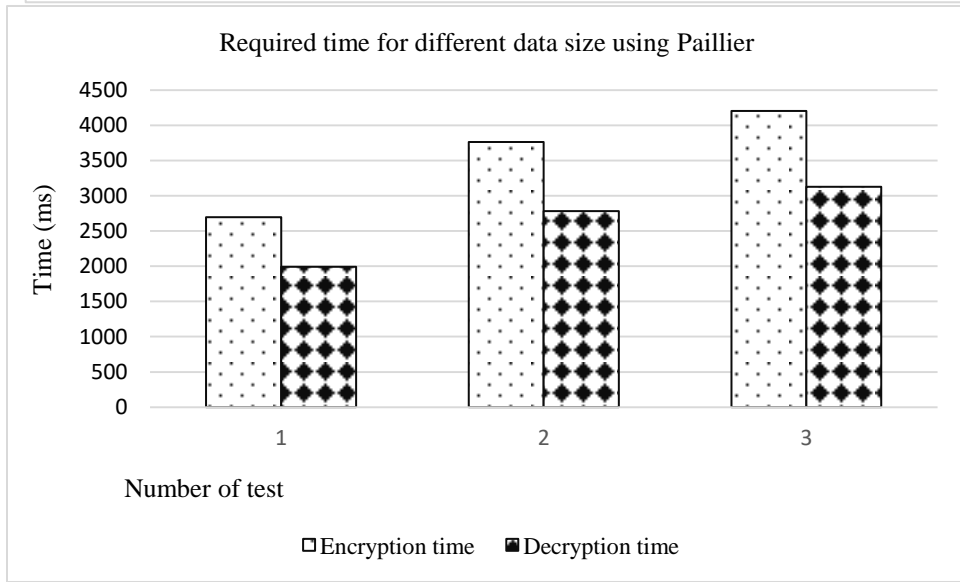
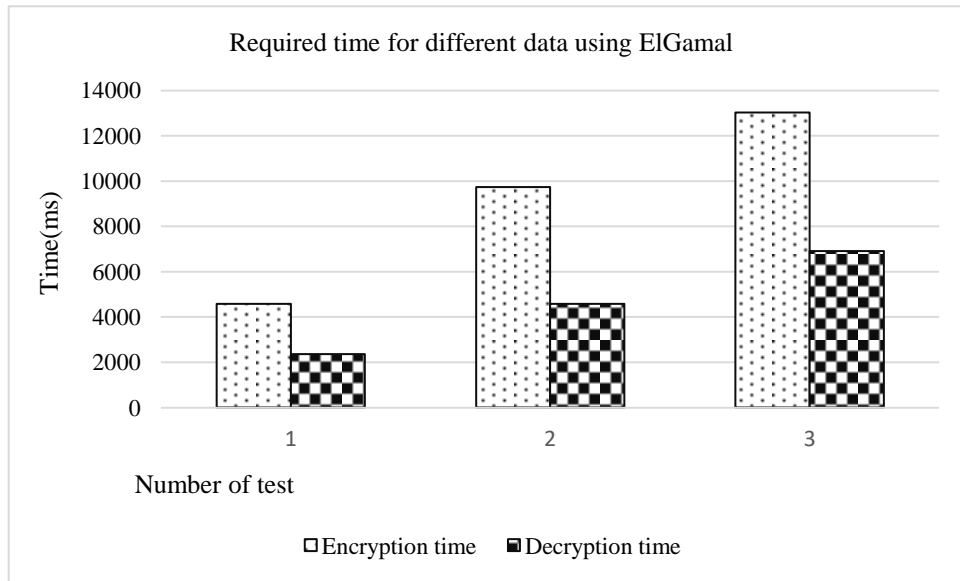


Figure 5.3 (c)

Figure 5.3 Comparisons between encryption time and decryption time for different data size. (a) RSA; (b) ElGamal; (c) Paillier.

For ElGamal cryptosystems, it performs two modular exponentiation operations. For this reason the encryption time is greater than the decryption time. For Paillier cryptosystems, the decryption time requires less than encryption time.

### 5.3.3 Comparison of time among asymmetric cryptosystems based proposed technique

It has been observed from Fig. 5.4 that the required encryption time for ElGamal is highest. As the characteristics of Paillier is probabilistic, it's elapsed time varies into a range. While comparing the decryption time RSA requires highest time which has been shown in Fig. 5.5.

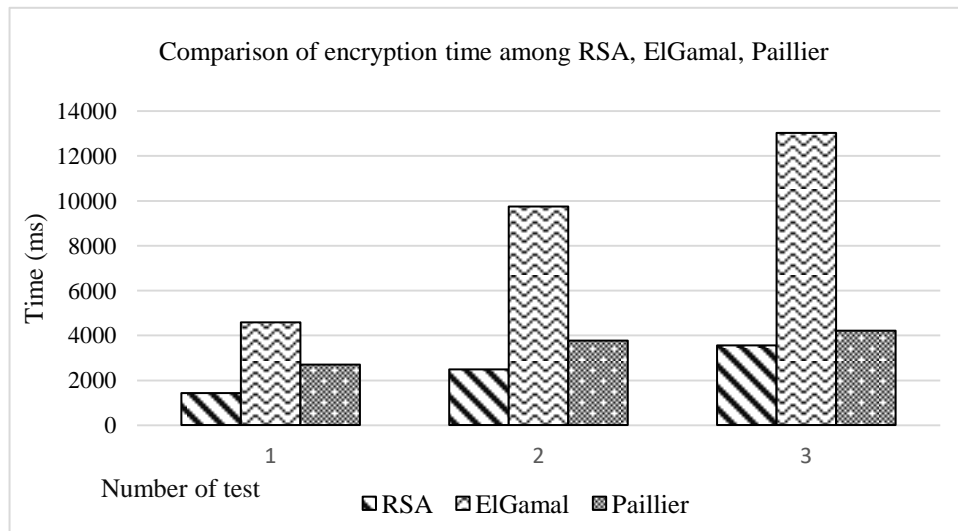


Figure 5.4: Comparisons of encryption time among RSA, ElGamal and Paillier based proposed technique

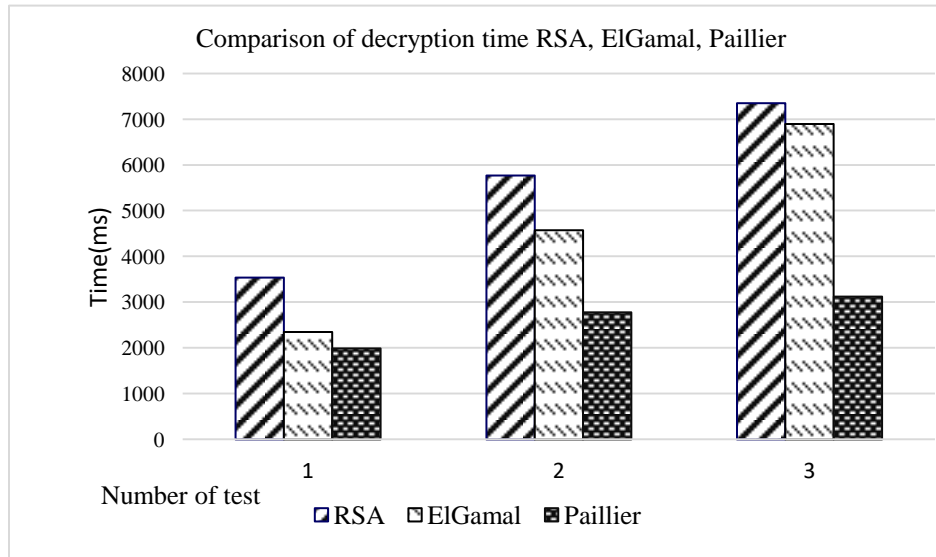


Figure 5.5: Comparisons of decryption time among RSA, ElGamal and Paillier based proposed technique

### 5.3.4 Comparisons with other related techniques

Considering the time requirement of encryption and decryption processes, comparisons with techniques proposed in [35] and [36] are presented in Fig. 5.6 and Fig. 5.7, respectively. Here to compare with other techniques we implement our proposed technique with Paillier cryptosystems.

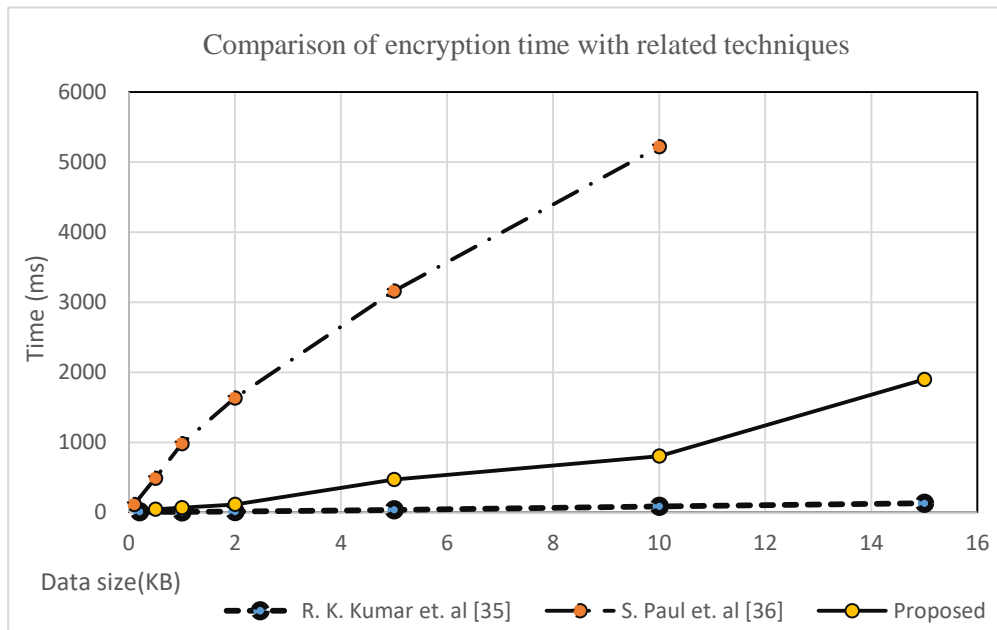


Figure 5.6: Comparisons of encryption time with other related techniques.

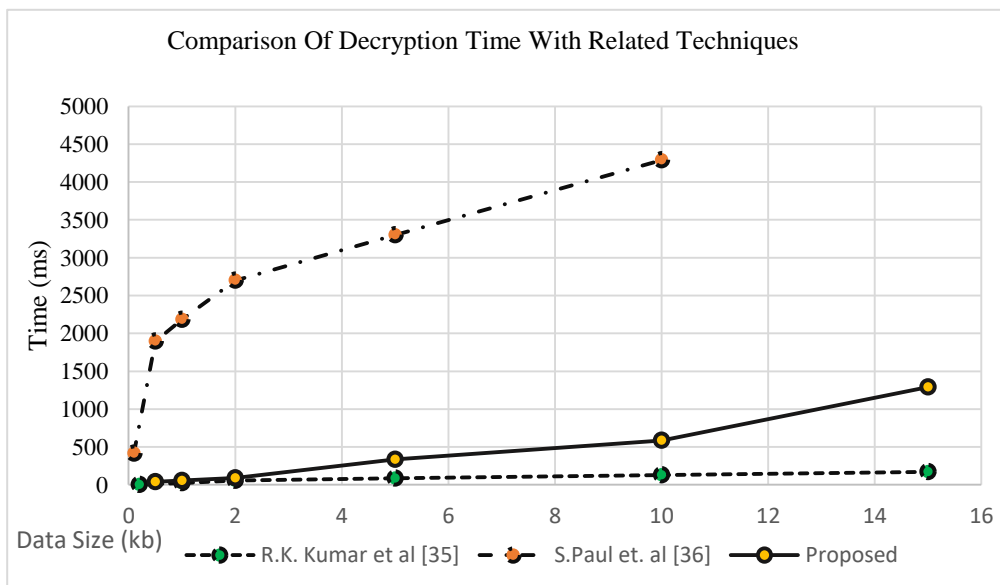


Figure 5.7: Comparisons of decryption time with other related techniques.

For the proposed technique, the result of Paillier based implementation is considered. From the figure it is seen that, the technique proposed in [36] requires comparatively large time whereas the technique proposed in [35] requires comparatively less time than the proposed technique.

The reason is, the technique proposed in [36] considers symmetric key exchange, matrix operation and XOR technique. In contrast, the technique proposed in [35] simply considers only random substitution of DNA bases with 256 ASCII characters; do not consider any other robust tool. But in our proposed technique we consider dynamisms with an asymmetric cryptosystems. To increase the secrecy level we performed a number of iteration of the dynamic sequence table. Thus, it needs comparatively more time than the technique proposed in [36].

However, if iteration of dynamic sequence table is reduced and the number of chunk is decreased then it will need less time for the proposed technique.

The execution time it needs is comparatively less. This technique can be applied in any real time application without any loss of data.

## CHAPTER VI

### Security and Statistical Analysis

#### 6.1 Introduction

This chapter discusses the randomness of key and security of ciphertext. For this reason, the key strength has been analyzed and the National Institute of Standards and Technology (NIST) [37] test has been performed.

#### 6.2 Ciphertext Strength Analysis

Three level of security are embedded with generating ciphertext. Firstly, generating of dynamic sequence table from plaintext. Secondly, apply 2048 bit asymmetric key to encrypt the message. Finally dynamic DNA encoding is applied to merge the ciphertext of each chunk.

##### Level 1:

- Firstly each ASCII character is converted to ASCII value in integer form. Then it is converted to binary value. Each binary value is converted to DNA bases. The DNA base sequence that means four (4) DNA bases can be selected by  $4!$
- 256 ASCII character are randomly assigned to each DNA sequence. Here the ASCII value can be assigned by  $256!$  Numbers.
- Here the table is iterated from left to right or right to left according to Fibonacci series. Here the complexity is of finding the series is  $n$ . Where  $n$  is the chosen value from Fibonacci series.

Total complexity of level 1 =  $4! * 256! * n$ .

It has been observed from the proposed technique on [34] that to test all the possible combination for  $26!$  Cases a computer will take 12 million years. Here the computer processing power is 1,000,000 keys per second. According to this configuration of computer, total elapsed time to break the secrecy of level 1 will be,  $\text{Time} = (4! * 256! * n * 12 \text{ million}) / 26!$  Years.

**Level 2:**

To encrypt the message we choose 2048 bit key. According to the FIPS 140-2 Implementation Guidance of NIST [38], we found the key length for RSA in equation 6.1.

Key length

$$x = \frac{1.932 \times \sqrt[3]{L \times \ln(2)} \sqrt[3]{[\ln(L \times \ln(2))]^2}}{\ln(2)} \dots \dots \dots (6.1)$$

2048 bit RSA key is expressed as the modulus in the expression.  $2^b$  is the approximation of the time to need to factor a  $b$ -bit integer. For  $2^{2048}$  length, the strength of RSA is about around 112.

In level 2 to break the secrecy of 2048 bit RSA, the attacker has to perform  $2^{112}$  times brute force attack on the key.

**Level 3:**

In dynamic DNA encoding portion, sufficient random binary sequence is generated. To break the secrecy of random sequence a mathematical calculation is given in equation 6.2

$$keyspace = alphabet^{length} \dots \dots \dots (6.2)$$

Where, *keyspace = the number of possible codes*

*length = the amount of randomly entered character*

*alphabet = alphabet of size*

*Rate = Guess rate*

For average case, the probability of success rate is calculated as equation 6.3

$$expectedtime = probability * \frac{keyspace}{rate \times 2} \dots \dots \dots (6.3)$$

For above equation, let the probability of success is 10% and the keyspace size is 1 billion number. The attacker can perform brute force attack on 1 million number per second. Then the attacker needs fifty seconds as equation 6.4



$$expectedtime = 10\% \times 1 \frac{billion}{1million \times 2} \dots \dots \dots (6.4)$$

But if an attacker can perform parallel attack on random sequence then the required time is reduced. Then the time needed as equation 6.5,

$$expectedtime = probability \times \frac{keyspace}{rate \times amount \times 2} \dots \dots \dots (6.5)$$

### 6.3 NIST Statistical Test

Randomness testing of data (both key and ciphertext) is an important factor in the field of cryptography. Secrecy of data depends on how much variations the generated ciphertext has from the plaintext. Mostly used test suits for data is National Institute of Standards and Technology (NIST) test [33]. It provides standardized test sets for random number and pseudorandom number. To compare and evaluate if the random sequence is truly random sequence, different kinds of statistical tests can be applied here. The property of randomness is probabilistic. It means the property of a random sequence can be characterized and described in terms of probability.

A statistical test is expressed to test a specific **null hypothesis (H0)**. For the reason, the null hypothesis (H0) is performed where the sequence is random. Accompanying with this null hypothesis (H0) another is the **alternative hypothesis (Ha)** where the sequence is not random.

This test statistical results is compared to the **critical value**. If the test statistical value exceeds the critical value, the null hypothesis for randomness is rejected. Otherwise, the null hypothesis (the randomness hypothesis) is *not* rejected (i.e., the hypothesis is accepted).

**P-value** precise the strength of the evidence against the null hypothesis. If a P-value for a test is determined to be equal to 1, then the sequence appears to have perfect randomness. A P-value of zero indicates that the sequence appears to be completely non-random. A **significance level (α)** can be chosen for the tests. If P-value  $\geq \alpha$ , then the null hypothesis is accepted; i.e., the sequence appears to be random. If P-value  $< \alpha$ , then the null hypothesis is rejected.

## 6.4 Random Number Generation Tests

The NIST Test Suite is a statistical package consist of 15 tests that were developed to test the randomness of binary sequences which is based cryptographic random or pseudorandom number generators. The 15 tests are:

1. The Frequency Test,
2. Frequency Test within a Block,
3. The Runs Test,
4. Tests for the Longest-Run-of-Ones in a Block,
5. The Binary Matrix Rank Test,
6. The Discrete Fourier Transform (Spectral) Test,
7. The Non-overlapping Template Matching Test,
8. The Overlapping Template Matching Test,
9. Maurer's "Universal Statistical" Test,
10. The Linear Complexity Test,
11. The Serial Test,
12. The Approximate Entropy Test,
13. The Cumulative Sums Test,
14. The Random Excursions Test, and
15. The Random Excursions Variant Test.

In this section each test is briefly described and its corresponding mathematical expression is explained.

### 6.4.1 The Frequency Test

The purpose of this test is to determine whether the proportion of zeroes and ones for the entire sequence are same as would be expected for a truly random sequence.

The test is derived from the well-known limit Central Limit Theorem for the random walk,

$$S_n = X_1 + \dots + X_n.$$

#### Rules:

- a) Conversion to  $\pm 1$ : The zeros and ones of the input sequence ( $\epsilon$ ) are converted to values of  $-1$  and  $+1$  and are added together to produce  $S_n = X_1 + X_2 + \dots + X_n$ , where  $X_i = 2\epsilon_i - 1$ .

b) Compute the test statistic  $S_{obs} = \frac{[Sn]}{\sqrt{n}}$

c) Compute P-value =  $\text{erfc} \left( \frac{S_{obs}}{\sqrt{2}} \right)$

**Decision:**

If the computed *P-value* is < 0.01, then conclude that the sequence is non-random. Otherwise, conclude that the sequence is random.

**6.4.2 Frequency Test within a Block**

The focus of the test is the proportion of ones within M-bit blocks.

The parameters of this test are M and N, then n=MN. The sum

$$X^2(\text{obs}) = 4M \sum_{i=1}^N \left[ \pi_i - \frac{1}{2} \right]^2 \dots \dots \dots (2)$$

The P-value is

$$\frac{\int_{x^2(\text{obs})}^{\infty} e^{-\frac{u}{2}} u^{\frac{N}{2}-1} du}{\tau\left(\frac{N}{2}\right) 2^{\frac{N}{2}}} = \frac{\int_{x^2(\text{obs})}^{\infty} e^{-u} u^{\frac{N}{2}-1} du}{\tau\left(\frac{N}{2}\right)} = \text{igamc}\left(\frac{N}{2}, \frac{x^2(\text{obs})}{2}\right)$$

**Rules:**

a) Partition the input sequence into  $N = \left\lceil \frac{n}{M} \right\rceil$  non-overlapping blocks.

b) Determine the proportion  $\pi_i$  of ones in each M-bit block using the equation

$$\pi_i = \frac{\sum_{j=1}^M \delta_{(i-1)M+j}}{M}$$

c) Compute the  $\chi^2$  statistic  $\chi^2(\text{obs}) = 4M \sum (\pi_i - 1/2)^2$ .

**Decision:** If the computed *P-value* is < 0.01, then conclude that the sequence is non-random. Otherwise, conclude that the sequence is random.

### 6.4.3 Runs Test

A run is an uninterrupted sequence of identical bits. The focus of this test is the total number of runs in the sequence. The purpose of the runs test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence.

$V_n$ , define for  $k = 1, \dots, n-1$ ,  $r(k) = 0$  if  $\varepsilon_k = \varepsilon_{k+1}$  and  $r(k) = 1$  if  $\varepsilon_k \neq \varepsilon_{k+1}$ . Then  $V_n = \sum_{k=1}^{n-1} r(k) + 1$ .

$$P\text{-value} = \text{erfc} \left( \frac{|V_n(\text{obs}) - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)} \right)$$

#### Rules:

a) Pre-test proportion  $\pi$  of ones in the input sequence  $\pi = \frac{\sum_j \varepsilon_j}{n}$

b) Determine if the prerequisite Frequency test is passed. If it can be shown that  $\left| \pi - \frac{1}{2} \right| \geq x$ , then the Runs test need not be performed and the *P-value* is set to 0.0000.

c) Test statistic  $V(\text{obs}) = \sum_{k=1}^{n-1} r(k) + 1$

$$d) P\text{-value} = \text{erfc} \left( \frac{|V_n(\text{obs}) - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)} \right)$$

### 6.4.4 Test for the Longest Run of Ones in a Block

The focus of the test is the longest run of ones within M-bit blocks. An irregularity in the expected length of the longest run of ones implies that there is also an irregularity in the expected length of the longest run of zeroes.

$$P \text{ value} = \text{igamc} \left( \frac{N}{2}, \frac{x^2(\text{obs})}{2} \right)$$

$$X^2 = \sum_{i=0}^k \frac{(v_i - N\pi_i)^2}{N\pi_i}$$

a) Divide the sequence into M-bit blocks.

- b) Tabulate the frequencies  $v_i$  of the longest runs of ones in each block into categories.
- c) Compute  $\chi^2$  (obs).
- d) Compute P-value.

#### 6.4.5 Rank Test

The focus of the test is the rank of disjoint sub-matrices of the entire sequence. The purpose of this test is to check for linear dependence among fixed length substrings of the original sequence.

$$\text{P-value} = \exp\{-X^2(\text{obs})/2\}$$

$$X^2 = \frac{(F_m - 0.288N)^2}{0.288N} + \frac{(F_{m-1} - 0.5776N)^2}{0.5776N} + \frac{(N - F_m - F_{m-1} - 0.1336N)^2}{0.1336N}$$

#### Rules:

- a) Sequentially divide the sequence into  $M \cdot Q$ -bit disjoint blocks.
- b) Determine the binary rank ( $R_{\square}$ ) of each matrix.
- c) Compute  $\chi^2$  (obs).
- d) Compute P-value.

#### 6.4.6 Discrete Fourier Transform (Spectral) Test

The purpose of this test is to detect periodic features in the tested sequence that would indicate a deviation from the assumption of randomness. The intention is to detect whether the number of peaks exceeding the 95 % threshold is significantly different than 5 %.

*P-value* is

$$2(1 - \varphi(|d|)) = \text{erfc}\left(\frac{|d|}{\sqrt{2}}\right)$$

#### Rules:

- (a) The zeros and ones of the input sequence ( $\epsilon$ ) are converted to values of  $-1$  and  $+1$  to create the sequence  $X = x_1, x_2, \dots, x_n$ , where  $x_i = 2\epsilon_i - 1$ .
- (b) Apply a Discrete Fourier transform (DFT) on  $X$  to produce:  $S = \text{DFT}(X)$ .

(c) Calculate  $M = \text{modulus}(S') \equiv |S'|$ , where  $S'$  is the substring consisting of the first  $n/2$  elements in  $S$ ,

(d) Compute  $T = \sqrt{\log\left(\frac{1}{0.05}\right)n}$

(e) Compute  $N0 = .95n/2$ .  $N0$

(f) Compute  $N1$  = the actual observed number of peaks in  $M$  that are less than  $T$ .

(g) Compute  $d = \frac{N1 - N0}{\sqrt{n(.95)(.05)/4}}$

(h) Compute  $P$ -value.

#### 6.4.7 Non-overlapping Template Matching Test

The focus of this test is the number of occurrences of pre-specified target strings. The purpose of this test is to detect generators that produce too many occurrences of a given non-periodic (aperiodic) pattern.

$$W = \sum_{i=1}^{n-m+1} I(\epsilon_{i+k-1} = \epsilon_k^0, k = 1, \dots, m)$$

$$\mu = \frac{n-m+1}{2^m}$$

$$\sigma^2 = n \left[ \frac{1}{2^m} - \frac{2m-1}{2^{2m}} \right]$$

$$\chi^2(\text{obs}) = \sum_{j=1}^N \frac{(W_j - \mu)^2}{\sigma^2}$$

$$P\text{-value} = 1 - P\left(\frac{N}{2}, \frac{\chi^2(\text{obs})}{2}\right)$$

(1) Partition the sequence into  $N$  independent blocks of length  $M$ .

(2) Under an assumption of randomness, compute the theoretical mean  $\mu$  and variance  $\sigma^2$ .

(3) Compute  $\chi^2(\text{obs})$ .

(4) Compute  $P$ -value.

#### 6.4.8 Overlapping Template Matching Test

The focus of the Overlapping Template Matching test is the number of occurrences of pre-specified target strings. The difference between this test and the test in Section 6.3.7 is that when the pattern is found, the window slides only one bit before resuming the search.

**Rules:**

- (1) Partition the sequence into  $N$  independent blocks of length  $M$ .
- (2) Calculate the number of occurrences of  $B$  in each of the  $N$  blocks.
- (3) Compute values for  $\lambda$  and  $\eta$  that will be used to compute the theoretical probabilities  $\pi_i$  corresponding to the classes of  $v_0$ :  $\lambda = (M-m+1)/2^m$   $\eta = \lambda/2$ .
- (4) Compute  $\chi^2$  ( *obs* ).
- (5) Compute  $P$ -value.

**6.4.9 Maurer’s “Universal Statistical” Test**

The focus of this test is the number of bits between matching patterns (a measure that is related to the length of a compressed sequence). The purpose of the test is to detect whether or not the sequence can be significantly compressed without loss of information.

**Rules:**

- (1) The  $n$ -bit sequence ( $\epsilon$ ) is partitioned into two segments: an initialization segment consisting of  $Q$   $L$ -bit non-overlapping blocks, and a test segment consisting of  $KL$ -bit non-overlapping blocks.
- (2) Using the initialization segment, a table is created for each possible  $L$ -bit value (i.e., the  $L$ -bit value is used as an index into the table).
- (3) Examine each of the  $K$  blocks in the test segment and determine the number of blocks since the last occurrence of the same  $L$ -bit block.

**6.4.10 Linear Complexity Test**

The focus of this test is the length of a linear feedback shift register (LFSR). The purpose of this test is to determine whether or not the sequence is complex enough to be considered random.

- (1) Under an assumption of randomness, calculate the theoretical mean  $\mu$ :

$$\mu = \frac{M}{2} + \frac{(9 + (-1)^{M+1})}{36} - \frac{(M/3 + 2/9)}{2^M}.$$

(2) For each substring, calculate a value of  $T_i$ , where  $T = (-1) \cdot (L - \mu) + \frac{2}{9}$ .

(3) Compute  $\chi^2 (obs) = \sum_{i=0}^K \frac{(v_i - N\pi_i)^2}{N\pi_i}$

(4) Compute  $P\text{-value} = \text{igamc} \left( \frac{K}{2}, \frac{\chi^2(obs)}{2} \right)$ .

#### 6.4.11 Serial Test

The focus of this test is the frequency of all possible overlapping  $m$ -bit patterns across the entire sequence. Random sequences have uniformity; that is, every  $m$ -bit pattern has the same chance of appearing as every other  $m$ -bit pattern.

##### Rules:

(1) Form an augmented sequence  $\varepsilon'$ : Extend the sequence by appending the first  $m-1$  bits.

(2) Determine the frequency of all possible overlapping  $m$ -bit blocks.

(3) Compute:  $\psi^2 = \frac{2^m}{n} \sum_{i_m} (v_{i_1 \dots i_m} - \frac{n}{2^m})^2$

(4) Compute:  $\nabla \psi^2_{m-1} = \psi^2_{m-1}$

(5) Compute:  $P\text{-value} = \text{igamc} \left( 2^{m-2}, \nabla \psi^2_m \right)$

#### 6.4.12 Approximate Entropy Test

The focus of this test is the frequency of all possible overlapping  $m$ -bit patterns across the entire sequence. The purpose of the test is to compare the frequency of overlapping blocks of two consecutive/adjacent lengths ( $m$  and  $m+1$ ) against the expected result for a random sequence.

##### Rules:

(1) Augment the  $n$ -bit sequence to create  $n$  overlapping  $m$ -bit sequences by appending  $m-1$  bits from the beginning of the sequence to the end of the sequence.

(2) A frequency count is made of the  $n$  overlapping blocks.



(3) Compute  $C_i^m = \frac{\#i}{n}$

(4) Compute the test statistic:  $\chi^2 = 2n [\log 2 - ApEn(m)]$ .

(5) Compute  $P\text{-value} = igamc(2^{m-1}, \frac{\chi^2}{2})$

#### 6.4.13 Cumulative Sums Test

The focus of this test is the maximal excursion (from zero) of the random walk defined by the cumulative sum of adjusted (-1, +1) digits in the sequence. The purpose of the test is to determine whether the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of that cumulative sum for random sequences.

##### Rules:

(1) Form a normalized sequence: The zeros and ones of the input sequence ( $\epsilon$ ) are converted to values  $X_i$  of -1 and +1 using  $X_i = 2\epsilon_i - 1$ .

(2) Compute partial sums  $S_i$  of successively larger subsequences, each starting with  $X_1$  (if  $mode = 0$ ) or  $X_n$  (if  $mode = 1$ ).

(3) Compute the test statistic  $z = \max_{1 \leq k \leq n} |S_k|$ , where  $\max_{1 \leq k \leq n} |S_k|$  is the largest of the absolute values of the partial sums  $S_k$ .

(4) Compute  $P\text{-value}$

#### 6.4.14 Random Excursions Test

The focus of this test is the number of cycles having exactly  $K$  visits in a cumulative sum random walk. The purpose of this test is to determine if the number of visits to a particular state within a cycle deviates from what one would expect for a random sequence.

(1) Form a normalized (-1, +1) sequence  $X$ : The zeros and ones of the input sequence ( $\epsilon$ ) are changed to values of -1 and +1 via  $X_i = 2\epsilon_i - 1$ .

(2) Compute the partial sums  $S_i$  of successively larger subsequences, each starting with  $X_1$ . Form the set  $S = \{S_i\}$ .

(3) Form a new sequence  $S'$  by attaching zeros before and after the set  $S$ . That is,  $S' = 0, s_1, s_2, \dots, s_n, 0$ .

$$(4) \text{Statistic } \chi^2(\text{obs}) = \sum_{k=0}^5 \frac{(v_k(x) - J\pi_k(x))^2}{J\pi_k(x)}$$

$$(5) \text{P-value} = \text{igamc}(5/2, \chi^2(\text{obs})).$$

#### 6.4.15 Random Excursions Variant Test

The focus of this test is the total number of times that a particular state is visited (i.e., occurs) in a cumulative sum random walk. The purpose of this test is to detect deviations from the expected number of visits to various states in the random walk.

##### Rules:

- (1) Form the normalized (-1, +1) sequence  $X$  in which the zeros and ones of the input sequence ( $\varepsilon$ ) are converted to values of -1 and +1 via  $X = X_1, X_2, \dots, X_n$ , where  $X_i = 2\varepsilon_i - 1$ .
- (2) Compute partial sums  $S_i$  of successively larger subsequences, each starting with  $x_1$ . Form the set  $S = \{S_i\}$ .
- (3) Form a new sequence  $S'$  by attaching zeros before and after the set  $S$ . That is,  $S' = 0, s_1, s_2, \dots, s_n, 0$ .
- (4) For each of the eighteen non-zero states of  $x$ , compute  $\xi(x)$  = the total number of times that state  $x$  occurred across all  $J$  cycles.

- (5) Compute *P-value*

#### 6.5 Analysis the results of the proposed technique

In this section the resultant output for  $10^6$  sized binary value of the generated ciphertext is analyzed under NIST test. Each 100 binary sequences are applied a block of data. The level of significance ( $\alpha$ ) is 0.01. To test the output must be in binary form. For each test the P-value  $\geq 0.01$  is considered as succeeded. Otherwise the output of the result is considered as fail. For each test the output is shown below.

##### 6.5.1 The Frequency Test

According to the section of 6.4.1 the output result of the Frequency Test is shown in Table 6.1.

Table 6.1 Frequency Test

<b>P_value</b>	<b>Rate</b>
0.254286	SUCCESS
0.6599370	SUCCESS
0.2076690	SUCCESS
0.8414810	SUCCESS
0.548506	SUCCESS
0.327086	SUCCESS
0.034006	SUCCESS
0.065768	SUCCESS
0.603064	SUCCESS
0.617075	SUCCESS

### 6.5.2. Frequency Test within a Block

According to the section of 6.4.2 the output result of the Block Frequency Test is shown in Table 6.2

Table 6.2 Block Frequency Test

<b>P_value</b>	<b>Rate</b>
0.408377	SUCCESS
0.753495	SUCCESS
0.992512	SUCCESS
0.964120	SUCCESS
0.404572	SUCCESS
0.583290	SUCCESS
0.011942	SUCCESS
0.228786	SUCCESS
0.826012	SUCCESS
0.864119	SUCCESS

### 6.5.3 The Runs Test

According to the section of 6.4.3 the output result of the Runs Test is shown in Table 6.3

Table 6.3 Runs Test

<b>P_value</b>	<b>Rate</b>
0.669336	SUCCESS
0.618432	SUCCESS
0.137714	SUCCESS
0.841793	SUCCESS
0.069299	SUCCESS
0.087162	SUCCESS
0.578686	SUCCESS
0.475167	SUCCESS
0.297072	SUCCESS
0.002905	FAILURE

#### 6.5.4 Tests for the Longest-Run-of-Ones in a Block

According to the section of 6.4.4 the output result of the Longest Runs Test is shown in Table 6.4

Table 6.4 Longest Runs Test

<b>P_value</b>	<b>Rate</b>
0.701135	SUCCESS
0.201082	SUCCESS
0.455465	SUCCESS
0.621593	SUCCESS
0.341382	SUCCESS
0.804919	SUCCESS
0.122349	SUCCESS
0.343385	SUCCESS
0.477678	SUCCESS
0.999573	SUCCESS

### 6.5.5 Rank Test

According to the section of 6.4.5 the output result of the Rank Test is shown in Table 6.5

Table 6.5 Rank Test

<b>P_value</b>	<b>Rate</b>
0.157494	SUCCESS
0.212245	SUCCESS
0.374306	SUCCESS
0.159044	SUCCESS
0.862457	SUCCESS
0.330688	SUCCESS
0.374306	SUCCESS
0.066920	SUCCESS
0.862457	SUCCESS
0.648387	SUCCESS

### 6.5.6 Discrete Fourier Transform (Spectral) Test

According to the section of 6.4.6 the output result of the FFT Test is shown in Table 6.6

Table 6.6 FFT Test

<b>P_value</b>	<b>Rate</b>
0.854380	SUCCESS
0.783087	SUCCESS
0.081236	SUCCESS
0.003284	FAILURE
0.408863	SUCCESS
0.098577	SUCCESS
0.926884	SUCCESS
0.926884	SUCCESS
0.098577	SUCCESS
0.783087	SUCCESS

### 6.5.7 The Non-overlapping Template Matching Test

According to the section of 6.4.7 the output result of the Non Overlapping Test is shown in Table 6.7

Table 6.7 Non Overlapping test

NONPERIODIC TEMPLATES TEST

-----  
 COMPUTATIONAL INFORMATION  
 -----

LAMBDA = 2.425781      M = 1250      N = 8    m = 9    n = 10000  
 -----

FREQUENCY

Template    W\_1   W\_2   W\_3   W\_4   W\_5   W\_6   W\_7   W\_8   Chi^2   P\_value   Assignment  
 Index

-----

000000001	1	2	3	3	5	3	4	2	5.291339	0.726032	SUCCESS	0
000000011	3	4	1	2	6	2	1	2	8.554899	0.381222	SUCCESS	1
000000101	2	2	4	1	4	1	3	5	6.923119	0.544950	SUCCESS	2
000000111	1	1	2	0	3	1	1	2	6.231350	0.621335	SUCCESS	3
000001001	1	1	1	0	0	1	4	4	10.530909	0.229720	SUCCESS	4
000001011	2	3	6	0	2	0	8	1	24.717136	0.001736	FAILURE	5
000001101	5	4	3	5	2	3	0	4	10.564008	0.227647	SUCCESS	6
000001111	1	1	0	2	1	6	2	2	10.719573	0.218100	SUCCESS	7
000010001	1	1	6	3	4	5	5	0	16.432459	0.036592	SUCCESS	8
000010011	4	1	2	1	1	1	2	4	5.698457	0.680967	SUCCESS	9
000010101	2	2	1	1	5	1	2	3	5.761345	0.673945	SUCCESS	10
000010111	2	4	5	2	5	1	4	2	8.806452	0.358886	SUCCESS	11
000011001	3	1	0	0	2	0	1	4	10.468021	0.233700	SUCCESS	12
000011011	3	3	4	5	1	2	1	4	6.986007	0.538143	SUCCESS	13
000011101	3	2	3	2	5	2	2	2	3.470895	0.901437	SUCCESS	14
000011111	2	3	2	3	3	6	3	3	6.264449	0.617636	SUCCESS	15
000100011	1	1	7	2	3	3	2	0	13.513128	0.095372	SUCCESS	16
000100101	1	3	1	0	4	1	1	3	7.267349	0.508082	SUCCESS	17
000100111	2	2	0	0	1	4	1	2	7.988907	0.434555	SUCCESS	18

000101001 4 1 3 1 2 3 3 1 4.129564 0.845248 SUCCESS 19  
000101011 2 2 1 3 6 2 5 2 9.528010 0.299727 SUCCESS 20  
000101101 5 2 0 1 1 1 2 3 8.177571 0.416320 SUCCESS 21  
000101111 1 3 4 2 4 1 3 1 5.039787 0.753315 SUCCESS 22  
000110011 2 1 4 1 3 0 1 1 7.204461 0.514742 SUCCESS 23  
000110101 3 5 1 2 3 4 1 2 6.012897 0.645787 SUCCESS 24  
000110111 2 0 2 0 0 5 1 3 11.441131 0.177934 SUCCESS 25  
000111001 3 2 2 6 2 2 1 3 6.860231 0.551785 SUCCESS 26  
000111011 3 3 4 5 3 3 7 2 13.357563 0.100125 SUCCESS 27  
000111101 2 3 0 3 2 2 4 3 4.192453 0.839356 SUCCESS 28  
000111111 0 4 0 6 3 3 2 1 12.665794 0.123878 SUCCESS 29  
001000011 4 1 1 2 1 3 1 4 5.761345 0.673945 SUCCESS 30  
001000101 3 1 2 1 4 2 5 0 8.366235 0.398537 SUCCESS 31  
001000111 2 4 2 1 4 2 2 1 4.129564 0.845248 SUCCESS 32  
001001011 1 4 2 3 5 0 2 5 10.312456 0.243777 SUCCESS 33  
001001101 3 4 8 0 2 5 6 3 25.283129 0.001392 FAILURE 34  
001001111 2 1 0 1 1 2 2 2 5.384016 0.715853 SUCCESS 35

#### 6.4.8 The Overlapping Template Matching Test

According to the section of 6.4.8 the output result of the Overlapping Template Test is shown in Table 6.8

Table 6.8 Overlapping Template

<b>P-Value</b>	<b>Rate</b>
0.287317	SUCCESS
0.343160	SUCCESS
0.673149	SUCCESS
0.905643	SUCCESS
0.006393	FAILURE
0.179453	SUCCESS
0.309028	SUCCESS
0.037005	SUCCESS
0.618602	SUCCESS
0.456546	SUCCESS

### 6.5.9 Maurer's "Universal Statistical" Test

According to the section of 6.4.9 the output result of the Universal Statistical Test is shown in Table 6.9

Table 6.9 Universal test

UNIVERSAL STATISTICAL TEST

-----  
 ERROR: L IS OUT OF RANGE.

-OR- : Q IS LESS THAN 320.000000.

-OR- : Unable to allocate T. UNIVERSAL STATISTICAL TEST

### 6.5.10 The Linear Complexity Test

According to the section of 6.4.10 the output result of the Linear Complexity Test is shown in Table 6.10

Table 6.10 Linear Complexity Test

<b>P_value</b>	<b>Rate</b>
0.845322	SUCCESS
0.239819	SUCCESS
0.582685	SUCCESS
0.198826	SUCCESS
0.481329	SUCCESS
0.928566	SUCCESS
0.609198	SUCCESS
0.833365	SUCCESS
0.928569	SUCCESS
0.505815	SUCCESS

### 6.5.11 The Serial Test

According to the section of 6.4.11 the output result of the Serial Test is shown in Table 6.11

Table 6.11 Serial Test

Test	
SUCCESS	p_value1 = 0.600085
SUCCESS	p_value2 = 0.350752
SUCCESS	p_value1 = 0.938169
SUCCESS	p_value2 = 0.964099



SUCCESS	p_value1 = 0.762861
SUCCESS	p_value2 = 0.826621
SUCCESS	p_value1 = 0.580161
SUCCESS	p_value2 = 0.666933
SUCCESS	p_value1 = 0.165289
SUCCESS	p_value2 = 0.570425
SUCCESS	p_value1 = 0.498961
SUCCESS	p_value2 = 0.377875
SUCCESS	p_value1 = 0.885780
SUCCESS	p_value2 = 0.913212
SUCCESS	p_value1 = 0.754853
SUCCESS	p_value2 = 0.324383
SUCCESS	p_value1 = 0.109819
SUCCESS	p_value2 = 0.192203
SUCCESS	p_value1 = 0.998206
SUCCESS	p_value2 = 0.994094

### 6.5.12 The Approximate Entropy Test

According to the section of 6.4.12 the output result of the Entropy Test is shown in Table 6.12

Table 6.12 Entropy Test

<b>P_value</b>	<b>Rate</b>
0.017977	SUCCESS
0.000211	FAILURE
0.000352	FAILURE
0.095714	SUCCESS
0.000607	FAILURE
0.000264	FAILURE
0.028771	SUCCESS
0.000213	FAILURE
0.007446	FAILURE
0.247591	SUCCESS

### 6.5.13 The Cumulative Sums (Cusums) Test

According to the section of 6.4.13 the output result of the Cumulative Test is shown in Table 6.13

Table 6.13 Cumulative Test

<b>P_value</b>	<b>Rate</b>
0.262075	SUCCESS
0.769148	SUCCESS
0.857965	SUCCESS
0.008414	FAILURE
0.620100	SUCCESS
0.048898	SUCCESS
0.009023	FAILURE
0.057048	SUCCESS
0.515555	SUCCESS
0.206201	SUCCESS

#### 6.5.14 The Random Excursions Test

According to the section of 6.4.14 the output result of the Random Excursions Test is shown in Table 6.14

Table 6.14 Random Excursions Test

<p><b>RANDOM EXCURSIONS TEST</b></p> <p>-----</p> <p><b>COMPUTATIONAL INFORMATION:</b></p> <p>-----</p> <p>(a) Number Of Cycles (J) = 0037</p> <p>(b) Sequence Length (n) = 100000</p> <p>-----</p> <p><b>WARNING: TEST NOT APPLICABLE. THERE ARE AN INSUFFICIENT NUMBER OF CYCLES.</b></p>
---

#### 6.5.15 The Random Excursions Variant Test

According to the section of 6.4.15 the output result of the Random Excursions Variant Test is shown in Table 6.15

Table 6.15: Random Excursions Variant Test

<p>RANDOM EXCURSIONS VARIANT TEST</p> <p>-----</p> <p>COMPUTATIONAL INFORMATION:</p> <p>-----</p> <p>(a) Number Of Cycles (J) = 37</p> <p>(b) Sequence Length (n) = 10000</p> <p>-----</p> <p>WARNING: TEST NOT APPLICABLE. THERE ARE AN INSUFFICIENT NUMBER OF CYCLES.</p>
---

### 6.6 Discussion from the Analysis

An analytical routine has been included to facilitate interpretation of the results. A file **finalAnalysisReport** is generated when statistical testing is complete. The results are represented via a table with  $p$  rows and  $q$  columns. The number of rows,  $p$ , corresponds to the number of statistical tests applied. The number of columns,  $q = 13$ , are distributed as follows: columns 1-10 correspond to the frequency of P-values, column 11 is the P-value that arises via the application of a chi-square test, column 12 is the proportion of binary sequences that passed, and the 13<sup>th</sup> column is the corresponding statistical test.

Table 6.16 illustrates the final analysis report that was obtained after processing 100 binary sequences each consisting of  $10^6$  bits. For each test the level of significance  $\alpha=0.01$ . The results fix up the pseudorandom characteristics of the sequence. The results which are not random marked by an asterisk.

NIST has approved two methods to test the randomness of data which include (1) the test of the proportion of sequences passing a statistical test and (2) the distribution of P-values to check for uniformity. The acceptable proportion of passing sequences should fall within the interval  $0:99 \pm 0:0094392$ . The proportion for the longest run, Non-Overlapping Template and

Approximate Entropy are not in the interval. So the data can be considered as not random and marked with asterisk. While considering uniformity of data the level significance is  $\alpha=0.0001$  and the p-value should be  $P\text{-value} \geq 0.0001$ . Also to provide meaningful results at least 55 sequences must be processed. From the above table the p-value for Rank test, Universal test and Approximate Entropy test are below the standard of P-value. Therefore for these three tests statistics uniformity test fails.

Table 6.16 Depiction of the Final Analysis Report

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES													
generator is <data/data.encryption>													
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST	
2	0	2	1	0	1	3	0	1	0	0.350485	10/10	Frequency	
1	0	1	0	2	1	0	1	2	2	0.739918	10/10	BlockFrequency	
2	0	1	0	3	1	0	0	2	1	0.350485	9/10	CumulativeSums	
2	1	2	1	0	1	1	2	0	0	0.739918	10/10	CumulativeSums	
3	1	1	0	1	1	2	0	1	0	0.534146	9/10	Runs	
4	0	1	1	0	1	1	2	0	0	0.122325	10/10	LongestRun	
1	2	1	3	0	0	1	0	2	0	0.350485	10/10	Rank	
3	0	1	0	1	0	0	2	1	2	0.350485	10/10	FFT	
1	0	0	3	0	0	2	2	1	1	0.350485	10/10	NonOverlappingTemplate	
2	1	2	2	1	0	1	0	1	0	0.739918	10/10	NonOverlappingTemplate	
1	1	1	1	1	1	0	0	1	3	0.739918	10/10	NonOverlappingTemplate	
2	0	1	1	0	0	2	2	1	1	0.739918	10/10	NonOverlappingTemplate	
2	1	1	2	0	1	2	0	0	1	0.739918	9/10	OverlappingTemplate	
0	0	0	0	0	0	10	0	0	0	0.000000	*	Universal	
9	0	1	0	0	0	0	0	0	0	0.000000	*	4/10	* ApproximateEntropy
0	0	0	0	0	0	0	0	0	0	----	-----	RandomExcursions	
0	0	0	0	0	0	0	0	0	0	----	-----	RandomExcursions	
0	0	0	0	0	0	0	0	0	0	----	-----	RandomExcursionsVariant	
0	0	0	0	0	0	0	0	0	0	----	-----	RandomExcursionsVariant	
0	2	0	0	1	1	1	2	1	2	0.739918	10/10	Serial	
0	1	0	3	0	1	1	0	1	3	0.213309	10/10	Serial	
0	1	1	0	1	2	1	0	2	2	0.739918	10/10	LinearComplexity	

The minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately = 8 for a sample size = 10 binary sequences.

The minimum pass rate for the random excursion (variant) test is undefined.

For further guidelines construct a probability table using the MAPLE program provided in the addendum section of the documentation.



## CHAPTER VII

### Conclusions

#### 7.1 Concluding Discussion

The proposed Dynamic DNA cryptographic technique enhances the level of secrecy of data. This technique is applied dynamic sequence table and dynamic DNA encoding over asymmetric cryptosystems. Dynamic sequence table is formed through: at first 256 ASCII characters are assigned to each DNA sequence randomly and then the table is iterated following a mathematical series. Dynamic DNA encoding is formed through: at first sufficient number of random binary strings are generated and then it is converted to DNA bases. To pick the number of DNA bases a mathematical series i.e. Fibonacci series is followed and it is used to merge the chunk of ciphertext. Hence the ciphertext dynamically changes for same public key. Thus the proposed technique enhances the level of secrecy of traditional asymmetric cryptosystems. The mathematical calculation to break the secrecy of ciphertext proves that it is almost impossible for any intruder or attacker to breach the secrecy of the ciphertext. In addition to test the randomness of ciphertext NIST test has been performed. The final analysis report shows that for the 15 test cases the p-value is larger than the threshold value. It has been observed that the success rate for each test is greater than the failure rate. Thereby, it produces random number for ciphertext. Thus the generated ciphertext is random. Thereby the ultimate ciphertext generated by the proposed technique becomes highly robust and secure.

#### 7.2 Future work

The proposed technique is a generalized form of cryptographic technique. The technique can be employed to any form data security such as image, text, audio, and video. This technique can be applied in any cryptographic application in the cloud storage and data distribution platform. Further modification of the technique will apply this in real world security protocol.

## REFERENCES

- [1] Adleman, Leonard M. "Molecular computation of solutions to combinatorial problems." *Science* 266, no. 5187, pp. 1021-1024, 1994.
- [2] X.C. Zhang, "Breaking the NTRU Public-key Cryptosystem Using Self-Assembly of DNA Tilings," *Chinese Journal of Computers*, pp. 2129-2137, 2008.
- [3] M. Hirabayashi, H. Kojima and K. Oiwa, "Design of True Random One-Time Pads in DNA XOR Cryptosystem", F. Peper et al. (Eds.): *IWNC 2009, PICT 2*, pp. 174-183, Springer 2010..
- [4] ZHANG, X.C., NIU, Y., CUI, G.Z. and XU, J., "Breaking the RSA public key cryptosystem using self-assembly of DNA tilings," *Systems Engineering and Electronics*, 5, p.046, 2010.
- [5] Z. Zheng, "Nondeterministic Algorithm for Breaking Diffie-Hellman Key Exchange using Self-Assembly of DNA Tiles," *International Journal of Computers, Communications & Control*, 7, 2012.
- [6] B. Anam, K. Sakib, M. Hossain and K. Dahal, "Review on the Advancements of DNA Cryptography," *arXiv preprint arXiv: 1010.0186*, 2010.
- [7] O. Tornea, and M. E. Borda, "DNA cryptographic algorithms," *Int. Conf. on Advancements of Medicine and Health Care through Technology*," Springer, Berlin, Heidelberg, pp. 223-226, 2009.
- [8] O. Tornea, "Contributions to DNA cryptography: applications to text and image secure transmission", *Diss. Université Nice Sophia Antipolis; Technical University of Cluj-Napoca (Roumanie)*, 2013.
- [9] E. S. Babu, C. N. Raju, and M. HM K. Prasad, "Inspired Pseudo Biotic DNA Based Cryptographic Mechanism Against Adaptive Cryptographic Attacks", *Int. Journal of Network Security*, Vol.18, No.2, pp.291-303, 2016.
- [10] Y. Huang, C. Chang and C. Wu, "A DNA-based data hiding technique with low modification rates", *Multimedia Tools and applications*. Vol. 70, No. 3, pp. 1439-1451, 2014.

- [11] K. S. Kabir, T. Chakraborty, and A.B.M. Alim Al Islam, "SuperCrypt: A Technique for Quantum Cryptography through Simultaneously Improving Both Security Level and Data Rate," Proc. of 2016 Int. Conf. on Networking System and Security, pp. 25-33, 2016
- [12] E. M. S. Hossain, A. K. Md. Rokibul, M. R. Biswas, and Y. Morimoto, "A DNA cryptographic technique based on dynamic DNA sequence table," In 19th Int. Conf. on Computer and Information Technology (ICCIT), pp. 270-275, IEEE, 2016.
- [13] M. R. Biswas,, A. K. Md. Rokibul, A. Akber and Y. Morimoto "A DNA Cryptographic Technique Based on Dynamic DNA Encoding and Asymmetric Cryptosystem," In Networking, Systems and Security (NSysS), 2017 4th International Conference on, pp. 1-8. IEEE, 2017.
- [14] Zhang Y, Liu X, Sun M. DNA based random key generation and management for OTP encryption. Biosystems. pp. 51-63, 2017.
- [15] S. Kalyani and N. Gulati, "Pseudo DNA cryptography technique using OTP key for secure data transfer, " Int. J. Eng. Sci, 6, pp.5657-5663, 2016.
- [16] M. Borda and O. Tornea. "DNA secret writing techniques," In Communications (COMM), 2010 8th International Conference on, pages 451–456, June 2010.
- [17] Aich, A., Sen, A., Dash, S. R., & Dehuri, S. A symmetric key cryptosystem using DNA sequence with OTP key. In Information Systems Design and Intelligent Applications. Springer, New Delhi, pp. 207-215, 2015.
- [18] A. Gehani, T. LaBean, and J. Reif, "DNA-based cryptography," Lecture notes in computer science 2950, pp. 167-188, 2003.
- [19] Khalifa and A. Atito. High-capacity DNA-based steganography. In Informatics and Systems (INFOS), 2012 8th International Conference on, pages BIO–76–BIO– 80, May 2012.
- [20] Gao, Qinghai. "A few DNA-based security techniques." In IEEE Long Island Systems Applications and Technology Conference (LISAT), pp. 1-5. 2011.
- [21] Risca, V.I., "DNA-based steganography," Cryptologia, 25(1), pp.37-49, 2001.
- [22] Malathi, P., Manoj, M., Manoj, R., Raghavan, V., & Vinodhini, R. E. (2017). "Highly Improved DNA Based Steganography," Procedia Computer Science, 115, 651-659.



- [23] Vinodhini, R. E., and P. Malathi. "DNA Based Image Steganography," In Computational Vision and Bio Inspired Computing, Springer, Cham, pp. 819-829, 2018.
- [24] Das, P., & Kar, N. (2014, February). A DNA based image steganography using 2D chaotic map. In Electronics and Communication Systems (ICECS), International Conference on (pp. 1-5). IEEE 2014.
- [25] R. P. K. Reddy, C. Nagaraju and N. Subramanyam, "Text Encryption Through Level Based Privacy Using DNA Steganography, " Int. Journal of Emerging Trends & Technology in Computer Science (IJETTCS). ISSN 2278-6856, Volume 3, Issue 3, May-June 2014.
- [26] G. Cui, C. Li, H. Li, and X. Li "DNA Computing and Its Application to Information Security Field, " Fifth Int. Conf. on Natural Computation 2009.
- [27] Jossy P. George and Joseph Varghese Kureethara, "An efficient 2-Step DNA symmetric cryptography algorithm based on dynamic data structures," International Journal of Engineering & Technology, 7 (2.6) pp.141-146, IJET-2018.
- [28] F. K. A. and D. Antony. A multiphase cryptosystem with secure key encapsulation scheme based on principles of DNA computing. In Advances in Computing and Communications (ICACC), 2014 Fourth International Conference on., pages 1–4, Aug 2014.
- [29] B. Beegom and S. Jose, "An Enhanced Cryptographic Model Based on DNA Approach," Int. Conf. on Electronics, Communication and Aerospace Technology (ICECA), pp.317-322, 2017.
- [30] S. Kalsi, H. Kaur, and V. Chang, "DNA Cryptography and Deep Learning using Genetic Algorithm with NW algorithm for Key Generation," S 42: 17. <https://doi.org/10.1007/s10916-017-0851-z>, Springer US, ISSN: 0148-5598, 2018.
- [31] S. K. Pujari, G. Bhattacharjee and S. Bhoi, "A Hybridized Model for Image Encryption through Genetic Algorithm and DNA Sequence," ELSEVIER, Procedia Computer Science, Volume 125, 2018, pp. 165-171, 2018.
- [32] Saranya MR, Mohan AK, Anusudha K. Algorithm for enhanced image security using DNA and genetic algorithm. In Signal Processing, Informatics, Communication and Energy Systems (SPICES), IEEE International Conference on Feb 19, pp. 1-5. 2015.
- [33] Thomas, M. Cimi, and S. Sheeja. "DNA based Feistel Cipher with Sub keys selected using Genetic Algorithm," pp. 153-158, 2017.

- [34] H. Hammami, H. Brahmi, and S. Ben.Yahia “Secured outsourcing towards a cloud computing environment based on DNA cryptography,” In Information Networking (ICOIN), 2018 International Conference on, pp. 31-36. IEEE, 2018.
- [35] R. K. Kumar and P. Devi. Bharathi, “A Novel Text Encryption Algorithm Using DNA ASCII Table With a Spiral Approach.” International Journal Recent Scientific Research Vol. 9, Issue, 1(J), pp. 23588-23595, January, 2018.
- [36] S.Paul\*, T.Anwar and A.Kumar, “An innovative DNA cryptography technique for secure data transmission” Int. J. Bioinformatics Research and Applications, Vol. 12, No. 3, 2016.
- [37] A. Rukhin, J. Soto, J.Nechvatal, M.Smid, E.Barker, S.Leigh, M.Levenson, M.Vangel, D.Banks, A.Heckert, J.Dray and S.Vo, “ A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications” NIST Special Publication 800-22 Revision 1a Revised: Lawrence E Bassham III2, April 2010.
- [38] Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program. National Institute of Standards and Technology. Initial Release: March 28, 2003, Last Update: May 25, 2018.
- [39] Fasila K.A. and Deepthy Antony, "A Multiphase Cryptosystem with Secure Key Encapsulation Scheme Based on Principles of DNA Computing", Fourth IEEE International Conference on Advances in Computing and Communications, 978-1-4799-4363-0/14, 2014.

## Appendix A

Table A.1 Dynamic Sequence Table

No of items	DNA Bases	ASCII character	No of items	DNA Bases	ASCII character
0	AAAA	NULL	23	AATG	SO
1	AAAC	SOH	24	AATT	SI
2	AAAG	STX	25	ACAA	DLE
3	AAAT	ETX	26	ACAC	DC1
4	AACA	EOT	27	ACAG	DC2
5	AACC	ENQ	28	ACAT	DC3
6	AACG	ACK	29	ACCA	DC4
7	AACT	BEL	30	ACCC	NAK
8	AAGA	BS	31	ACCG	SYN
9	AAGC	HT	32	ACCT	ETB
10	AAGG	LF	33	ACGA	CAN
11	AAGT	VT	34	ACGC	EM
12	AATA	FF	35	ACGG	SUB
13	AATC	CR	36	ACGT	ESC
14	ACTA	FS	37	CCAC	1
15	ACTC	GS	38	CCAG	2
16	ACTG	RS	39	CCAT	3
17	ACTT	US	40	CCCA	4
18	CAAA	Space	41	CCCC	5
19	CAAC	!	42	CCCG	6
20	CAAG	“	43	CCCT	7
21	CAAT	#	44	CCGA	8
22	CACA	\$	45	CCGC	9

No of items	DNA Bases	ASCII character	No of items	DNA Bases	ASCII character
46	CACC	%	75	CCGG	:
47	CACG	&	76	CCGT	;
48	CACT	'	77	CCTA	<
49	CAGA	(	78	CCTC	=
50	CAGC	)	79	CCTG	>
51	CAGG	*	80	CCTT	?
52	CAGT	+	81	GAAA	@
53	CATA	,	82	GAAC	A
54	CATC	-	83	GAAG	B
55	CATG	.	84	GAAT	C
56	CATT	/	85	GACA	D
57	CCAA	0	86	GACC	E
58	GACG	F	87	GCGT	[
59	GACT	G	88	GCTA	\
60	GAGA	H	89	GCTC	]
61	GAGC	I	90	GCTG	^
62	GAGG	J	91	GCTT	_
63	GAGT	K	92	TAAA	'
64	GATA	L	93	TAAC	a
65	GATC	M	94	TAAG	b
66	GATG	N	95	TAAT	c
67	GATT	O	96	TACA	d
68	GCAA	P	97	TACC	e
69	GCAC	Q	98	TACG	f
70	GCAG	R	99	TACT	g
71	GCAT	S	100	TAGA	h
72	GCCA	T	101	TAGC	i
73	GCCC	U	102	TAGG	j
74	GCCG	V	103	TAGT	k

No of items	DNA Bases	ASCII character	No of items	DNA Bases	ASCII character
104	GCCT	W	132	TATA	l
105	GCGA	X	133	TATC	m
106	GCGC	Y	134	TATG	n
107	GCGG	Z	135	TATT	o
108	TCAA	p	136	AGCC	ä
109	TCAC	q	137	AGCG	å
110	TCAG	r	138	AGCT	ç
111	TCAT	s	139	AGGA	ê
112	TCCA	t	140	AGGC	ë
113	TCCC	u	141	AGGG	è
114	TCCG	v	142	AGGT	ï
115	TCCT	w	143	AGTA	î
116	TCGA	x	144	AGTC	ì
117	TCGC	y	145	AGTG	í
118	TCGG	z	146	AGTT	î
119	TCGT	{	147	ATAA	É
120	TCTA		148	ATAC	æ
121	TCTC	}	149	ATAG	Æ
122	TCTG	~	150	ATAT	ô
123	TCTT	DEL	151	ATCA	Ö
124	AGAA	ç	152	ATCC	Ó
125	AGAC	ù	153	ATCG	Û
126	AGAG	é	154	ATCT	Ú
127	AGAT	â	155	ATGA	ÿ
128	AGCA	ä	156	ATGC	Ö
129	ATGG	Û	157	CGTT	»
130	ATGT	ç	158	CTAA	
131	ATTA	£	159	CTAC	

No of items	DNA Bases	ASCII character	No of items	DNA Bases	ASCII character
160	ATTC	∅	189	CTAG	
161	ATTG	×	190	CTAT	
162	ATTT	<i>f</i>	191	CTCA	
163	CGAA	á	192	CTCC	Á
164	CGAC	í	193	CTCG	Î
165	CGAG	ó	194	CTCT	À
166	CGAT	ú	195	CTGA	©
167	CGCA	ñ	196	CTGC	
168	CGCC	Ñ	197	CTGG	
169	CGCG	•	198	CTGT	
170	CGCT	°	199	CTTA	
171	CGGA	ı	200	CTTC	¢
172	CGGC	®	201	CTTG	¥
173	CGGG	¬	202	CTTT	¬
174	CGGT	½	203	GGAA	
175	CGTA	¼	204	GGAC	
176	CGTC	ì	205	GGAG	
177	CGTG	«	206	GGAT	
178	GGCA		207	GTGC	
179	GGCC	†	208	GTGG	
180	GGCG	ã	209	GTGT	
181	GGCT	Ã	210	GTTA	
182	GGGA		211	GTTC	
183	GGGC		212	GTTG	
184	GGGG		213	GTTT	
185	GGGT		214	TGAA	Ó
186	GGTA		215	TGAC	β
187	GGTC		216	TGAG	Ô
188	GGTG		217	TGAT	Ò

No of items	DNA Bases	ASCII character	No of items	DNA Bases	ASCII character
218	GGTT		237	TGCA	
219	GTAA		238	TGCC	
220	GTAC	Đ	239	TGCG	
221	GTAG	Ê	240	TGCT	
222	GTAT	Ë	241	TGGA	
223	GTCA	É	242	TGGC	Ú
224	GTCC		243	TGGG	Û
225	GTCG		244	TGGT	Ü
226	GTCT		245	TGTA	
227	GTGA		246	TGTC	
228	TGTG	-	247	TTCT	
229	TGTT	'	248	TTGA	
230	TTAA		249	TTGC	”
231	TTAC	±	250	TTGG	
232	TTAG		251	TTGT	<sup>1</sup>
233	TTAT		252	TTTA	<sup>3</sup>
234	TTCA	¶	253	TTTC	<sup>2</sup>
235	TTCC	§	254	TTTG	
236	TTCG		255	TTTT	nbsp