

**IMPLEMENTATION OF MAC USING DNA-BERNOULLI MODEL  
AND NOVEL HASH ALGORITHM**

**DISSERTATION-II**

*Submitted in partial fulfillment of the  
Requirement for the award of the  
Degree of*

**MASTER OF TECHNOLOGY IN  
Electronics and Communication Engineering**

*By*

***Gurpreet Kour Sodhi (11503689)***

***Under the Guidance of  
Mr.Gurjot Singh  
Assistant Professor, L.P.U***



**L** LOVELY  
**P** ROFESSIONAL  
**U** NIVERSITY

---

*Transforming Education Transforming India*

**School of Electronics and Communication Engineering  
Lovely Professional University  
Phagwara, Punjab  
April 2017**

**TOPIC APPROVAL PERFORMA**

School of Electronics and Electrical Engineering

**Program :** P175::M.Tech. (Electronics and Communication Engineering) [Full Time]

**COURSE CODE :** ECE521

**REGULAR/BACKLOG :** Regular

**GROUP NUMBER :** EEERGD0023

**Supervisor Name :** Gurjot Singh

**UID :** 17023

**Designation :** Assistant Professor

**Qualification :** \_\_\_\_\_

**Research Experience :** \_\_\_\_\_

SR.NO.	NAME OF STUDENT	REGISTRATION NO	BATCH	SECTION	CONTACT NUMBER
1	Gurpreet Kour Sodhi	11503689	2015	E1514	9023339193

**SPECIALIZATION AREA :** Wireless Communication

**Supervisor Signature:** \_\_\_\_\_

**PROPOSED TOPIC :** Implementation of Message Authentication Code using DNA-Bernoulli model and novel Hash Algorithm

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	Parameter	Rating (out of 10)
1	Project Novelty: Potential of the project to create new knowledge	7.50
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	8.00
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.50
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	8.00
5	Social Applicability: Project work intends to solve a practical problem.	8.00
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	8.00

PAC Committee Members		
PAC Member 1 Name: Rajeev Kumar Patial	UID: 12301	Recommended (Y/N): Yes
PAC Member 2 Name: Lavish Kansal	UID: 15911	Recommended (Y/N): Yes
PAC Member 3 Name: Dr. Gursharanjeet Singh	UID: 13586	Recommended (Y/N): NA
DAA Nominee Name: Amanjot Singh	UID: 15848	Recommended (Y/N): NA

**Final Topic Approved by PAC:** Implementation of Message Authentication Code using DNA-Bernoulli model and novel Hash Algorithm

**Overall Remarks:** Approved

**PAC CHAIRPERSON Name:** 11106::Dr. Gaurav Sethi

**Approval Date:** 05 Oct 2016



## **CERTIFICATE**

This is to certify that Gurpreet Kour Sodhi bearing Registration no. 11503689 has accomplished the objectives of the thesis titled, “**Implementation of MAC using DNA-Bernoulli Model and novel Hash Algorithm**” under my guidance and supervision. To the best of my knowledge, the present work is the result of her original investigation and study. No part of thesis has ever been submitted for any other degree at any university.

**Mr. Gurjot Singh**  
**Assistant Professor**  
**School of Electronics and Communication**  
Lovely Professional University  
Phagwara, Punjab

Date:

## **DECLARATION**

I, Gurpreet Kour Sodhi, student of M.Tech under Department of Electronics and Communication of Lovely Professional University, Punjab, hereby declare that all the information furnished in this Dissertation-II report is based on my own intensive research and is genuine.

This report does not, to the best of my knowledge, contain part of my work which has been submitted for the award of my degree either of this University or any other University without proper citation.

**Gurpreet Kour Sodhi**

**11503689**

## **ACKNOWLEDGEMENT**

First and foremost, I would like to express my sincere gratitude and appreciation to my guide Mr. Gurjot Singh, for his whole-hearted and invaluable guidance, inspiring discussions, encouragement, and support throughout my work. I found him always sincere in helping me even during his busiest hours of the day. His ardor and earnestness for studies are respected and will never be forgotten. Without his sustained and sincere effort, this report would not have taken this shape.

We are also indebted to all authors of the research papers and books referred to, which have helped us in carrying out the research work.

**Gurpreet Kour Sodhi**  
**Reg. No: 11503689**

## **ABSTRACT**

Communication refers to imparting or exchanging of information between two or more parties using the data links. With the advancement in the field of electronic commerce and internet, the major area of concern in the communication sector, is security. Communication security is the discipline of preventing unauthorized interceptors from accessing the telecommunication in an intelligible form, while still delivering content to the intended recipients. The proposed technique aims to implement a MAC scheme using security key generated from the fusion of biological characteristics of a human body and the random number generator sequences and a novel hash algorithm. It is observed that, this not only ensures different keys for different individuals but also makes it difficult to duplicate the key and since the complexity increases the security is enhanced as well. The presented approach finds its application in various data sensitive environments, as it ensures trustworthy and reliable communication. The presented technique is tested using NIST tests and uniqueness is verified using Avalanche criteria. In communication networks, the most important concern is to verify the integrity and authenticity of the data received; therefore the combination of biometrics and mathematical functions, when used to formulate MAC, prevents the alteration of original message content by the adversaries.

## LIST OF ABBREVIATIONS

BBSG	Blum Blum Shub Generator
BRNG	Bernoulli Random Number Generator
DNA	DeoxyriboNucleic Acid
ECG	Electrocardiogram Random Number Generator
FIPS	Federal information processing standard
HMAC	Hash based Message Authentication Code
LCG	Linear Congruential Generator
MAC	Message Authentication Code
MD	Message Digest
NIST	National Institute of Standards and Technology
RNA	Ribonucleic acid
SHA	Secure Hash Algorithm
TCP/IP	Transmission Control Protocol/Internet Protocol



## TABLE OF CONTENTS

<b>Title Page</b>	<b>Page Number</b>
<b>PAC</b>	<b>i</b>
<b>CERTIFICATE</b>	<b>ii</b>
<b>DECLARATION</b>	<b>iii</b>
<b>ACKNOWLEDGEMENT</b>	<b>iv</b>
<b>ABSTRACT</b>	<b>v</b>
<b>LIST OF ABBREVIATIONS</b>	<b>vi</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>LIST OF TABLES</b>	<b>x</b>
<b>LIST OF EQUATIONS</b>	<b>xi</b>
<b>CHAPTER 1: INTRODUCTION</b>	<b>1-16</b>
1.1 Communication Networks	1
1.1.1 Challenges in Communication Networks	1
1.1.2 Applications of Communication Networks	2
1.2 Need of Security	2
1.3 Model for Network Security	3
1.4 Security Measures	5
1.5 Significance of MAC	6
1.6 SHA-160	7
1.7 DNA and Random Number Generators	10
1.7.1 Characteristics of DNA	10
1.7.2 Random Number Generators	12
1.8 Effectiveness of Data Integrity	14
<b>CHAPTER 2: REVIEW OF LITERATURE</b>	<b>17-21</b>
<b>CHAPTER 3: PROBLEM FORMULATION</b>	<b>22</b>
<b>CHAPTER 4: OBJECTIVES</b>	<b>23</b>
<b>CHAPTER 5: RESEARCH METHODOLOGY</b>	<b>24-40</b>
5.1 <b>Phase 1: Security Key Generation</b>	25
5.1.1 DNA Sequence Formulation	25
5.2 Generation of RNG Sequences	26
5.2.1 BRNG Sequence Generation	26
5.2.2 BBSG Sequence Generation	27
5.2.3 LCG Sequence Generation	28
5.3 Fusion of DNA and RNG Sequences	28

5.3.1 Fusion of DNA and BRNG Sequences	28
5.3.2 Fusion of DNA and BBSG Sequences	30
5.3.3 Fusion of DNA and LCG Sequences	32
5.4 <b>Phase 2:</b> Generation of a Novel Hash Algorithm	34
5.4.1 Integration of ‘f’ function	35
5.5 <b>Phase 3:</b> Formation of MAC	37
5.5.1 Formation of MAC using DNA-BRNG and Novel Hash Algorithm	37
5.5.2 Formation of MAC using DNA-BBSG and Novel Hash Algorithm	39
5.5.3 Formation of MAC using DNA-LCG and Novel Hash Algorithm	40
<b>CHAPTER 6: RESULTS AND DISCUSSIONS</b>	<b>41-62</b>
6.1 Analysis using NIST Tests	41
6.1.1 NIST Test Analysis of Key Generated using DNA and BRNG	43
6.1.2 NIST Test Analysis of Key Generated using DNA and BBSG	43
6.1.3 NIST Test Analysis of Key Generated using DNA and LCG	44
6.1.4 NIST Test Analysis of Novel Hash Algorithm	45
6.1.5 NIST Test Analysis of MAC formed using DNA-BRNG key and Novel Hash	46
6.1.6 NIST Test Analysis of MAC formed using DNA-BBSG key and novel Hash	48
6.1.7 NIST Test Analysis of MAC formed using DNA-LCG key and novel Hash	50
6.2 Avalanche Test Analysis	52
6.2.1 Avalanche Test Analysis for Key Generated using DNA and BRNG	53
6.2.2 Avalanche Test Analysis for Key Generated using DNA and BBSG	55
6.2.3 Avalanche Test Analysis for Key Generated using DNA and LCG	57
6.2.4 Avalanche Test Analysis for Novel Hash Algorithm	59
6.2.5 Avalanche Test Analysis for DNA-BRNG based MAC	59
6.2.6 Avalanche Test Analysis for DNA-BBSG based MAC	60
6.2.7 Avalanche Test Analysis for DNA-LCG based MAC	60
6.3 Network Attack analysis	61
6.3.1 Network Attack analysis for DNA and BRNG MAC	61
6.3.2 Network Attack analysis for DNA and BBSG MAC	61
6.3.3 Network Attack analysis for DNA and LCG MAC	62
<b>CHAPTER 7: CONCLUSION AND FUTURE SCOPE</b>	<b>63</b>
<b>REFERENCES</b>	<b>64</b>
<b>APPENDIX-1</b>	<b>66</b>
S-Boxes used in the Proposed Hash Technique	
<b>APPENDIX-2</b>	<b>67</b>
Proposed Work Plan with Timeline	
<b>APPENDIX-3</b>	<b>68</b>
Autobiography	
<b>APPENDIX-4</b>	<b>69</b>
Publications	

## LIST OF FIGURES

<b>Figure</b>	<b>Caption</b>	<b>Page No.</b>
Figure 1.1	Security requirements Triad	3
Figure 1.2	Model for Network Security	3
Figure 1.3	SHA-160 Compression function	8
Figure 1.4	Chemical Structure of DNA	11
Figure 1.5	Location of Nuclear DNA within the Chromosome	11
Figure 5.1	Security Key Formation	24
Figure 5.2	Proposed Research Methodology	25
Figure 5.3	Key Generation Process for DNA and BRNG	29
Figure 5.4	Key Generation Process for DNA and BBSG	31
Figure 5.5	Key Generation Process for DNA and LCG	33
Figure 5.6	SHA-160 Compression function	34
Figure 5.7	Structure of Proposed Hash Algorithm	36
Figure 5.8	Operations applied on DNA-BRNG Key	38

## LIST OF TABLES

<b>Table No.</b>	<b>Caption</b>	<b>Page No.</b>
Table 1.1	Buffer values for SHA-160	8
Table 5.1	DNA Sequence Formation	26
Table 5.2	BRNG Produced Random Sequences	27
Table 5.3	BBSG Produced Random Sequences	27
Table 5.4	LCG Produced Random Sequences	28
Table 5.5	Security Keys Generated using DNA and BRNG	30
Table 5.6	Security Keys Generated using DNA and BBSG	32
Table 5.7	Security Keys Generated using DNA and LCG	34
Table 5.8	Message Digest Values for Different inputs	37
Table 5.9	Security Keys using DNA-BRNG for SHA160	38
Table 5.10	MAC values for Novel Hash and DNA-BRNG Keys	38
Table 5.11	Security Keys using DNA-BBSG for SHA-160	39
Table 5.12	MAC values for Novel Hash and DNA-BBSG Keys	39
Table 5.13	Pseudo Code for operation on DNA-LCG Keys	40
Table 5.14	Security Key using DNA-LCG Keys	40
Table 5.15	MAC values for Novel Hash and DNA-LCG Keys	40
Table 6.1	NIST Test Results for DNA and BRNG	43
Table 6.2	NIST Test Results for DNA and BBSG	44
Table 6.3	NIST Test Results for DNA and LCG	44
Table 6.4	Frequency Test Result for Hash Algorithm	45
Table 6.5	Binary Derivative Test result for Hash algorithm	45
Table 6.6	DFT Test Result for Hash Algorithm	45
Table 6.7	Approximate Entropy test result for Hash algorithm	46
Table 6.8	Maurer Test analysis for Hash Algorithm	46
Table 6.9	Frequency Test result for DNA-BRNG based MAC	47
Table 6.10	Binary Derivative Test result for DNA-BRNG based MAC	47
Table 6.11	DFT Test results for DNA-BRNG based MAC	47
Table 6.12	Approximate Entropy Test for DNA-BRNG based MAC	48
Table 6.13	Maurer Test result for DNA-BRNG based MAC	48
Table 6.14	Frequency Test result for DNA-BBSG based MAC	49
Table 6.15	Binary Derivative Test result for DNA-BBSG based MAC	49
Table 6.16	DFT Test result for DNA-BBSG based MAC	49
Table 6.17	Approximate Entropy Test for DNA-BBSG based MAC	50
Table 6.18	Maurer Test result for DNA-BBSG based MAC	50

Table 6.19	Frequency Test result for DNA-LCG based MAC	51
Table 6.20	Binary Derivative Test result for DNA-LCG based MAC	51
Table 6.21	DFT Test result for DNA-LCG based MAC	51
Table 6.22	Approximate Entropy test for DNA-LCG based MAC	52
Table 6.23	Maurer Test result for DNA-LCG based MAC	52
Table 6.24	Avalanche Test Analysis: Case 1(DNA-BRNG)	53
Table 6.25	Avalanche Test Analysis: Case 2(DNA-BRNG)	54
Table 6.26	Avalanche Test Analysis: Case 3(DNA-BRNG)	54
Table 6.27	Avalanche Test Analysis: Case 1(DNA-BBSG)	55
Table 6.28	Avalanche Test Analysis: Case 2(DNA-BBSG)	56
Table 6.29	Avalanche Test Analysis: Case 3(DNA-BBSG)	56
Table 6.30	Avalanche Test Analysis: Case 1(DNA-LCG)	57
Table 6.31	Avalanche Test Analysis: Case 2(DNA-LCG)	58
Table 6.32	Avalanche Test Analysis: Case 3(DNA-LCG)	58
Table 6.33	Avalanche Test Analysis For Novel Hash	59
Table 6.34	Avalanche Test Analysis for DNA-BRNG based MAC	60
Table 6.35	Avalanche Test Analysis for DNA-BBSG based MAC	60
Table 6.36	Avalanche Test Analysis for DNA-LCG based MAC	61
Table 6.37	Resistance of DNA-BRNG based MAC against attack	61
Table 6.38	Resistance of DNA-BBSG based MAC against attack	61
Table 6.39	Resistance of DNA-KCG based MAC against attack	62

## LIST OF EQUATIONS

Equation No.	Equation Name	Page No.
1	Hash Function	6
2	MAC Algorithm	7
3	SHA-160 Processing Steps	9
4	Computation of 'f' Function	9
5	K Constant Values	9
6	Operations on Buffer Values	9
7	Outputs Transformation Step	10
8	BRNG Formula	13
9	BBSG Formula	14
10	LCG Formula	14
11	Expansion and Substitution In 'f' Function	35
12	P-value of Random Excursion Variant Test	42
13	Avalanche Effect	52
14	Throughput Formula	59

# CHAPTER 1

## INTRODUCTION

---

Communication (from the Latin word *commūnicāre*, meaning “to share”) is an act of conveying intended messages from one entity or group to another through the use of mutually understood signs and rules.

### 1.1 Communication Networks

A series of points or nodes interconnected by communication paths, which exchange data among themselves form a network. The communication paths can be wired or wireless links and the process is governed by certain protocols which are confined to a particular network. The best known communication network is the internet.

Providing access to information on shared storage devices is an important feature of networks. A network is referred to as an efficient network if the data is exchanged reliably, which further means that the data is received by the intended receiver within the desired time span retaining its authenticity [1].

#### 1.1.1 Challenges in Communication Networks

The communication sector faces many challenges, which as summarized as:

- 1) **Cost:** The upgrades to network communications infrastructure mandated by regulatory standards or selected for competitive reasons may incur unnecessarily high costs unless implemented efficiently.
- 2) **Standards and regulation:** This includes developing a framework for interoperability of smart grid devices and systems.
- 3) **Security:** The primary task of utilities is to maintain high levels of security. Any enhancement to this infrastructure must therefore be made with reliability in mind. The security here is a broad term which includes integrity, confidentiality and authentication.

- 4) **Maintenance and Repair Processes:** A variety of factors complicate maintenance and repair of utility communication networks. These factors include diverse topography and geography, population density, local regulations and environmental concerns.
- 5) **Equipment portability:** Wired technology is not portable. Network configuration may limit the options for employee and equipment placement. Additional network cabling, installing new electrical outlets and configuring network structures are other issues.
- 6) **Power:** The equipments must have power supply for their operations. There are various issues related to power such as weather conditions.

### 1.1.2 Application of Communication Networks

Communication networks are of two types depending upon the type of links between the devices: wired or wireless. Communication Networks find their application in various areas. Applications of wireless communication involve security systems, television remote control, Wi-Fi, mobile communication, microwave communication, infra red communication, wireless power transfer, computer interface devices, satellite communication and radio broadcasting. Wired communication includes optical fiber communication, Local Area Network, Ethernet etc.

## 1.2 Need for Security

Security of the data becomes most important concern in order to avoid an unauthorized or unintended access. It includes information security to prevent any sort of data theft. There are mainly three objectives in security provisioning:

- 1) **Confidentiality:** It assures that the private information transmitted by the source is not disclosed to any unauthorized individuals. It also ensures privacy that the specific user will influence or collect only the data which is related to him.
- 2) **Integrity:** It assures that the information at the receiver end is changed only according to the authorized access. It ensures that the data has not been altered; the final output at the receiver is changed according to the specified algorithm or by an authorized party.



3) **Availability:** It assures that the system is working properly and service to the authorized user has not been denied.

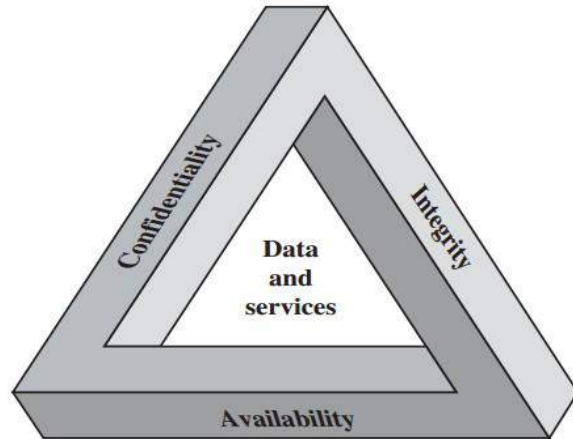


Figure 1.1: Security Requirements Triad

### 1.3 Model for Network Security

A message is transmitted by a source on a wired or wireless channel towards the destination. Routing process has been taken care of by using certain routing protocols depending on the constraints with respect to the environment in which the network has been established (e.g. TCP/IP). Security comes into role when it becomes necessary to protect data from the intruders. This can be done by various methods such as by encrypting a message or by adding additional code into the original message or by assigning different secret keys to both sender and receiver. By doing this attacker or any other third party is unable to decrypt the original message content [2].

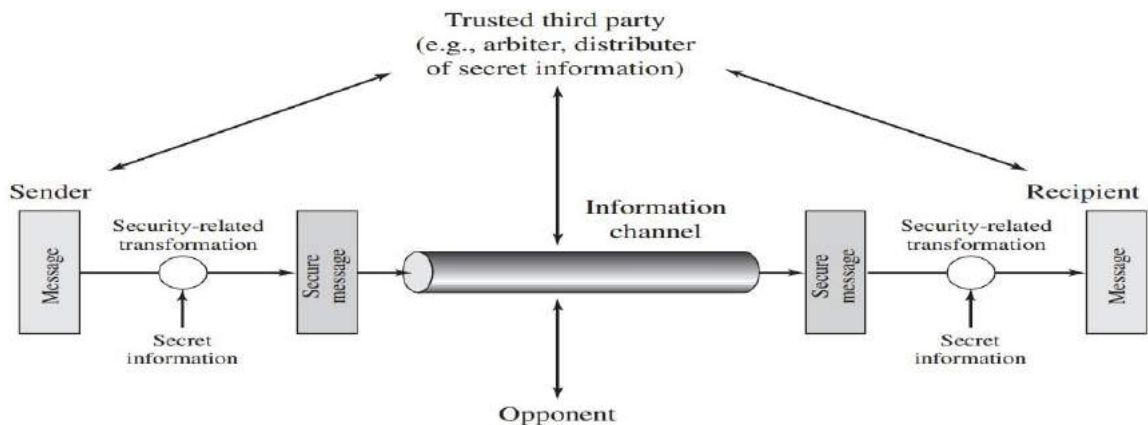


Figure 1.2: Model for Network Security

This model performs following tasks:

- (i) Design such algorithm for security related transformation, which is complex, so that the intruder is unable to break it.
- (ii) Generate some secret information and also add it to the original message.
- (iii) Develop efficient methods for sharing and distribution of secret data.

Security against various attacks is required to ensure reliability. Attacks are classified into two categories:

1) **Passive Attacks:**

In passive attacks, an opponent or an unauthorized member can monitor and listen to the communication between authorized members, passive attacks are basically attacks against the privacy of the data, and it does not affect the system resources. This monitoring and listening to the transmissions by opponent can cause insecurity to data or even entire network, because by continuously monitoring opponent can easily examine many other factors like topology of network and position of the authorized members[2]. Different types of passive attacks are as follows:

**Monitor and Eavesdropping:** In this type of attack the opponent continuously investigates the communication occurring between various members of the given network, opponent can easily find out data contents, by secretly listening to the conversation going on between various authorized members, opponent can easily get much information about the entire network, and it can effect privacy protection [2].

**Traffic Analysis:** When data is transfer from one member to another, it leaves many information for opponent (even if it is in encrypted form), because when communication occurs opponent can easily analyze the communication patterns, and this information is sufficient for an opponent to cause harm to the entire network [2].

2) **Active Attacks:**

In case of active attacks, an opponent or an unauthorized member can monitor and listen to the communication between different authorized members. Active attacks modify the data stream in

communication channel and they can also affect system resources. Different types of active attacks are as follows:

**Masquerade:** It includes insertion of message into an given network from a false or fraudulent source, this insertion of message behaves as it comes from an authorized member. Masquerade basically occur when one specific entity start behaving as other different entity [1].

**Modification of Message:** It includes the alteration in the original message contents by an unauthorized access of opponents. Sometimes message can be delayed also to produce an unauthorized effect [1].

**Replay:** It includes the passive capturing of a data unit and then its subsequent retransmission in order to waste the system resources.

**Denial of Service:** It may lead to the disruption of an entire network due to the unavailability of system resources, or by disabling the network, or by overloading the network with messages so as to degrade its performance [3].

## 1.4 Security Measures

In order to prevent entire network system from above discussed attacks, various techniques are available such as:

- **Data Integrity:** It provides protection to original message content from being modified by any unauthorized party. It ensures us that data has not been altered; the final output information received at the destination is changed according to the specified algorithm or by an authorized party only.

- **Data Confidentiality:** It assures that the information transmitted on a wired or wireless channel has not been disclosed to any unauthorized user. A basic key component for attain confidentiality would be encryption using public key and decryption using private key.

- **Encryption:** It assures for the secure communication, because by applying this technique in the network, only the authorized user or a user having a right secret key can be able to read the original message content. There are different types of encryption schemes which exist such as

Public key encryption and Symmetric key encryption. Encryption helps to achieve data confidentiality among neighbors.

- **Authentication:** It refers to a process of actually confirming about an authorized user. It is a special case of data integrity, it assures that the entity to which an authorized user wants to interact or start communication with, is also an authorized user.

- **Availability:** It assures that the system resources are sufficient for the effective communication and service to an authorized user has not been denied throughout the entire life cycle of the network [2, 4].

## 1.5 Significance of MAC

In cryptography, a Message Authentication Code (MAC) is a short piece of information used to authenticate a message; in other words, to confirm that the message came from the stated sender (its authenticity) and has not been changed in transit (its integrity).

A MAC algorithm, sometimes called a keyed (cryptographic) hash function, accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC. The MAC value protects both a message's data integrity as well as its authenticity, by allowing verifiers (who also possess the secret key) to detect any changes to the message content.

**Hash Function:** In a Hash Function, a message of variable length maps into a fix length hashes value, or Message Digest. All the Hashing Algorithms involve iterative use of compression function. Hash Function 'H' accepts message 'M' of variable length as input and produces a fixed size hash value, as shown in Equation 1.

$$h = H(M) \quad (1)$$

A good hash function has a property of converting large set of inputs to fixed length of evenly distributed and apparently random output [2].

A keyed-Hash Message Authentication Code (HMAC) is a specific type of message authentication code (MAC) involving a cryptographic hash function (hence the 'H') in combination with a secret cryptographic key.

As with any MAC, it may be used to simultaneously verify both the data integrity and the authentication of a message. Any cryptographic hash function, such as MD5 or SHA-1, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC-MD5 or HMAC-SHA1 accordingly. The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output, also on the size and quality of the key [2].

MAC algorithm takes a message and a secret key as input and produces an authentication code as shown in Equation 2.

$$MAC = C(K, M) \quad (2)$$

where,

*M*: Input Message

*C*: MAC Function

*K*: Shared Secret Key

*MAC*: Message Authentication Code

The receiver is in possession of the secret key and it thus generates the authentication code to verify the integrity of the received message. One way of framing a MAC is to combine a cryptographic hash function with a secret key as an input.

## 1.6 SHA-160

The SHA-160 belongs to the family of SHA (Secure Hash Algorithm) cryptographic hash functions. It was proposed by the U.S National Security Agency in 1995 as an U.S Federal information processing standard (FIPS), which is published by the National Institute of Standards and Technology (NIST).It takes an input of arbitrary length and then produce 160-bit message digest. The processing consists of 80 rounds.

### **Pre-Processing Step:**

It consists of three steps: Padding the message, dividing of message, and initialization of hash values.

**Step 1:** Hash algorithms have a constraint of input length. Therefore padding has been done in

order to ensure that padded message is multiple of 512. Padding is done by appending single ‘1’ bit followed by ‘0’ bits till the length of bits in the message becomes congruent to 448 modulo 512.

**Step 2:** The message is divided into blocks of specified length. In the recommended technique, a block of 512 bits is used. Then entire message is divided into blocks of 512 bits as seen in figure 5.1, and each 512 bit block further divided into 16 blocks of 32 bits each.

**Step 3:** Before further processing, five 32 bit words registers A, B, C, D and E are initialized with following values mentioned in Table 1.1.

Table 1.1: Buffer Values for SHA-160

Registers	32-bit Words (hexadecimal)
A	67452301
B	efcdab89
C	98badcfe
D	10325476
E	c3d2e1f0

**Step 4:** Save the values in different variables like,  $A_0 = A$ ,  $B_0 = B$ ,  $C_0 = C$ ,  $D_0 = D$  and  $E_0 = E$

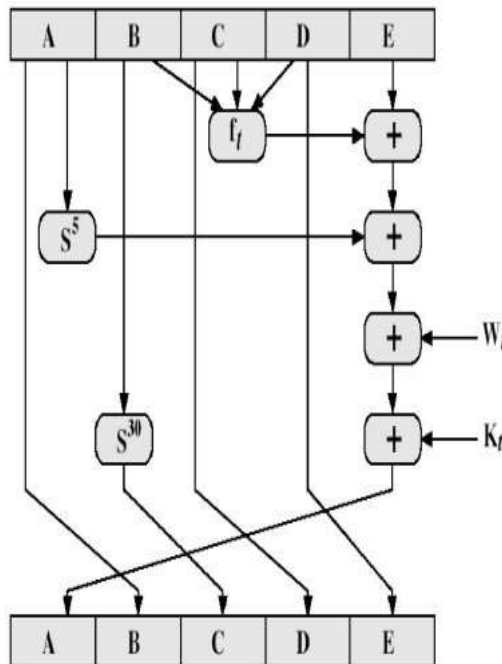


Figure 1.3: SHA-160 Compression Function

**Processing Step:**

(i) Each 512-bit block is further divided into sixteen 32-bit blocks and these sixteen blocks are further expanded to eighty 32-bit blocks by using various mixing and shifting operations as shown in Equation 3

$$\left[ \begin{array}{c} \text{for } i = 17:80 \\ W(t) = W(t-3) \text{ Xor } W(t-8) \text{ Xor } W(t-14) \text{ Xor } W(t-16) \ll 1 \\ \text{end} \end{array} \right] \quad (3)$$

(ii) Use 4 rounds of 20 bit operations on message block & buffer.

(iii) Computation of ‘ $f_t$ ’ function, where  $f_t(b,c,d)$  is a different nonlinear function in each round:

$$\left[ \begin{array}{c} \text{for } i = 1 \text{ to } 20 \Rightarrow f(B,C,D) = (B \text{ and } C) \text{ or } (\text{not}(B) \text{ and } D) \\ \text{for } i = 21 \text{ to } 40 \Rightarrow f(B,C,D) = B \text{ Xor } C \text{ Xor } D \\ \text{for } i = 41 \text{ to } 60 \Rightarrow f(B,C,D) = (B \text{ and } C) \text{ or } (B \text{ and } D) \text{ or } (C \text{ and } D) \\ \text{for } i = 61 \text{ to } 80 \Rightarrow f(B,C,D) = B \text{ Xor } C \text{ Xor } D \end{array} \right] \quad (4)$$

(iv)  $W_t$  is derived from message blocks.

(v)  $K_t$  is a constant value derived from the Sin function as follows:

$$\left[ \begin{array}{c} \text{for } i = 1 \text{ to } 20 \Rightarrow K = 5A827999 \\ \text{for } i = 21 \text{ to } 40 \Rightarrow K = 6ED9EBA1 \\ \text{for } i = 41 \text{ to } 60 \Rightarrow K = 8F1BBCDC \\ \text{for } i = 61 \text{ to } 80 \Rightarrow K = CA62C1D6 \end{array} \right] \quad (5)$$

(vi) Then at last we perform following operations:

$$\left[ \begin{array}{c} \text{Temp} = E + f(b,c,d) + (A \ll 5) + Wt + Kt \\ E = D, D = C, C = (B \ll 30), B = A, A = \text{Temp} \end{array} \right] \quad (6)$$

(vii) Here ‘+’ indicates  $2^{32}$  modulo addition and ‘ $\ll$ ’ indicates left shift.

**Output Transformation Step:**

- The word registers are updated after execution of  $2^{32}$  modulo addition operation between the initial values with final output value of word register as shown in Equation 7. After the generation of preliminary message digest, next 512 bit block of message and updated value of

all the four words register acts as a next input for compression function. The message digest of the complete message is obtained after the processing of the last block of the input message.

$$\left[ \begin{array}{l} A = \text{mod}((A_0 + A), 4294967296) \\ B = \text{mod}((B_0 + B), 4294967296) \\ C = \text{mod}((C_0 + C), 4294967296) \\ D = \text{mod}((D_0 + D), 4294967296) \\ E = \text{mod}((E_0 + E), 4294967296) \end{array} \right] \quad (7)$$

## 1.7 DNA and Random Number Generators

The proposed security system involves the fusion of DNA and Random Number Generator (RNG) outputs. This fusion ensures an efficient security system which produces a security key that is unique and cannot be duplicated, thus can be used to prevent attacks to confidential data.

### 1.7.1 Characteristics of DNA

The measurement and analysis of a human's physical and behavioral characteristics is known as biometrics. It involves the study of those features of a human body which are unique to an individual. Now-a-days the threat to security especially to integrity and authentication is increasing at an alarming rate, due to e-commerce systems or online transactions which involve the personal official details of an individual which by no means should be shared with anyone. Now the today's online scenario involves the use of these details directly or indirectly by the interface thus causing a threat to the person's authenticity. So, here is the case when methods are needed which are purely related to a person and are not susceptible to being stolen or used by false means.

Biometric features include finger prints, retina scan, iris pattern recognizing, voice matching, facial characteristics, images and many other features which are totally unique and possessed by an individual. Leading all the features in the list is DNA (Deoxyribonucleic acid) it is a molecule that carries the genetic instructions.

DNA and RNA are nucleic acids that are essential for life. Most DNA molecules consist of two biopolymer strands which are coiled around each other forming a double helix. The two DNA strands are termed polynucleotides as they are made of simpler monomer units called nucleotides.



Each nucleotide is composed of one of four nitrogen-containing nucleobases either cytosine (C), guanine (G), adenine (A), or thymine (T)—and a sugar called deoxyribose and a phosphate group [7]. The nitrogenous bases of the two separate polynucleotide strands are bound together (according to base pairing rules (A with T, and C with G) with hydrogen bonds to make double-stranded DNA. It is the sequence of these four nucleobases along the backbone which is encoding the biological information [6].

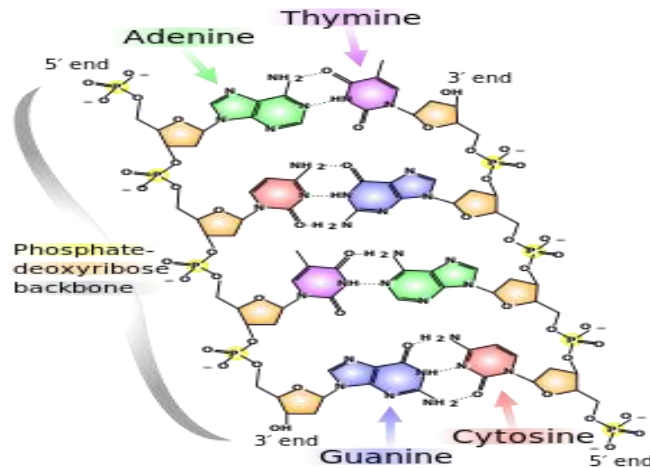


Figure 1.4: Chemical structure of DNA

DNA is a long polymer made from repeating units called nucleotides. DNA polymers can be very large molecules containing millions of nucleotides. The two types of base pairs form different numbers of hydrogen bonds, AT forming two hydrogen bonds, and GC forming three hydrogen bonds. Many DNA sequences of various organisms have been successfully sequenced with higher accuracy [6]. Analyzing DNA sequences investigates the biological relationships of different species. Many distributed databases have been constructed and can be easily accessed from the World Wide Web [8, 9].

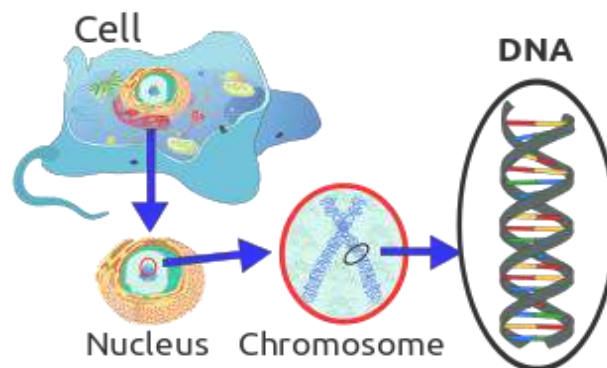


Figure 1.5: Location of nuclear DNA within the chromosomes

DNA is found in hair, blood, spit, saliva etc. As in the current state of the world, every human's genome is about 0.1% different from every other human's genome, this translates to a few million differences. Thus it is the most unique feature of an individual and something that is possessed by an individual and in no means can be replicated.

### **1.7.2 Random number generators**

A random number generator is a system which when provided with an input sometimes known as a seed value, generates a random sequence i.e. a sequence of identically distributed random variables that follow no specific pattern. The input if kept confidential can ensure that the sequence generated will be random and confidential. A security key is said to be efficient if it is random and unique therefore the random number generated outputs can be used as security keys.

Random numbers can be used for cryptography [10]. Random number generators are of two types:

- (i) **True random number generators:** These involve natural phenomena like tossing of coin. Therefore cannot be reproduced
- (ii) **Pseudo random number generators:** These generate random numbers using mathematical functions or algorithms and therefore are reproducible [10].

If the generate key is weak, the whole system is affected. Well designed key in well designed algorithm can ensure better security and confidentiality. The various types of inputs provided to a RNG can be an image source, voice data or a numerical input known as seed. The more the randomness of the key the more secure it is.

Further to strengthen the bond of security a random key is generated by three different random number generators respectively. These Random number generators are explained as:

#### **1 ) Bernoulli Random Number Generator (BRNG)**

Bernoulli numbers have a prominent place in mathematics, for instance they appear in Taylor expansion of tangent and in Euler-Maclaurin formula. Equation 8 represents the BRNG method for generation of n random numbers:

$$\sum_{n=0}^{\infty} \frac{B_n}{n!} x^n = \frac{x}{e^x - 1} \quad (8)$$

The fusion of BRNG and DNA sequence forms a very strong 256 bit key which is less susceptible to attacks and thus provides higher level of security.

### **Parameters used for BRNG Sequence generation:**

*Probability of a zero:* The probability with which a zero output occurs. Specify the probability as a scalar or row vector whose elements are real numbers between 0 and 1. The number of elements in the Probability of a zero parameter correspond to the number of independent channels output from the block.

*Source of initial seed:* The source of the initial seed for the random number generator. Specify the source as either Auto or Parameter. When set to Auto, the block uses the global random number stream.

*Initial seed:* The initial seed value for the random number generator. Specify the seed as a nonnegative integer scalar. Initial seed is available when the Source of initial seed parameter is set to Parameter.

*Sample time:* The time between each sample of a column of the output signal.

*Samples per frame:* The number of samples per frame in one channel of output signal. Specify samples per frame as a positive integer scalar.

*Output data type:* The output type of the block can be specified as Boolean, uint8, uint16, uint32, single, or double. The default is double. Single outputs may lead to different results when compared with double outputs for the same set of parameters.

## **2) Blum Blum Shub Generator (BBSG)**

This is the second random number generator used which also gives a random key based on the input seed value. This random number works on the algorithm explained in Equation 9. This pseudo random generator takes the form:

$$\text{Random number} = \text{mod}((X_o)^2, M) \quad (9)$$

Where,

$M$ : it's the product of two large primes 'p' and 'q'

$X_o$ : is the seed value

The seed should be an integer that is co-prime to  $M$  (i.e.  $p$  and  $q$  are not factors of seed and not equal to 0 or 1. The fusion of BRNG and DNA sequence forms a very strong 256 bit key which is less susceptible to attacks and thus provides higher level of security.

### 3) Linear Congruential Generator (LCG)

Further to strengthen the bond of security a random sequence is generated by LCG, this sequence is generated using a seed value which is kept secret by the user. LCG uses an algorithm that produces a sequence of pseudo-randomized numbers calculated through a linear equation. Its a robust and efficient method of generating pseudo-random numbers..

The generation of output for LCG can be analyzed by the Equation. 10:

$$X_{n+1} = (aX_n + c) \text{ mod } m \quad (10)$$

Where,

$X$ : the sequence of pseudorandom values

$m$ :  $0 < m$  the modulus

$a$ :  $0 < a < m$ , the multiplier

$c$ :  $0 \leq c < m$ , the increment

$X_n$ :  $0 < X_n < m$ , is the seed or start value

This sequence along with the DNA sequence forms a very strong 256 bit key which is not only less susceptible to attacks but also provides a higher level of security.

## 1.8 Effectiveness of Data Integrity

In order to maintain data integrity or to maintain the secure transmission of information in hostile environment different security measures such as authentication, encryption, decryption, etc processes are used, where encryption and decryption provides confidentiality and it hides or

protects the original message contents from adversaries, whereas authentication verifies that weather the sender is an authorized member or not and by combining both of the above schemes we have attained data integrity. For maintaining data integrity different Hash Algorithms, Message Authentication Codes (MAC), Digital Signatures and various Encryption techniques have been used.

Integrity basically refers to a Message Authentication. Authentication is of two types:

**Source Authentication:** It assures the receiver that the sender node is an authorized member of a network.

**Message Authentication:** It assures that data received at the receiver node has been fresh and has not been altered by any intruder.

Effectiveness of data integrity can be expressed as:

- Data Integrity refers to the consistency and accuracy of information transmitted over an unfriendly environment over its entire life cycle. It ensures that data is recorded exactly as intended. It prevents unintentional changes to the original message content.
- Any unintended change to data as a result of retrieval or processing operation, human error, unexpected hardware failure, including malicious node and storage will lead to failure of data integrity.
- It ensures that the data received by the cluster head or any other sensor node does not alter, or it has not been tampered or harmed by any attacker node. The integrity of the network is very important in communication because sometime an attacker may add false node which starts generating false data, then in that case it plays very important role.
- Data Integrity can be implemented by using different types of hash functions or hashing algorithms
- In order to ensure data integrity, various hashing algorithms are available. Hash function simply maps an input of arbitrary length to a fixed output by using a noninvertible compression function. Hash functions are very hard to reverse, due to which they have been used for providing security services such as data integrity, origin authentication. They are

widely used in applications such as secure email, digital signatures, electronic voting, e-commerce and digital cash. Hash functions are more efficient than cryptographic primitives such as symmetric and asymmetric ciphers [2].

## CHAPTER 2

### REVIEW OF LITERATURE

---

Biometrics and integrity are two technologies which are potentially complementary. Biometrics ensures the identification and authentication of the individuals based on observation of their personal unique features, integrity guarantees trust in the transactions. The idea of combining biometrics with security has taken more importance due to the threats involved with e-transactions.

Chen et al. (2007) in the paper entitled “**Biometric Based Cryptographic Key Generation from Faces**”, framed a system to generate stable cryptographic keys from biometric data which involved the use of distinguishable facial features. Pin numbers and passwords have a possibility of being forgotten or stolen thus giving rise to use of biometric features. Therefore, a long and more stable bit stream is generated as the cryptographic key. The verification is carried out on face database. Thus, the biometric data can be taken as a good alternative, or supplements, to PINs and passwords. The whole system is tested for False acceptance rate and false rejection rates, however no tests on the generated keys has been done. [11].

Hao et al. (2006) have presented a technique by the integration of iris biometric data into cryptographic systems in the paper entitled “**Combining Cryptography with Biometrics Effectively**”, A repeatable binary string, known as the key has been generated from the iris codes. The main issue is generation of an error free key, to resolve this, the error patterns have been studied and a two-layer error correction technique has been presented which is a combination of Hadamard and Reed-Solomon codes. Evaluation is done on different eye samples considering many samples of each eye. The key generated is of 140 bits [12].

Chen et al. (2011) proposed a qualified RNG in the paper entitled “**Audio Random Number Generator And Its Application**”, the RNG generates random number sequences from microphone or film’s sound source. This requires only a microphone and is verified using NIST tests. This algorithm is transferred into personal devices like smart cellular phone so as to protect personal privacy. Hence, a qualified random number sequences can be made from single

sound source as well. The sequence passes the randomness criteria significantly and can be used in various applications [13].

Wei et al. (2013) in paper entitled “**Image Encryption Algorithm Based on the Key Extracted From Iris Characteristics**”, have formulated a new system based on vascular pattern of human retina is presented which can be used for security purposes. It comprises of three stages; i.e. preprocessing, feature extraction and finally the matching process. In preprocessing, it extracts the vascular pattern from input retinal image. Second stage extracts all possible feature points and represents each feature point with a feature vector. The proposed system matches the template feature vectors and input image feature vector [14].

Garcia et al. (2009) have worked on development of an approach for biometric key generation which uses electrocardiogram (ECG) signals. In the paper entitled “**A Wavelet-Based 128-bit Key Generator using Electrocardiogram Signals**”. The stages comprising the approach are one time enrollment and key derivation. This work is based on the uniqueness and behavior of ECG signals with respect to an individual. Thus, it is concluded that the ECG signal can be taken as an efficient biometric characteristic and it also guarantees the generation of different information [15].

Hedayatpour et al. (2011) proposed an inexpensive and less complicated method to generate true random values, in paper entitled “**Hash Functions-based Random Number Generator with Image Data Source**”. Hash functions are used to combine two data extractors to transform normal image data and make it suitable as the source of random number generator. In general, when a hash function operates on the any source of normal data, the result will be in form of pseudorandom. Thus, in this proposed method, two hash functions (MD5 & SHA-2) are used as the data extractor that will transform a normal image data into a data source, which is suitable in generating the random numbers [16].

Ying et al. (2010) presented a novel authentication protocol involving use of a contactless undistorted fingerprint images to generate authentication key, in the paper entitled “**Design of A Random Number Generator from Fingerprint**”. An additional layer of security can also be provided using blood vessel map to determine a live finger is being used. A system is developed to improve security of network communications using contactless fingerprint images which provide undistorted and consistent image of the fingerprint, capable of generating unique



passwords. If keys from one session are available to attackers, they will not be able to use this information to create new sessions. Even if multiple keys are available to an eavesdropper, it would not be possible predict the keys used by the server for a particular transaction, nor will a single key provide information to recreate the entire fingerprint image [17].

Khokher et al. (2013) have used an ECG signal to generate a 128 bit key, in the paper entitled **“Generation of Security Key using ECG Signal”**. An ECG vector is formed by measuring the R-R distance of the consecutive peak values, thus forming a vector that consists of whole numbers. The final security key is obtained by adding these three sequences and representing the values in hex form. The final vector is the security key which can be used for various security purposes. The key has although not been tested for its randomness, which is a basic criteria for an efficient security key. The efficiency of this security system is evaluated using various system efficiency testing algorithms which include FAR, FRR [18].

Liu et al. (2010) have formed a biometric cryptosystem based on face Biometrics, in the paper entitled **“A Novel Key Generation Cryptosystem Based on Face Features”**. At the encryption stage a 128 bit vector is initially extracted from the face image. Further, an error-correct-code (ECC) is also generated using Reed-Solomon algorithm. In decryption phase, a 128- dimensional features vector is extracted from the query face image. Then a key is generated using the look-up table generated at encryption stage. The final key is obtained using both bio-key and Error correct code (ECC). Finally, the symmetric decryption algorithm implemented to obtain message using final key [19].

Pal et al. (2014) in the paper entitled **“Secured Data Transmission through Audio Signal”** propose a novel technique for secured transmission of sensitive data by hiding any type of message in a carrier audio signal. The algorithm uses Discrete Fourier Transform (DFT) to convert the carrier audio samples into frequency domain after they are decomposed into small non-overlapping frames. Message bits are fabricated in transformed frequency coefficients at pseudo-random bit positions to defeat any unauthorized extraction attempt as the bits can only be retrieved accurately by reproducing the exact sequence of bit positions at receiving end with the help of the same pseudo-random number generator. Also to guarantee the integrity and authenticity of the hidden data a message digest of the hidden message is inserted into the carrier. At receiving end both the message and message digest are extracted. The extraction

algorithm regenerates a new message digest of the extracted message and compares it with the one received [20].

Varma et al. (2014) present a DNA cryptographic scheme based on matrix manipulation and generation of a security key which can be used for encryption, in the paper “**Cryptography Based on DNA using Random Key Generation Scheme**”. The main advantage of the scheme is that a new cipher text is obtained for the same plain text every time. Thus, a higher level of security is provided. This technique is helpful in preventing attacks on data sensitive systems [21].

Van et al. (2015) presents a MAC Scheme in the paper entitled “**Privacy Preserving Message Authentication Code**” which supports to verify data integrity from partly information of the original data. In addition, it has been proved to be chosen-message-attack secure and privacy-preserving. Also an experiment is conducted to compare its computation cost with a hash message authentication code [22].

Abduljabbarl et al. (2014) proposed a message authentication code in the paper entitled “**An Efficient and Robust One-Time Message Authentication Code Scheme Using Feature Extraction of Iris in Cloud Computing**”, based on the feature extraction of the user's iris in order to verify data integrity. This scheme enjoys several important security attributes such as a user's one time bio-key, robust message anonymity, data integrity for a user's message, phase key agreement, bio-key management, and one time message code for each user's session. The security analysis and experimental results demonstrate and prove the invulnerability and efficiency of this technique [23].

Lan et al. (2016) in the paper entitled “**An Area-Efficient Implementation of a Message Authentication Code (MAC) Algorithm for Cryptographic Systems**”, present an area-efficient hardware implementation of the lightweight Chaskey algorithm. The main targets of this work are resource-constrained devices. An efficient and simple design is employed in order to achieve the goal. Different implementation methods of Chaskey algorithm are investigated [24].

Verma et al. (2016) proposed a novel SHA algorithm in the paper entitled “**Robustness and Security Enhancement of SHA with Modified Message Digest and Larger Bit Difference**”,

The Proposed SHA algorithm has eighty steps in all and in each step there is an fundamental function which every time calculates a message digest and sends it to the next step. There are significant changes in the elementary function of the secure hash algorithm and also give us a message digest of length 176 [25].

**CHAPTER 3****PROBLEM FORMULATION**

---

- (1) In wireless devices, battery power conservation is a major task, therefore to decrease the consumption of battery, data security mechanisms are avoided. However in this condition the opponent is able to exploit the data confidentiality and integrity.
- (2) In data communication, due to the data being available on air interface, it is possible to be captured and altered by the attacker, which may lead to false information being received at the destination.
- (3) Forgery attacks in the past have proved that existing hash algorithms are not reliable enough to be used for verification and preservation of data integrity, due to lack of complexity.
- (4) The security keys traditionally generated using pseudo random number generators are susceptible to seed attacks.

## **CHAPTER 4**

### **OBJECTIVES**

---

1. To generate an efficient security key which is random and cannot be duplicated and thus can be used to prevent brute force and man-in-the middle attacks.
2. To prevent and detect any alterations in the received data.
3. To implement a robust novel hash algorithm for preventing data alteration, that can be used for preserving the data integrity.
4. Implementation of MAC through fusion of the secret key and a novel hash algorithm.
5. Comparison and analysis of the proposed technique with the existing ones.
6. Analyzing the behavior of the technique when subjected to various network attacks.

**CHAPTER 5****RESEARCH METHODOLOGY**

---

Security of confidential data is the most critical design parameter of any communication network. In communication networks the main aim is to provide better efficiency without compromising the security performance, in order to make it feasible for resource constraint environment.

Data security can be realized using cryptography which refers to the practice of hiding the data in order to prevent its open access to the intruder. In cryptography, a Message Authentication Code (MAC) is used, it's a short piece of information used to authenticate a message. In other words, it is used to confirm that the message has been received from the intended sender (its authenticity) and hasn't been altered in transit (its integrity). MAC uses a key that is known only to the originator of the message and its intended recipients. This helps in maintaining the integrity at the receiver end. The key formulation for the proposed algorithm is explained in Figure 5.1.

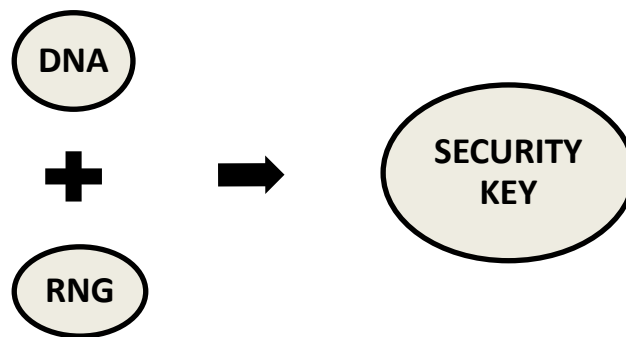


Figure 5.1: Security Key Formulation

The fusion of DNA and Random Number Generator sequence are used to formulate a security key which further would find its application in MAC algorithms thus for providing security. The work is to be carried into three phases, in order to generate an efficient MAC algorithm, the process is explained in Figure 5.2

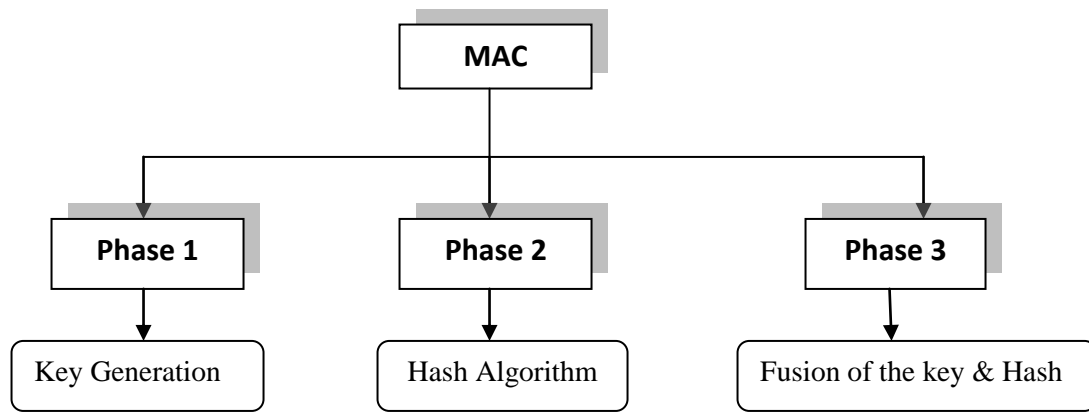


Figure 5.2: Proposed Research Methodology

## 5.1 Phase 1: Security Key Generation

In this phase the main objective is to generate a unique and random security key. The key proposed in this work is a combination of DNA sequence and the sequence obtained by the RNG using the secret seed provided by the user.

### 5.1.1 DNA sequence formulation

- 1) Obtaining a DNA sequence of 1024 characters from the DNA database [9].

The DNA sequence consists of base pairs ‘agct’.

- 2) *Obtaining the binary sequence from DNA characters:*

Each character of the DNA sequence is represented by 8-bit ASCII code. Hence, resulting in a DNA sequence of length 8192 bits.

- 3) *Framing a DNA sequence of 256 bits:*

- (i) The DNA sequence is then divided into equal halves.
- (ii) Apply exclusive-or operation on the obtained sequences.
- (iii) The result is further divided into two equal parts and exclusive-or operation is applied again.

The step (iii) is repeated until a sequence of 256 bits is obtained. The conversion from ‘agct’ sequence of 1024 characters to 256 bit binary DNA sequence is tabulated in Table 5.1. The DNA sequences are represented as  $D_1, D_2, D_3$ .

Table 5.1: DNA Sequence Formation

	DNA sequence (1024 Characters)	DNA sequence (8192 bits)	DNA sequence (256 bits)
D <sub>1</sub>	gcacaatcagaagcaggcggaggagacg gcggccttcgaggaggtcatgaaggacctg agcctgac..... ..... ..... ..... ..... .....gcacagaggaagcgctca gcaggcatcggccaccctgtctccgctgtca cccatcactcaggctgtagccatg [26].	0110011101100011011000010 1100011011000010110000101 11010001100011..... ..... ..... ..... .....011101000110000101100 1110110001101100011011000 010111010001100111	000001000000000000010101000 101110000001000000110000101 010000010000000110000001100 000011000000110000100110000 00000000000000100110000000 00000001000000110000000000 00011000000110000001000000 01000010011000001100000010 00000000000001000000000000 1001100000000
D <sub>2</sub>	ccacgcgtccggcgagaagatggcgact tcgaacaatccgcgaaattcagcgagaag atcgcgct..... ..... ..... ..... .....ggcgtcagccccctgtccctgagca cagaggcaaggcgtcagcaggcatcgccc gccctgtccccgctgtcaccat [27].	0110001101100011011000010 1100011011001110110001101 10011101110100..... ..... ..... ..... .....0111010001100011011000 0101100010110001101100011 0110000101110100	000000100000001000000010000 000100000011000010111000001 000001011100000100000000000 000000000000000000001100000 001000000110000101010001001 10000001000000000000000000 000010000100110000011000000 00000000100000011000000110 00000010000000000010111000 0010000000100
D <sub>3</sub>	agcccttaggggaagatcctgctgctg ttgatgctccagctccagaatcccagctac gcaactg..... ..... ..... .....tctggagcagcagctgccctacgctt cttaccagcgggctcccagcagccacc gccgagccccagccccg [28].	0110000101100111011000110 1100011011000110111010001 11010001100001..... ..... ..... .....10001101100011011000 1101100011011001110110001 10110001101100111	00010101000000000000010000 101110000010000010111000000 000000001000000100000001100 000000000000100000101010000 00100000010000000000000000 000010101000000100000001000 00000000000100000000000010 1110000001000010011000010011 0000011000000000000010111000 1001100010011

## 5.2 Generation of the RNG Sequences

The proposed algorithm uses three different random number generators and thus three random sequences are generated using each of the generators. These random sequences are generated by using a secret seed value every time, which is confidential to the user. The seed value ensures a different sequence each time, thus maintaining the uniqueness in the sequences generated.

### 5.2.1 BRNG Sequence Generation

Three sequences are generated using three different seed values. A seed value is a secret input given by the user at the beginning of generator operation.



The formula implemented here is given in equation 8. The obtained random sequences along with the corresponding seed value are summarized in Table 5.2.

Table 5.2: BRNG Produced Random Sequences

	Seed value	Random sequenced through BRNG (256 bits)
B <sub>1</sub>	1231	101100110000001101011010011000010101001000010111011110111001110100000001 1000001101010110100010101101110101000110111010101011000000011001110110 10000011011001100101111011100101000100010110010000000101110000111101101 0111011110000011011110000110010010010110
B <sub>2</sub>	101355	000101101001000101010001100101100111000000110101111110011110101011100010 010110010100001110110111001110000100110011001101101110100101110011011101 00011000001101111100111111110011010000011111001100100100110001000100011 0011011100100100100101010110101101010001
B <sub>3</sub>	2114	0111010100001101001100000101010100000010001000010101010101010100000110101 111010101010011010011010010100000110001000101011100000001000111001110001 000011001001011111000101110001010011010111011111101011000110001100010111 0110010110100100100001111011110001101011

\* B<sub>1</sub>, B<sub>2</sub>, B<sub>3</sub>: Three BRNG sequences

### 5.2.2 BBSG Sequence Generation

Three sequences are generated using three different seed values. The formula implement here is given in Equation 9. The obtained random sequences along with the corresponding seed values are summarized in Table 5.3.

The values assigned to the variables in Equation 9 are:  $p = 383$ ;  $q = 503$ ;  $m = p*q$ .

Table 5.3: BBSG Produced Random Sequences

	Seed value	Random sequences through BBSG (256 bits)
B <sub>1</sub>	101355	11001010000100111011110010001000011101001111001011011100110011111001 01010011011100101011000110010010001000010100101000000100000101000011 10100110001110101001101101011101100101100110010011001000011100111100 1100011011011101100100101101111011101110001001100110
B <sub>2</sub>	101357	10111010000110011000100010011111101011010101000101111101110111111000 11011011111101100110101000111110111101111000001100011010111111010010 00101010001011111011101000000101111000001011111000011000101011001 1010000111000101100111111010100011101100011011001
B <sub>3</sub>	101359	11100011001000101001110110101100000011101110101100111010111001111010 01001110101101110110011011110111101111010000010010111000001011111010 10110000000110110010101011111011110001111111100010001001101111001 1100011110101010100010110001101011011101001011001100

\* B<sub>1</sub>, B<sub>2</sub>, B<sub>3</sub>: Three random sequences.

Different seed values ensure generation of a different key every time.

**5.2.3 LCG Sequence Generation:**

LCG generates the random sequence on the basis of equation (10). The values assigned to the variables in the equation are:  $a = 23$ ;  $c = 0$ ;  $m = (10^8+1)$ ,

Three sequences are generated using three different seed values. The obtained random sequences are summarized in Table 5.4.

Table 5.4: LCG Produced Random Sequences

	Seed value	Random sequences through LCG(256 bits)
L <sub>1</sub>	47594118	01010000010001100011011010011011010110111100111110101010010010011110111 0000011111111001100000101101110010100001111010110111111000111011011111 00101111111111010111010111100110110001001000100100010100111000001110110 0101111010010000001001110100001110000000010
L <sub>2</sub>	47594122	01010001010001010011001101000111011111101011101000110001000011101100101 01001110100110100101101001111111000000101100111000000110100001000100101 01011101110101011011100000111101100000011110010100111000111101011010110 0000110001100010011000001101111000110101101
L <sub>3</sub>	34359738369	10010000001010110011000000111000101000110011010011011010111011000100000 10101001100010110000111010100100000110011000110110011110000001000011100 11111001001101101101100010000010001010010000110100001011000101111110100 1010011100100110110001110010111110011000000

\*L<sub>1</sub>, L<sub>2</sub>, L<sub>3</sub>: LCG sequences

**5.3 Fusion of DNA Sequences and RNG Sequences**

Further to escalate the impact of randomness, logical operation is applied between each DNA and RNG sequence. This is done to frame a final key which can be used for providing security. This is repeated for two other sequences also. Finally, three 256 bit keys are obtained for each RNG used.

**5.3.1 Fusion of DNA and BRNG sequences:**

The fusion of DNA and BRNG using exclusive-or results in a key with length of 256 bits which is strong enough to survive critical attacks on networks.

The method of the generation of these keys is portrayed in Figure 5.3

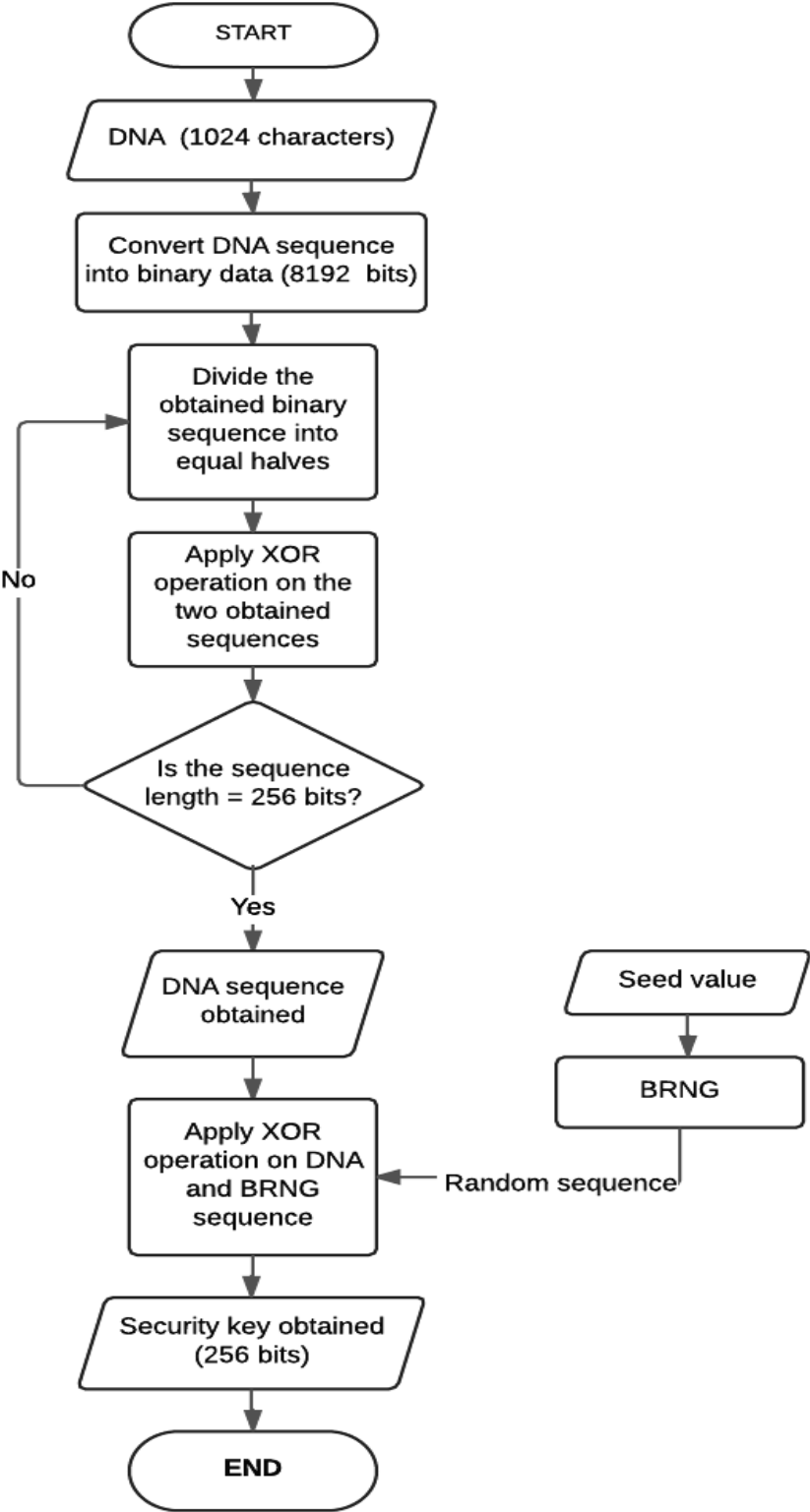


Figure 5.3: Key Generation Process for DNA and BRNG

The process is repeated for two other DNA and BRNG sequences. Therefore, three 256 bit keys are obtained as shown in Table 5.5.

These keys are represented as KEY<sub>1</sub>, KEY<sub>2</sub>, KEY<sub>3</sub>.

Table 5.5: Security Keys generated using DNA and BRNG

KEY <sub>1</sub>	0001001010010001010001001000000101110010001100111110110011101110111001000101111 1010001011011000100101011010011001100110110101001010111001101111100011110001101 1111001001111111111010010011111011100000010110010000100001001101110010000010010 1010111100001010001
KEY <sub>2</sub>	1011000100000001010110000110001010101111000111001011100111011001100001001100000 1101010110100010101101000101000010111001101000011000100000011100101000001101100 1100101101011000011000111010110010000000110111011011110000101110011100000110101 0110011011001001111
KEY <sub>3</sub>	0110000000001101001100100100001000000101000001111010101010101010001100011110110 0101001101001111001000101011000000010111110000000100011100110010000001110100101 011100010111000001001101011100100010101110011100000000100011000111010010010010 0001010111101111000

The final key is a combination of true random source (DNA) and pseudo random source (BRNG). The keys obtained have unique and random characteristics and thus are useful to prevent critical attacks on security.

### 5.3.2 Fusion of DNA and BBSG Sequences

The fusion of DNA and BBSG sequence is done in order to further increase the randomness in the sequences generated and to make the system more complex. Therefore, for fusion, the logical operation Exclusive-or is applied between every DNA sequence and the Blum Blum Shub Generator produced random sequence.

This leads to the formation of final 256 bit keys which are unique and random and thus can be used for providing security in a data sensitive environment. The process of fusion is explained in Figure 5.4.

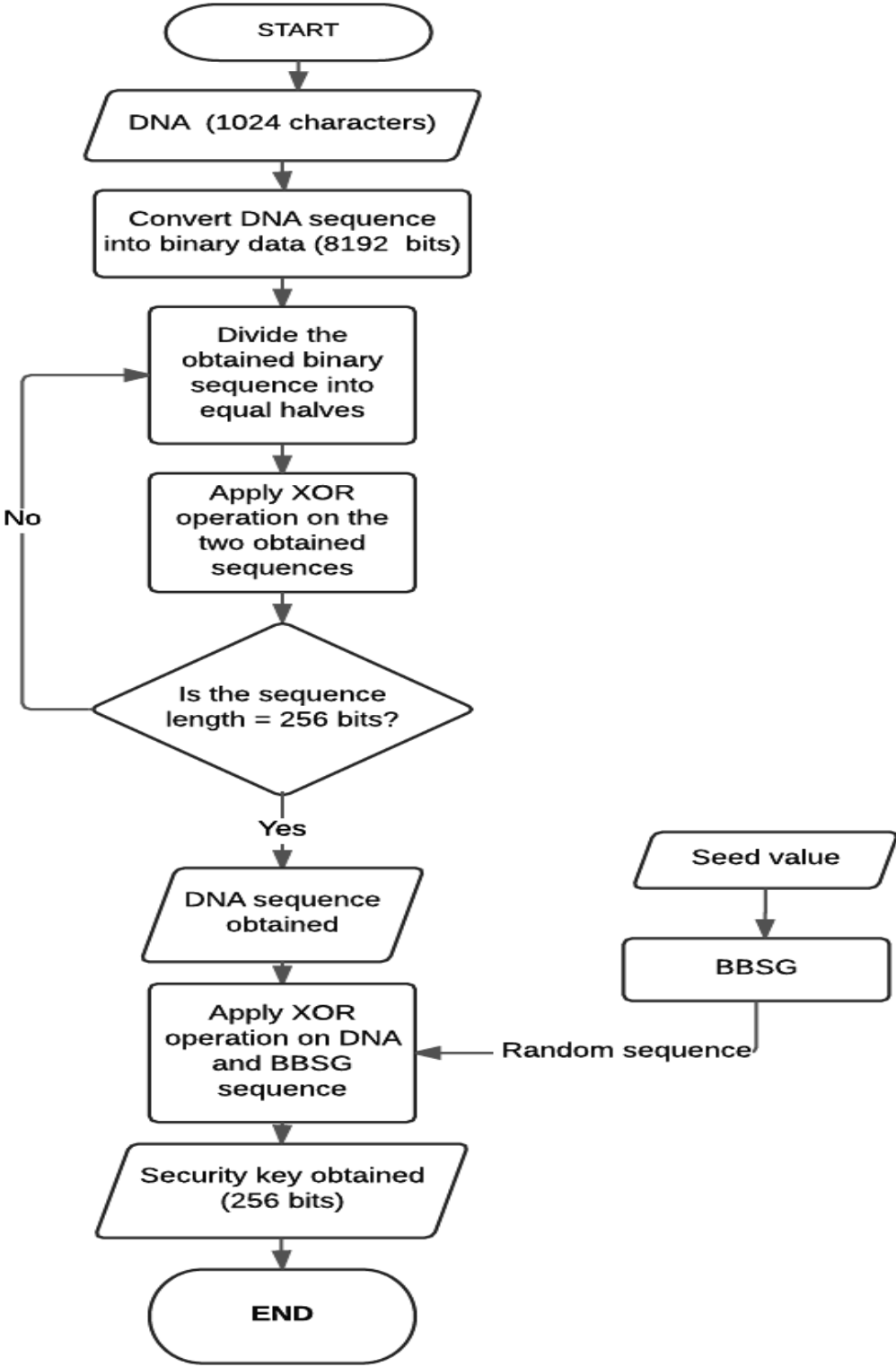


Figure 5.4 Key Generation Process for DNA and BBSG

The same process is repeated for two other DNA and BBSG sequences. Therefore, finally three 256 bit keys are obtained as shown in Table 5.6.

These keys are represented as KEY<sub>1</sub>, KEY<sub>2</sub>, KEY<sub>3</sub>.

Table 5.6: Security Keys generated using DNA and BBSG

KEY <sub>1</sub>	1100101000010011101111001000100001110100111100101101110011001111100101010011011 1001010110001100100100010000101001010000001000001010000111010011000111010100110 1101011101100101100110010011001000011100111100110001101101110110010010110111101 1101110001001100110
KEY <sub>2</sub>	1011101000011001100010001001111110101101010100010111111011101111110001101101111 1101100110101000111110111110111100000110001101011111101001000101010001011111101 1101000000010111110000010111111000011000101011001101000011100010110011111101010 0011101100011011001
KEY <sub>3</sub>	1110001100100010100111011010110000001110111010110011101011100111101001001110101 1011101100110111101111011110100000100101110000010111110101011000000011011001010 101111110111100011111111100010001001101111001110001111010101010001011000110101 1011101001011001100

The obtained keys are unique and random and thus increase the level of security. Therefore they can be widely used as a part of the high tech security systems.

### 5.3.3 Fusion of DNA and LCG Sequences

The next step taken is to apply Exclusive-or logic between every DNA sequence and the LCG produced random sequence. This is done to develop a final 256 bit key which can be used for providing security.

The fusion process increases the randomness in the keys and it also makes the system more complex and robust. The procedure followed is same for all the three DNA and LCG sequences.

The process of fusion is explained in Figure 5.5. The same procedure is applied for the two other sequences and the keys produced are referred to as KEY<sub>1</sub>, KEY<sub>2</sub>, KEY<sub>3</sub>.

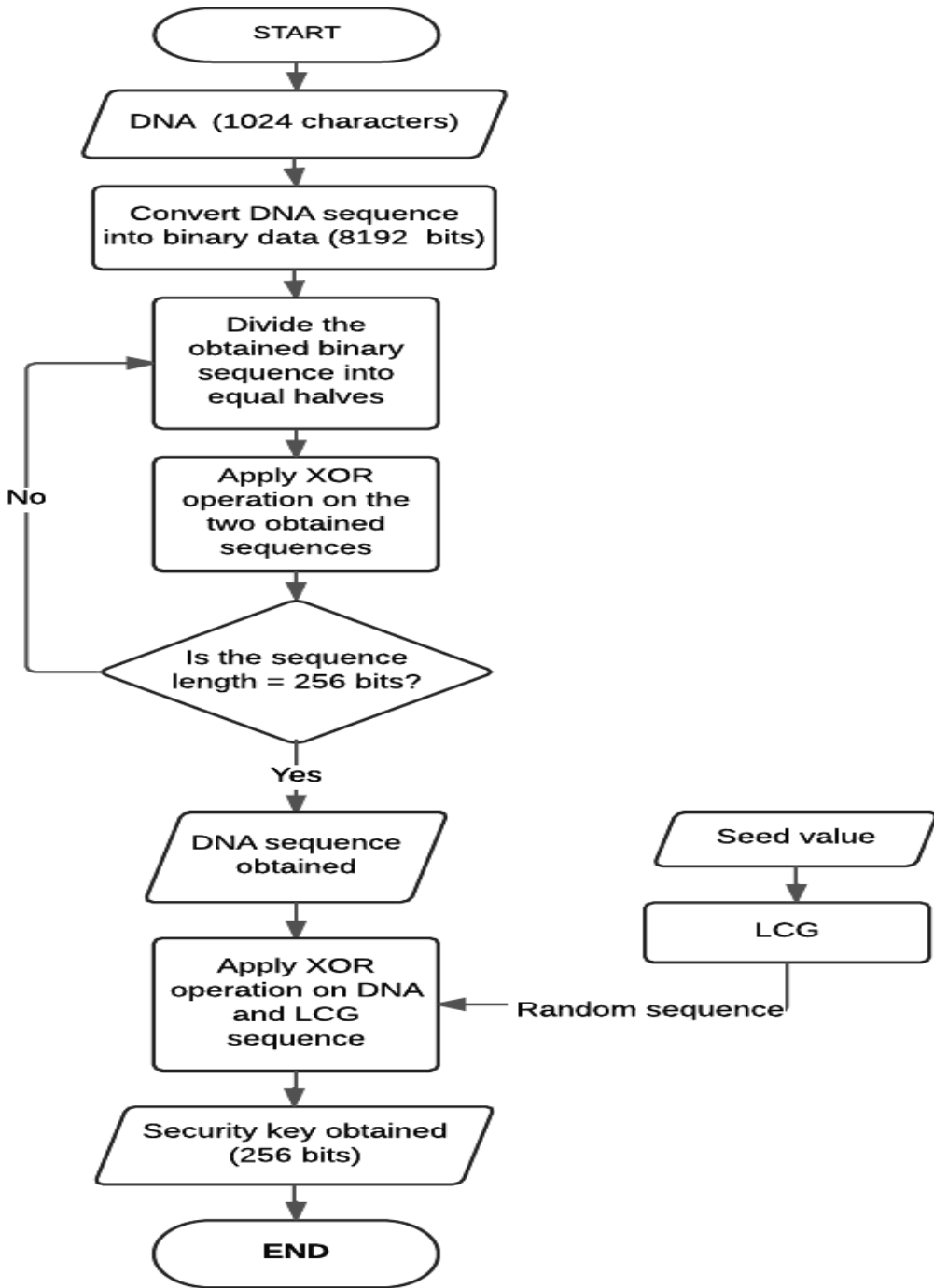


Figure 5.5: Key Generation Process for DNA and LCG

The fusion process is repeated for two other DNA and LCG sequences. Finally three 256 bit keys are obtained as shown in Table 5.7.

Table 5.7: Security Key generated using DNA and LCG

KEY <sub>1</sub>	010101000100011000100011100011000101100111001001101111110100110111101000000100111 110101000000111010101001000011110101101110111101110111111010111001111101011101 0001100111010001011000100110010000001000010110110000111101001000010100111010000011 1100000010
KEY <sub>2</sub>	0101001101000111001100010100010101111000101011010011010100011001110011101001110100 1101001011010011111000000001111001101000011000000110111001011101110111010101101110 0010111001010000000110010100111000011101000010110110011000010001001100010001111101 0110101001
KEY <sub>3</sub>	1000010100101011001100100010111110100111001000111101101011101110010001010101010100 0101100001100101011101001100010001111100111100000010000110011011100110110110010110 0010000011001010010000100011001011100100110010110110001111110011011000101110011011 1111010011

The obtained keys are unique and random and thus can be used to prevent brute force attacks in a communication network.

### 5.4 Phase 2: Generation of a Novel Hash Algorithm

The formulated novel hash algorithm is a secure one-way hashing algorithm of 160-bit, framed by enhancing the complexity of the existing hash function. It is clearly observed from Fig. 5.6 that the proposed scheme inherits the basic architecture of SHA-160 algorithm. SHA-160 algorithm is a cryptographic hash standard which was recommended by Network Working Group under RFC [5]. This was designed by NSA (National security Agency) to be a part of digital signature algorithm.

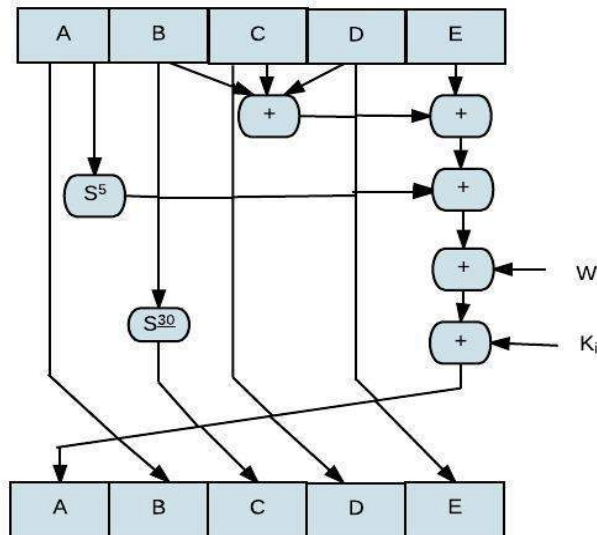


Fig. 5.6: SHA-160 Compression Function



In 1995, it was published by National Institute of Standards and Technology (NIST), as a U.S Federal information processing standard (FIPS). It takes an input of varying length and produces a 160-bit message digest. The whole compression function consists of 80 rounds [5]. To enhance the impact of SHA-160 in terms of randomness, Expansion and S-box operations have been added into the existing architecture. These amendments increase the system complexity and thus help in preserving data integrity.

The proposed scheme also enhances the system throughput by providing a message digest of smaller length as compared to other hashing algorithms of SHA family which include SHA-256, SHA-384 and SHA-512.

#### 5.4.1 Integration of ‘f’ function:

The proposed technique derives its strength from the ‘f’ function, which has been integrated into the existing procedure. This ‘f’ function constitutes of an expansion technique and Substitution block. The integration of this function results in an increase in the complexity of the overall algorithm.

The expansion technique used here is Symmetric extension which transforms the 32 bit input data into 48 bit data. Later,  $2^{48}$  modulo addition is applied followed by the substitution box, which is used to convert the 48 bit data to 32 bit data. The details are given in *Appendix A*.

‘f’ function performs the following tasks:

The expanded value of *register E* and *function ‘f<sub>t</sub>’* containing 48 bit data is provided as an input to perform  $2^{48}$  modulo operation as shown in equation 11 and then the output that consists of 48 bit is given to the ‘S’ block as in equation 22 in order to convert the 48 bit data to 32 bits.

$$\left[ \begin{array}{l} Y = \text{mod} \left( (\text{Exp}(E) + \text{Exp}(f_t)), 281474976710656 \right) \\ Y1 = S(Y) \end{array} \right] \quad (11)$$

- Further, this ‘Y1’ value consisting of 32 bits is used for proceeding operations. The structure of the proposed hash algorithm is explained using Fig. 5.7.

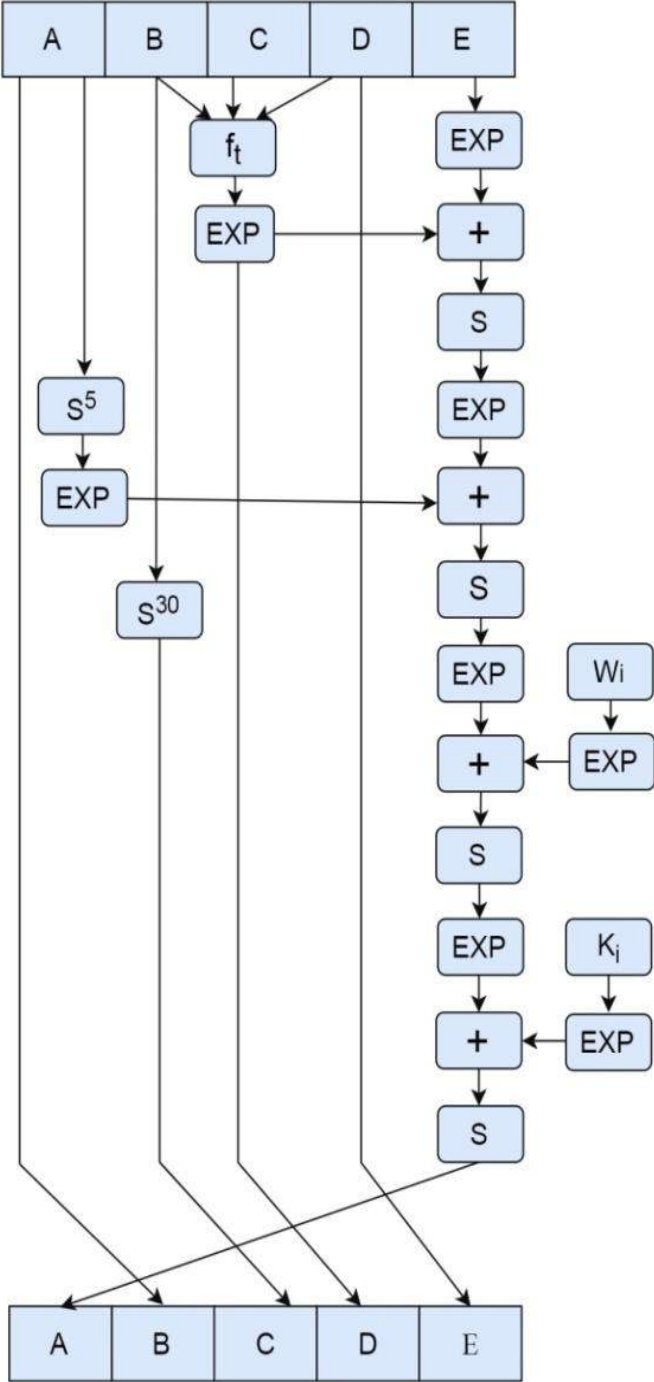


Figure 5.7: Structure of Proposed Hash Algorithm

The novel hash algorithm gives an output of 160-bits. The inputs of different lengths along with their corresponding hash values, resulting from various existing techniques and the recommended technique are given in Table 5.8.

Table 5.8 Message Digest Values for Different Inputs

Hash Techniques	Hash values		
	P	World	gurpreetkour
<b>MD2</b>	3687e026b0a5f81a9fabd5205d804615	bb37795d20176658552b6da07694861c	9c10bdde1d3aabbe95038e9062f9bb44
<b>MD5</b>	83878c91171338902e0fe0fb97a8c47a	7d793037a0760186574b0282f2f435e7	fa2c20167700f35b98fa7ec01b11d84c
<b>SHA-160</b>	516b9783fca517eecbd1d064da2d165310b19759	7c211433f02071597741e6ff5a8ea34789abbf43	eedb824ffed03619e62b42f61f6e7ded24010aae
<b>SHA-256</b>	148de9c5a7a44d19e56cd9ae1a554bf67847afb0c58f6e12fa29ac7ddfca9940	486ea46224d1bb4fb680f34f7c9ad96a8f24ec88be73ea8e5a6c65260e9cb8a7	fd878264b337d653f712a333cdc8cb0ab0bb08a9a5eb585c6c6fc55b7acd27c8
<b>SHA-384</b>	049e7caf67d83409ea363e89c09d67c7f1fd1bd679016ad9f422830ef105435e12a4c2dcad5a9e5a9602924d479574dc	ed7ced84875773603af90402e42c65f3b48a5e77f84adc7a19e8f3e8d310101022f552aec70e9e1087b225930c1d260a	805dba18a4c056b8391685ba8113a8372d4f2bb217b2f83a61d2a34efb9f7b8b6979e4ab51561a25168e915efcc11252
<b>SHA-512</b>	929872838cb9cfe6578e11f0a323438aee5ae7f61d41412d62db72b25dac52019de2d6a355eb2d033336fb70e73f0ec0afeca3ef36dd8a90d83f998fee23b78d	11853df40f4b2b919d3815f64792e58d08663767a494bcbb38c0b2389d9140bbb170281b4a847be7757bde12c9cd0054ce3652d0ad3a1a0c92babb69798246ee	f1b814483d475967960077ad0868aacc96f635c9c535b79f1ed5e0e00f7afef710830c73f5d06cb6850093129433061bf87d02753e5f12f082da4477dba54311
<b>Proposed Hash</b>	4283483b3e3a8b6d5dcd699c274d1d3774e17d44	617d79dc7a2b56691f5be1fa94a76db8354d62f3	db13fea29dc8d45725e6fe179dac64111107a091

### 5.5 Phase 3: Formation of MAC:

The third and final stage involves combining the two phases and thus developing a secure and robust MAC algorithm which will ensure data integrity. Since, the key formulation involved generating the key in three different ways, therefore the MAC is generated in three ways, each is described in a different section.

#### 5.5.1 Formation of MAC using DNA-BRNG Key and Novel Hash Algorithm:

MAC is also known as keyed Hash in reference to its components i.e. a hash algorithm and a secret key. The DNA-BRNG key here is of 256 bits and this key has to be further used in the form of four 32-bit keys, therefore the operations applied on the 256-bit key to convert it into four 32-bit keys are explained using Fig. 5.8.

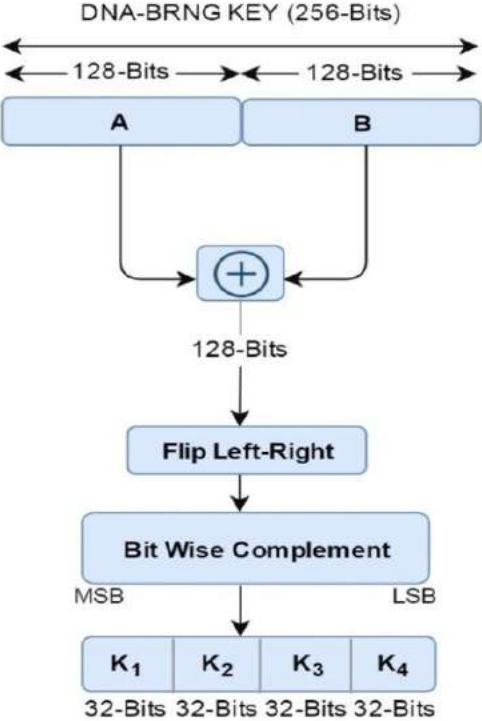


Figure 5.8: Operations applied on DNA-BRNG Key

The four final keys are in hexadecimal form as shown in Table 5.9. The SHA-160 algorithm uses four 32-bit constant values [9] which are replaced with these four keys.

Table 5.9: Security Keys using DNA-BRNG for SHA-160

Key. No.	32-bit Keys (Hexadecimal)
K <sub>1</sub>	E052642F
K <sub>2</sub>	9ED92359
K <sub>3</sub>	57EDCC22
K <sub>4</sub>	92A58D8D

The MAC values obtained, using the proposed technique are presented in Table 5.10.

Table 5.10: MAC Values for novel hash and DNA-BRNG key

Input	Hex form	MAC Values (Hexadecimal)
g	67	a15118eef9ee8f1159656890c691c7979164110a
Sodhi	536f646869	11ec223fea6b36c2d93893d65ef1369cc8e21864
unitedstates	756e69746564737461746573	ae1d2843dcc97133003918c0ebd47c4f583e01c9

The computed MAC values are then converted into binary form for evaluation on randomness and avalanche criteria.

### 5.5.2 Formation of MAC using DNA-BBSG Key and Novel Hash Algorithm:

The DNA-BBSG key is of 256 bits and this key has to be further split into four 32-bit keys in order to be used for MAC, therefore various operations are applied on the 256-bit key in order to convert it into four 32-bit keys, the operations are explained in the given steps:

1. Apply 6-bit circular shift on the 256-bit DNA-BBSG key four times and store the sequence every time its shifted.
2. Then split the 256 bit shifted key into four parts of 64-bit each.
3. Apply Exclusive-or operation on first two parts, followed by repeating it for the last two parts, resulting in a sequence of 64-bits.
4. Split the obtained 64-bit sequence into two parts of 32-bit each
5. Apply Exclusive-or operation between the obtained sequences.
6. Convert the obtained 32-bit sequence into hexadecimal form, forming the 8-bit hex key.
7. Repeat steps 2 to 6 every time the key is shifted by 6-bits, thus obtaining final four keys.

The final four keys are represented in hexadecimal form as shown in Table 5.11. The SHA-160 algorithm uses four 32-bit constant values [7] which are replaced with the four keys which were framed using the operations.

Table 5.11: Security Keys using DNA-BBSG for SHA-160

Key. No.	32-bit Keys (Hexadecimal)
K <sub>1</sub>	CD2740EB
K <sub>2</sub>	AF349D03
K <sub>3</sub>	0EBCD274
K <sub>4</sub>	D03AF349

The MAC values obtained, using the proposed technique are given in Table 5.12.

Table 5.12: MAC Values for novel hash and DNA-BBSG key

Input	Hex form	MAC Values (Hexadecimal)
g	67	d4be6361479e2705954f21641c92d9a37c65e3
Sodhi	536f646869	9ced64e520ccfd741b8ed38326e33a6b88a65b52
unitestates	56e69746564737461746573	f29d903139058701e39e3ceb1c1776316454ef0c

The computed MAC values are then converted into binary form for evaluation on randomness and avalanche criteria.

### 5.5.3 Formation of MAC using DNA-LCG Key and Novel Hash Algorithm:

The generated DNA-LCG key is of 256-bits and to use it in MAC it needs to be converted into 4 32-bit keys, this conversion is done by using few operations which are explained in Table 5.13.

Table 5.13: Pseudo Code for Operations on DNA-LCG Key

<pre> Y= initial key of 256-bits  for i=0:7 x(i+1,:)= y1(32*i+1:32*(i+1))    //splitting the key into 8 parts of 32-bit each end  for i=1:4 k(i,:)= ~xor(x(2*i-1,:),x(2*i,:)) //Applying ex-or operation between each consecutive pair forming four 32-bit sequences and then complementing the results end  //k represents the key// //The four key values are converted into hexadecimal form to be used in MAC//                 </pre>
--

The four final keys are in hexadecimal form as shown in Table 5.14; these four keys replace the four 32-bit constant values used in SHA-160 [8].

Table 5.14: Security Keys using DNA-LCG for SHA-160

Key. No.	32-bit Keys (Hexadecimal)
K <sub>1</sub>	F270633E
K <sub>2</sub>	BDB5DC13
K <sub>3</sub>	591C502C
K <sub>4</sub>	3A404009

This frames a novel MAC algorithm, the MAC values obtained, using the proposed technique are presented in Table 5.15.

Table 5.15: MAC Values for novel hash and DNA-BBSG key

Input	Hex form	MAC Values (Hexadecimal)
g	67	5caeebc819a2162167c042a9571bd535b8c80ed0
Sodhi	536f646869	1afbefabcc370d6d080e7a0c603313598ed17876
unitedstates	56e69746564737461746573	b12f957691b3161f7a5e56b3e58ceddf1c1c8f4c

The obtained MAC values are then converted into binary form to be evaluated on the basis of NIST tests.

## CHAPTER 6

# RESULT AND DISCUSSION

---

### 6.1 Analysis using NIST Tests

The efficiency of a security key is analyzed by inspecting its random characteristics and uniqueness. The National Institute of Standards and Technology (NIST) tests discuss some aspects of selecting and testing random number generators [8]. The outputs of such generators can be used in many security applications for the generation of security keys. The generators to be used for security applications need to be robust enough to handle attacks. In particular, their outputs should be unpredictable if there is no knowledge about the seed. These tests determine whether or not a generator is suitable for a particular security application. The randomness of a key is evaluated on the basis of its P-value, which should be greater than 0.01 for a random sequence. The keys obtained are evaluated on the basis of seven NIST randomness verification tests and the P-value is calculated for each test with respect to the security key. An overview of the NIST tests used for evaluating the sequences is given as:

#### 1) Runs test

The purpose of applying this test is to calculate the number of runs in an entire sequence, where run specifies the number of uninterrupted sequence of identical bits [8]. The results in Tabular form clearly depict that the proposed algorithm has higher rate of interruptions.

#### 2) Frequency Test

Frequency test is used for determining the proportion of number of ones and zeros in an entire sequence. It is used to check the closeness between the number of ones and number of zeros. A sequence is said to be random if the proportion of zeros and ones are close to each other [8]. The results clearly depict that the proposed algorithm produces better proximity between the count of ones and zeros as compared to the traditional techniques.

#### 3) Approximate Entropy Test

This test deals with finding the frequency of all the overlapping bit patterns in the sequence.

The aim of this test is to compare the frequency of overlapping blocks of two consecutive/adjacent lengths with the expected result for a random sequence.

#### 4) Discrete Fourier Transform Test (DFT)

The DFT test is used to find the peak heights in the Discrete Fourier Transform of a sequence. The aim of this test is to detect periodic features (i.e. repetitive patterns) in the tested sequence that indicates a deviation from the assumed randomness. The focus is to detect if the number of peaks exceeding the 95 % threshold are significantly different than 5%.

#### 5) Binary Derivative Test

The test is performed using exclusive-or operation between successive bits until only one bit is left. Afterwards, the ratio of number of ones to the length of entire sequence in each case is calculated. Finally, the average of the ratio of all the sequences is observed, if the value lies near to 0.5, then the sequence is considered as a random sequence [8]. The results in Table 4 show that the output of proposed algorithm is random.

#### 6) Maurer's "Universal Statistical" Test

The aim of this test is to detect if a sequence can be significantly compressed without any loss of information. The number of bits between matching patterns are calculated. A sequence that is significantly compressible is considered to be non-random. This test is also known as Universal test [8].

#### 7) Random Excursion Variant Test

The test focuses on the total number of times a particular state occurs in a cumulative sum random walk. It detects the deviations from the expected number of visits. The P-value computed using Equation 12 specifies if the sequence is random or not. This test considers successive sums of the binary bits as a one-dimensional random walk [8].

$$P_{value} = erfc \times \frac{(|\xi(x) - j|)}{\sqrt{(2 \times j \times ((4 \times |x|) - 2))}} \quad (12)$$

Where,

*erfc* : the error function



$\xi$ : the total number of times the state  $x$  occurs

$x$  : the state occurred

$j$  : the total number of cycles

The P-value must be greater than 0.01 for a sequence to be random [8].

### 6.1.1 NIST Test Analysis of Key Generated using DNA and BRNG.

The NIST results are compared with those of the traditional ones used and results tabulated in table 6.1. It is observed that the proposed technique produces better results in terms of randomness of the keys.

Table 6.1: NIST Test results for DNA and BRNG

S. No.	Input Source of random number generator	Key Length (bits)	Runs Test	Frequency Test	Approximate Entropy Test	DFT Test	Binary Derivative Test	Maurer's Test	Random Excursion Variant Test
			P-value	P-value	P-value	P-value	P-value	P-value	
1.	ECG [15]	128	0.1262	0.2487	0.5468	0.0294	0.5039	0.9428	Random
2.	Image [11]	256	0.0809	0.8026	0.9759	0.4220	0.4887	0.9780	Random
3.	Iris sequence [14]	128	0.1254	0.3768	0.9409	0.3304	0.5021	0.9062	Random
4.	Finger print [17]	128	0.3345	0.3041	0.3345	0.7597	-	0.2757	Random
5.	<b>DNA &amp; BRNG</b>	<b>256</b>	<b>0.0438</b>	<b>0.9005</b>	<b>0.9340</b>	<b>0.8185</b>	<b>0.5090</b>	<b>0.9920</b>	<b>Random</b>

It can be observed that the P-value obtained for the keys generated by proposed algorithm in all the seven tests is significantly greater than 0.01. Thus, it can be concluded that the keys generated are random in nature and hence fulfill the basic criteria required for security keys

### 6.1.2 NIST Test Analysis of Key Generated using DNA and BBSG

The efficiency of the proposed technique is evaluated by comparing it with other traditional techniques used in the field of authentication and security.

The tests have been performed on Key<sub>1</sub> and the results are shown in Table 6.2.

Table 6.2: NIST Test results for DNA and BBSG

S. No.	Input Source of random number generator	Key Length (bits)	Runs Test	Frequency Test	Approximate Entropy Test	DFT Test	Binary Derivative Test	Maurer's Test	Random Excursion Variant Test
			P-value	P-value	P-value	P-value	P-value	P-value	
1	ECG [15]	128	0.1262	0.2487	0.5468	0.0294	0.0039	0.9428	Random
2	Image [11]	256	0.0809	0.8026	0.9759	0.4220	0.0113	0.9780	Random
3	Iris sequence [14]	128	0.1254	0.3768	0.9409	0.3304	0.0021	0.9062	Random
4	Fingerprint [17]	128	0.3345	0.3041	0.3345	0.7597	-	0.2757	Random
5	<b>DNA &amp; BBSG</b>	<b>256</b>	<b>0.0809</b>	<b>0.8026</b>	<b>0.8540</b>	<b>0.4220</b>	<b>0.4995</b>	<b>0.9601</b>	<b>Random</b>

It is observed that the P-value obtained for the keys generated by the proposed algorithm, in all the seven tests, have significant values as compared to other techniques. Thus it can be concluded that the keys generated are random in nature and thus, fulfill the basic criteria for being used as security keys in a data sensitive environment.

### 6.1.3 NIST Test Analysis of Key Generated using DNA and LCG

The presented technique is evaluated by comparing it with other traditional techniques used in the field of authentication and security key generation using NIST Test. The tests have been performed on KEY<sub>1</sub> and the results are presented in Table 6.3.

Table 6.3: NIST Test results for DNA and LCG

S. No.	Input Source of random number generator	Key Length (bits)	Runs Test	Frequency Test	Approximate Entropy Test	DFT Test	Binary Derivative Test	Maurer's Test	Random Excursion Variant Test
			P-value	P-value	P-value	P-value	P-value	P-value	
1.	ECG [15]	128	0.1262	0.2487	0.5468	0.0294	0.5039	0.9428	Random
2.	Image [11]	256	0.0809	0.8026	0.9759	0.4220	0.4887	0.9780	Random
3.	Iris sequence [14]	128	0.1254	0.3768	0.9409	0.3304	0.5021	0.9062	Random
4.	Finger print [17]	128	0.3345	0.3041	0.3345	0.7597	-	0.2757	Random
5.	<b>DNA &amp; LCG</b>	<b>256</b>	<b>0.0809</b>	<b>0.8026</b>	<b>0.9497</b>	<b>0.4220</b>	<b>0.0608</b>	<b>0.9667</b>	<b>Random</b>

It is observed that the P-value obtained for the keys generated by proposed algorithm under all the seven tests is significantly greater than 0.01. Thus, it can be concluded that all these

generated keys have unique and random characteristics and therefore can be efficiently used for security provision.

#### 6.1.4 NIST Test Analysis of Novel Hash Algorithm

The message digest values obtained using novel hash algorithm are converted into binary form and NIST tests have been performed them. The results are summarized from Tables 6.4 to 6.8.

Table 6.4: Frequency Test results for Hash Algorithm

Hash Technique	P-values		
	p	World	Gurpreetkour
MD2	0.2888	0.7237	0.8597
MD5	0.3768	0.5959	0.4795
SHA-160	0.8744	0.8744	0.2059
SHA-256	0.4533	0.1498	0.4533
SHA-384	0.7595	0.2616	0.5403
SHA-512	0.2888	0.3768	0.5361
<b>Proposed Technique</b>	<b>0.8744</b>	<b>0.1138</b>	<b>0.8746</b>

Table 6.5: Binary Derivative Test results for Hash Algorithm

Hash Technique	P-values		
	p	World	Gurpreetkour
MD2	0.4849	0.5029	0.5049
MD5	0.5038	0.5042	0.4961
SHA-160		0.5110	0.5017
SHA-256	0.5006	0.5022	0.5014
SHA-384	0.5023	0.4997	0.5032
SHA-512	0.4980	0.4972	0.5021
<b>Proposed technique</b>	<b>0.4930</b>	<b>0.4974</b>	<b>0.5111</b>

Table 6.6: DFT Test results for Hash Algorithm

Hash Technique	P-values		
	p	World	Gurpreetkour
MD2	0.5164	0.8711	0.5164
MD5	0.3304	0.5164	0.5164
SHA-160	0.1000	0.4682	0.1000
SHA-256	0.3588	-	0.7308
SHA-384	0.0027	0.4537	0.7787
SHA-512	0.7456	0.6265	0.1233
<b>Proposed technique</b>	<b>0.4682</b>	<b>0.4688</b>	<b>0.4680</b>

Table 6.7: Approximate Entropy Test Results for Hash Algorithm

Hash Technique	P-values		
	p	World	gurpreetkour
MD2	0.6169	0.3499	0.5739
MD5	0.4108	0.8312	0.6896
SHA-160	0.8897	0.8018	0.6803
SHA-256	0.9963	0.7720	0.9080
SHA-384	0.9954	0.9915	0.9712
SHA-512	0.9841	0.9801	0.9965
<b>Proposed technique</b>	<b>0.5619</b>	<b>0.9163</b>	<b>0.8896</b>

Table 6.8: Maurer Test Results for Hash Algorithm

Hash Technique	P-Values		
	p	World	gurpreetkour
MD2	0.8981	0.9880	0.9664
MD5	0.9753	0.9985	0.9675
SHA-160	0.9914	0.9135	0.9818
SHA-256	0.9976	0.9690	0.9935
SHA-384	0.9836	0.9978	0.9993
SHA-512	0.9831	0.9600	0.9850
<b>Proposed technique</b>	<b>0.9943</b>	<b>0.9892</b>	<b>0.9926</b>

As clearly noticed from Table 6.4 to Table 6.8, the proposed technique performs better by passing the NIST criteria of generating a random message digest. Thus, it signifies its effectiveness as a Hash algorithm.

### 6.1.5 NIST Test Analysis of MAC formed using DNA-BRNG Key and novel Hash

The presented scheme is accessed on the basis of NIST tests of randomness and avalanche criteria. This is done for three different inputs values having varying lengths.

In order to validate the efficiency of our proposed algorithm, the NIST results of the proposed MAC scheme are compared with those of the existing ones. The key used for these algorithms is '3A54E26B', which is kept constant for all the traditional techniques.

The results are summarized in Tables 6.9 to Table 6.13.

Table 6.9: Frequency Test Results for DNA-BRNG based MAC

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.3768	0.3768	0.4795
HMAC MD5	0.8597	0.5959	0.8597
HMAC SHA-160	0.8744	1.0000	0.8744
HMAC SHA-256	0.2606	0.9005	0.0801
HMAC SHA-384	0.2207	0.1258	0.3074
HMAC SHA-512	0.7909	0.9296	0.9296
<b>Proposed Technique</b>	<b>0.8917</b>	<b>0.9744</b>	<b>0.9303</b>

Table 6.10: Binary Derivative Test Results for DNA-BRNG based MAC

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.4952	0.5126	0.5016
HMAC MD5	0.5129	0.4901	0.5149
HMAC SHA-160	0.5069	0.4924	0.5026
HMAC SHA-256	0.5046	0.5007	0.5040
HMAC SHA-384	0.5005	0.4964	0.4993
HMAC SHA-512	0.5026	0.5034	0.4987
<b>Proposed Technique</b>	<b>0.5240</b>	<b>0.5186</b>	<b>0.5112</b>

Table 6.11: DFT Test Results for DNA-BRNG based MAC

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.1443	0.0940	0.3304
HMAC MD5	0.8711	0.5164	0.0744
HMAC SHA-160	0.1468	0.4682	0.0295
HMAC SHA-256	0.4220	0.4220	0.1359
HMAC SHA-384	0.7787	0.7787	0.5121
HMAC SHA-512	0.3723	0.2561	0.6265
<b>Proposed Technique</b>	<b>0.8730</b>	<b>0.8068</b>	<b>0.6437</b>

Table 6.12: Approximate Entropy Test Results for DNA-BRNG based MAC

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.7464	0.7727	0.7310
HMAC MD5	0.4533	0.8983	0.8863
HMAC SHA-160	0.9288	0.8835	0.9883
HMAC SHA-256	0.8330	0.9440	0.9587
HMAC SHA-384	0.9817	0.9836	0.9865
HMAC SHA-512	0.9949	0.9891	0.9855
<b>Proposed Technique</b>	<b>0.9390</b>	<b>0.9893</b>	<b>0.9880</b>

Table 6.13: Maurer Test .Results for DNA-BRNG based MAC

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.9268	0.9528	0.9553
HMAC MD5	0.9831	0.9833	0.9951
HMAC SHA-160	0.9713	0.9600	0.9255
HMAC SHA-256	0.9912	0.9599	0.9705
HMAC SHA-384	0.9774	0.9909	0.9913
HMAC SHA-512	0.9865	0.9909	0.9765
<b>Proposed Technique</b>	<b>0.9967</b>	<b>0.9930</b>	<b>0.9968</b>

As clearly observed from Table 6.8 to Table 6.13, the proposed technique performs better by passing the NIST criteria of generating a random MAC. Thus, indicating its significance as a MAC Scheme.

#### 6.1.6 NIST Test Analysis of MAC formed using DNA-BBSG Key and novel Hash

In order to validate the efficiency of the proposed technique, the NIST results of the proposed MAC scheme are compared with those of the existing techniques.

The eight digit hexadecimal key used for these HMAC techniques is '3A54E26B', which is kept constant for all the schemes. The test results are summarized in tabular form in Table 6.14 to Table 6.18.

Table 6.14: Frequency Test Results for DNA-BBSG based MAC.

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.3768	0.3768	0.4795
HMAC MD5	0.8597	0.5959	0.8597
HMAC SHA-160	0.8744	1.0000	0.8744
HMAC SHA-256	0.2606	0.9005	0.0801
HMAC SHA-384	0.2207	0.1258	0.3074
HMAC SHA-512	0.7909	0.9296	0.9296
<b>Proposed Technique</b>	<b>0.8774</b>	<b>0.9389</b>	<b>0.9219</b>

Table 6.15: Binary Derivative Test Results for DNA-BBSG MAC.

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.4952	0.5126	0.5016
HMAC MD5	0.5129	0.4901	0.5149
HMAC SHA-160	0.5069	0.4924	0.5026
HMAC SHA-256	0.5046	0.5007	0.5040
HMAC SHA-384	0.5005	0.4964	0.4993
HMAC SHA-512	0.5026	0.5034	0.4987
<b>Proposed Technique</b>	<b>0.5160</b>	<b>0.5136</b>	<b>0.5092</b>

Table 6.16: DFT Test Results for DNA-BBSG MAC.

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.1443	0.0940	0.3304
HMAC MD5	0.8711	0.5164	0.0744
HMAC SHA-160	0.1468	0.4682	0.0295
HMAC SHA-256	0.4220	0.4220	0.1359
HMAC SHA-384	0.7787	0.7787	0.5121
HMAC SHA-512	0.3723	0.2561	0.6265
<b>Proposed Technique</b>	<b>0.8740</b>	<b>0.7798</b>	<b>0.6318</b>

Table 6.17: Approximate Entropy Test Results for DNA-BBSG MAC

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.7464	0.7727	0.7310
HMAC MD5	0.4533	0.8983	0.8863
HMAC SHA-160	0.9288	0.8835	0.9883
HMAC SHA-256	0.8330	0.9440	0.9587
HMAC SHA-384	0.9817	0.9836	0.9865
HMAC SHA-512	0.9949	0.9891	0.9855
<b>Proposed Technique</b>	<b>0.9403</b>	<b>0.9889</b>	<b>0.9974</b>

Table 6.18: Maurer Test Results for DNA-BBSG MAC

MAC Technique	P-values		
	g	Sodhi	unitedStates
HMAC MD2	0.9268	0.9528	0.9553
HMAC MD5	0.9831	0.9833	0.9951
HMAC SHA-160	0.9713	0.9600	0.9255
HMAC SHA-256	0.9912	0.9599	0.9705
HMAC SHA-384	0.9774	0.9909	0.9913
HMAC SHA-512	0.9865	0.9909	0.9765
<b>Proposed Technique</b>	<b>0.9987</b>	<b>0.9939</b>	<b>0.9993</b>

As it is observed from Table 6.14 to Table 6.18, the proposed scheme performs better by passing the NIST criteria of generating a random MAC. Thus, indicating its efficiency as a MAC technique.

### 6.1.7 NIST Test Analysis of MAC formed using DNA-LCG Key and novel Hash

The performance of the proposed scheme is analyzed on the basis of NIST tests of randomness and avalanche criteria. This is done for three inputs values having varying lengths. In order to certify the efficiency of our presented scheme, the NIST results of the proposed MAC scheme are compared with those of the traditional ones. The eight digit hexadecimal key used for these algorithms is '3A54E26B', which is kept constant throughout for all the traditional techniques. The results are summarized in Table 6.19 to Table 6.23.



Table 6.19: Frequency Test Results for DNA-LCG based MAC

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.3768	0.3768	0.4795
HMAC MD5	0.8597	0.5959	0.8597
HMAC SHA-160	0.8744	1.0000	0.8744
HMAC SHA-256	0.2606	0.9005	0.0801
HMAC SHA-384	0.2207	0.1258	0.3074
HMAC SHA-512	0.7909	0.9296	0.9296
<b>Proposed Technique</b>	<b>0.8884</b>	<b>0.9776</b>	<b>0.9289</b>

Table 6.20: Binary Derivative Test Results for DNA-LCG based MAC

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.4952	0.5126	0.5016
HMAC MD5	0.5129	0.4901	0.5149
HMAC SHA-160	0.5069	0.4924	0.5026
HMAC SHA-256	0.5046	0.5007	0.5040
HMAC SHA-384	0.5005	0.4964	0.4993
HMAC SHA-512	0.5026	0.5034	0.4987
<b>Proposed Technique</b>	<b>0.5191</b>	<b>0.5138</b>	<b>0.5121</b>

Table 6.21: DFT Test Results for DNA-LCG based MAC

MAC Technique	P-values		
	g	Sodhi	Unitedstates
HMAC MD2	0.1443	0.0940	0.3304
HMAC MD5	0.8711	0.5164	0.0744
HMAC SHA-160	0.1468	0.4682	0.0295
HMAC SHA-256	0.4220	0.4220	0.1359
HMAC SHA-384	0.7787	0.7787	0.5121
HMAC SHA-512	0.3723	0.2561	0.6265
<b>Proposed Technique</b>	<b>0.8763</b>	<b>0.7812</b>	<b>0.6192</b>

Table 6.22: Approximate Entropy Test Results for DNA-LCG based MAC

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.7464	0.7727	0.7310
HMAC MD5	0.4533	0.8983	0.8863
HMAC SHA-160	0.9288	0.8835	0.9883
HMAC SHA-256	0.8330	0.9440	0.9587
HMAC SHA-384	0.9817	0.9836	0.9865
HMAC SHA-512	0.9949	0.9891	0.9855
<b>Proposed Technique</b>	<b>0.9929</b>	<b>0.9889</b>	<b>0.9985</b>

Table 6.23: Maurer Test Results for DNA-LCG based MAC

MAC Technique	P-values		
	g	Sodhi	unitedstates
HMAC MD2	0.9268	0.9528	0.9553
HMAC MD5	0.9831	0.9833	0.9951
HMAC SHA-160	0.9713	0.9600	0.9255
HMAC SHA-256	0.9912	0.9599	0.9705
HMAC SHA-384	0.9774	0.9909	0.9913
HMAC SHA-512	0.9865	0.9909	0.9765
<b>Proposed Technique</b>	<b>0.9923</b>	<b>0.9916</b>	<b>0.9914</b>

As observed from Table 6.19 to Table 6.23, the proposed algorithm performs better by passing the NIST criteria of generating a random MAC. Thus, indicating its efficiency as a MAC technique.

## 6.2 Avalanche Test Analysis

The purpose of this test is to check the avalanche effect, which is a desirable property of the security keys. Wherein if the input is changed slightly the output changes significantly. It gives the percentage of bits flipped with a change in the input. It is a desirable property of security keys. The avalanche effect can be calculated using the formula given in equation (13)

$$Avalanche\ effect = \frac{No.of\ bits\ flipped\ in\ the\ sequence}{Total\ no.of\ bits\ in\ the\ sequence} \times 100 \quad (13)$$

**6.2.1 Avalanche Test Analysis for key Generated using DNA and BRNG**

The result of avalanche effect on changing the inputs is summarized in tabular form, where  $D_1$ ,  $D_2$ ,  $D_3$  represent the DNA sequences and  $B_1$ ,  $B_2$ ,  $B_3$  represent the BRNG sequences characterized by the seed values. The results are given in Tables 6.24, 6.25 and 6.26 respectively considering each case separately.

The test is performed on three sets of DNA and BRNG sequences:

*Case 1:* In the initial set, two security keys are generated through two DNA sequences while keeping the same BRNG sequence.

*Case 2:* The second set involves generation of two security keys through the same DNA and two BRNG sequences.

*Case 3:* In the third set, two security keys are generated through two DNA and BRNG sequences.

The main aim here is to know the amount of randomness the proposed technique produces on changing the input. The change in the output with a change in input is represented in the form of percentage; more the percentage of avalanche effect higher is the efficiency.

Table 6.24: Avalanche test analysis: Case 1 (DNA-BRNG)

DNA Sequences ( $D_n$ )	Seed Value	Bernoulli Random Sequence ( $B_n$ )	Key Generated $K = D_n \text{ xor } B_n$	Avalanche Result of Key (K)	
				No. of bits flipped	Avalanche Effect
$D_1$	1231	$B_1$	10110111000000110100111101110110010100000001 00010110111010011001000001111000010101010000 10001100110011100100011011101010101111110000 01100111010010000101011001100101100011100011 0001010101100110000100011110011111011110111 011110000111011110000111011110010110	58	22.65 %
$D_2$			10110001000000010101100001100011010101000000 00000111111110001010000001011000001101010110 10001010110110110100010011101100101110010001 01010111010010000011011001100101110011110110 00010111011001000000000011100111111010110111 010110000011011011110110000010010010		

\* Refer Table 5.1 for  $D_1, D_2, D_3$  and Table 5.2 for  $B_1, B_2, B_3$   
\*\* Different DNA sequences - Same BRNG sequence

Table 6.25: Avalanche test analysis: Case 2 (DNA-BRNG)

DNA Sequence (D <sub>n</sub> )	Seed Value	Bernoulli Random Sequence (B <sub>n</sub> )	Key Generated K= D <sub>n</sub> xor B <sub>n</sub>	Avalanche Result of Key (K)	
				No. of bits flipped	Avalanche effect
D <sub>1</sub>	1231	B <sub>1</sub>	1011011100000011010011110111011001010 0000001000101101110100110010000011110 0001010101000010001100110011100100011 0111010101011111100000110011101001000 0101011001100101100011100011000101010 1100110000100011110011111101111011101 1110000111011110000111011110010110	124	48.43 %
	101355	B <sub>2</sub>	0001001010010001010001001000000101110 0100011001111101100111011101110010001 0111110100010110110001001010110100110 0110011011010100101011100110111110001 111000110111110010011111111101001001 1111011100000010110010000100001001101 1100100000100101010111100001010001		

\*Refer Table 5.1 for D<sub>1</sub>, D<sub>2</sub>, D<sub>3</sub> and Table 5.2 for B<sub>1</sub>, B<sub>2</sub>, B<sub>3</sub>  
 \*\*Same DNA sequences - Different BRNG sequences

Table 6.26: Avalanche test analysis: Case 3 (DNA-BRNG)

DNA Sequence (D <sub>n</sub> )	Seed Value	Bernoulli Random Sequence (B <sub>n</sub> )	Key Generated K= D <sub>n</sub> xor B <sub>n</sub>	Avalanche Result of Key (K)	
				No. of bits flipped	Avalanche Effect
D <sub>1</sub>	1231	B <sub>1</sub>	1011011100000011010011110111011001010000000 1000101101110100110010000011110000101010100 0010001100110011100100011011101010101111110 0000110011101001000010101100110010110001110 0011000101010110011000010001111001111110111 10111011110000111011110000111011110010110	58	22.65 %
D <sub>2</sub>	101355	B <sub>2</sub>	0001010010010011010100111001010001110110001 0001011111101111110111110011001011001010000 1110110111001111100100111011001011101011110 100111111011111000110000011011110011011110 1010101001101111100110010000011001000010010 10011010100100100100000100110111101010101		

\*Refer Table 5.1 for D<sub>1</sub>, D<sub>2</sub>, D<sub>3</sub> and Table 5.2 for B<sub>1</sub>, B<sub>2</sub>, B<sub>3</sub>  
 \*\*Different DNA sequences – Different BRNG sequences

The avalanche test results clearly indicate that a slight change in the inputs leads to a significant change in the output. The higher the percentage value of avalanche effect, the more is the uniqueness of the key. As the seed value may vary according to the user’s choice and the DNA

being unique for every individual, it can be assumed that the security system built has higher efficiency and is less susceptible to attacks.

**6.2.2 Avalanche Test Analysis for key generated using DNA and BBSG**

The test is performed on three sets of DNA and BBSG sequences:

*Case 1:* In the initial set, two security keys are generated through two DNA sequences while keeping the same BBSG sequence.

*Case 2:* The second set involves generation of two security keys through the same DNA sequence and two BBSG sequences.

*Case 3:* In the third set, two security keys are generated through two DNA and BBSG sequences.

The main concern here is to calculate the amount of randomness the proposed technique produces on a change in the inputs. The result are summarized in tabular form, where  $D_1, D_2, D_3$  represent the DNA sequences and  $B_1, B_2, B_3$  represent the BBSG sequences. The results are given in Table 6.27, 6.28 and 6.29 considering each case separately.

Table 6.27: Avalanche test analysis: Case 1 (DNA-BBSG)

DNA Sequence ( $D_n$ )	Seed Value	BBSG Sequence ( $B_n$ )	Key Generated $K = D_n \text{ xor } B_n$	Avalanche Result of key (K)	
				No. of bits flipped	Avalanche Effect
$D_1$	101355	$B_1$	1100101000010011101111001000100001110100111 1001011011100110011111001010100110111001010 1100011001001000100001010010100000010000010 1000011101001100011101010011011010111011001 0110011001001100100001110011110011000110110 11101100100101101111011101110001001100110	58	22.65 %
$D_2$			1100110000010001101010111001110101110000111 0001111001101110111001001011100110001001011 010001111001101110001011010100110010001110 1010000101001100011110010011011010110011000 0011011001101100101001100010110011000110100 1110110110010100111110011111010101100010		

\*Refer Table 5.1 for  $D_1, D_2, D_3$  and Table 5.3 for  $B_1, B_2, B_3$   
\*Different DNA sequences – Same BBSG sequences

Table 6.28: Avalanche test analysis: Case 2 (DNA-BBSG)

DNA Sequence ( $D_n$ )	Seed Value	BBSG Sequence ( $B_n$ )	Key Generated $K= D_n \text{ xor } B_n$	Avalanche Result of Key (K)	
				No. of bits flipped	Avalanche Effect
D <sub>1</sub>	101355	B <sub>1</sub>	110010100001001110111100100010000111010011 110010110111001100111110010101001101110010 101100011001001000100001010010100000010000 010100001110100110001110101001101101011101 100101100110010011001000011100111100110001 101101110110010010110111101110111000100110 0110	118	46.09 %
	101357	B <sub>2</sub>	101111000001101110011111100010101010100101 000000011011111111110011000100110110011011 010101010111111000101101110000001010011011 01111001111000101010001101111011101000100 01001010000001111111010011100111011001101 00011110001001001110110100011011001111101 1101		

\*Refer Table 5.1 for D<sub>1</sub>, D<sub>2</sub>, D<sub>3</sub> and Table 5.3 for B<sub>1</sub>, B<sub>2</sub>, B<sub>3</sub>  
\*\*Same DNA sequences – Different BBSG sequences

Table 6.29: Avalanche test analysis: Case 3 (DNA-BBSG)

DNA Sequence ( $D_n$ )	Seed Value	BBSG Sequence ( $B_n$ )	Key Generated $K= D_n \text{ xor } B_n$	Avalanche Result of Key (K)	
				No. of bits flipped	Avalanche Effect
D <sub>1</sub>	101355	B <sub>1</sub>	110010100001001110111100100010000111010 011110010110111001100111110010101001101 110010101100011001001000100001010010100 000010000010100001110100110001110101001 101101011101100101100110010011001000011 100111100110001101101110110010010110111 1011101110001001100110	120	46.87 %
D <sub>2</sub>	101357	B <sub>2</sub>	101110100001100110001000100111111010110 101010001011111101110111111000110110111 111011001101010001111101111101111000001 100011010111111010010001010100010111111 01110100000001011111000001011111000011 000101011001101000011100010110011111101 0100011101100011011001		

\*Refer Table 5.1 for D<sub>1</sub>, D<sub>2</sub>, D<sub>3</sub> and Table 5.3 for B<sub>1</sub>, B<sub>2</sub>, B<sub>3</sub>  
\*\*Different DNA sequences – Different BBSG sequences

### 6.2.3 Avalanche Test Analysis for key generated using DNA and LCG

The test is performed on three sets of DNA and LCG sequences:

*Case 1:* In the initial set, two security keys are generated through two DNA sequences while keeping the same LCG sequence.

*Case 2:* The second set involves generation of two security keys through the same DNA sequence and two LCG sequences.

*Case 3:* In the third set, two security keys are generated through two DNA and LCG sequences.

The avalanche effect is calculated for each of the three sets and results are tabulated separately for each technique. This purpose here is to find out the amount of randomness the proposed technique produces on changing the input.

The result of avalanche effect on changing the inputs is summarized in tabular form, where  $D_1$ ,  $D_2$ ,  $D_3$  represent the DNA sequences and  $B_1$ ,  $B_2$ ,  $B_3$  represent the LCG sequences characterized by the seed values. The results are given in Tables 6.30, 6.31 and 6.32 considering each case separately.

Table 6.30: Avalanche Test Analysis: Case 1 (DNA-LCG)

DNA Sequences ( $D_n$ )	Seed Value	LCG Sequence ( $L_n$ )	Key Generated $K = D_n \text{ xor } L_n$	Avalanche result of $K_n$ (K)	
				No. of Bit Flipped	Avalanche Effect
$D_1$	47594118	$L_1$	0101010001000110001000111000110001011 0011100100110111111010011011110100000 0010011111010100000011101010100100001 1110101101110111101110111011111101011 1001111101011101000110011101000101100 0100110010000001000010110110000111101 0010000101001110100000111100000010	58	22.65 %
$D_2$			01010010010001000011101001001100101011 1011101100010101110010111101110101000 0011111111001100000101101111110100000 1110100001110100101100100011111101011 1111111101011101010110001000000101000 0100100010100011000010110110100111101 1010000001001011010001100000000110		

\* Refer Table 5.1 for  $D_1$ ,  $D_2$ ,  $D_3$  and Table 5.4 for  $L_1$ ,  $L_2$ ,  $L_3$   
\*\* Different DNA sequences - Same LCG sequence

Table 6.31: Avalanche Test Analysis: Case 2 (DNA-LCG)

DNA Sequences (D <sub>n</sub> )	Seed Value	LCG Sequence (L <sub>n</sub> )	Key Generated K= D <sub>n</sub> xor L <sub>n</sub>	Avalanche result of Key (K)	
				No. of Bits Flipped	Avalanche Effect
D <sub>1</sub>	47594118	L <sub>1</sub>	01010100010001100010001110001100010 11001110010011011111101001101111010 00000010011111010100000011101010100 10000111101011011101111011101110111 11101011100111110101110100011001110 10001011000100110010000001000010110 11000011110100100001010011101000001 11100000010	123	48.04 %
	47594122	L <sub>2</sub>	01010101010001010010011001010000011 11100101111000010010000001010110011 00100110110011001010110010111011010 00001011001110000011110000010001001 01110111000101010110111001101111000 00000001110010110111100001101000010 11001001100011000101110000011011100 01010101101		

\* Refer Table 5.1 for D<sub>1</sub>, D<sub>2</sub>, D<sub>3</sub> and Table 5.4 for L<sub>1</sub>, L<sub>2</sub>, L<sub>3</sub>  
 \*\* Same DNA Sequence - Different LCG sequence

Table 6.32: Avalanche Test Analysis: Case 3 (DNA-LCG)

DNA Sequences (D <sub>n</sub> )	Seed Value	LCG Sequence (L <sub>n</sub> )	Key Generated K= D <sub>n</sub> xor L <sub>n</sub>	Avalanche result of Key (K)	
				No. of Bits Flipped	Avalanche Effect
D <sub>1</sub>	47594118	L <sub>1</sub>	0101010001000110001000111000110001011 0011100100110111111010011011110100000 0010011111010100000011101010100100001 1110101101110111101110111111101011 1001111101011101000110011101000101100 0100110010000001000010110110000111101 00100001010011110100000111100000010	117	45.70 %
D <sub>2</sub>	47594122	L <sub>2</sub>	0101001101000111001100010100010101111 0001010110100110101000110011100111010 0111010011010010110100111110000000011 1100110100001100000011011100101110111 0111010101101110001011100101000000011 0010100111000011101000010110110011000 0100010011000100011111010110101001		

\* Refer Table 5.1 for D<sub>1</sub>, D<sub>2</sub>, D<sub>3</sub> and Table 5.4 for L<sub>1</sub>, L<sub>2</sub>, L<sub>3</sub>  
 \*\* Different DNA Sequences - Different LCG sequence



### 6.2.4 Avalanche Test Analysis for Novel Hash Algorithm

A message digest is designed in order to protect the integrity of a piece of data and to considerably detect any kind of changes or alteration in any part of the message. Also, one message digest specifically represents particular data content and thus it can reference a change made intentionally or unintentionally. Thus, there must be a change in a particular message digest following a change in the data file. This test has been applied by altering a single character of the input value, considering existing techniques and comparing the results with the presented one. The Avalanche Test results are summarized in Table 6.33.

Table 6.33: Avalanche Test analysis for Novel Hash

Original Input	Altered Input	Avalanche Effect (%)						
		MD2	MD5	SHA-160	SHA-256	SHA-384	SHA-512	Proposed Hash
P	G	50.00	46.87	46.45	52.43	51.82	49.80	46.90
World	Worlk	52.75	45.31	52.50	46,87	51.30	50.00	51.87
gurpreetkour	Gurpreetkous	46.09	54.68	52.50	45.31	53.90	50.39	49.87

It can be clearly noticed that the recommended technique performs well under this criteria too, thus demonstrating its efficiency.

Further the technique is evaluated on another factor i.e. throughput which is explained as the maximum output that can be produced by a system within the specified resources such as bandwidth. It is desirable to make the best use of resources by reducing the amount of redundant data transmission. Throughput can measure by using Equation 14.

$$Throughput = \frac{\text{Data without overheads}}{\text{Total data transmitted}} \quad (14)$$

Proposed technique produces an output of 160 bits given the input data of variable length. When compared to other hash algorithms of SHA family, the proposed technique enhances the throughput of the system significantly.

### 6.2.5 Avalanche Test Analysis for DNA-BRNG based MAC

The objective of MAC is to protect the integrity of the data and to detect any alteration in the message. Also, a particular MAC is in accordance with particular data content and thus it can

significantly indicate a change in the data. Thus, there is a change in a particular MAC value following a change in the data file. The test has been applied by altering a single character of the input value, it is applied on the traditional techniques and the outcomes are compared with the ones obtained for the presented scheme. The Avalanche Test results are summarized in Table 6.34.

Table 6.34: Avalanche Test analysis for DNA-BRNG based MAC

Original Input	Altered Input	No. of bits flipped	Avalanche Effect (%)
g	P	70	43.75
Sodhi	Sodhb	83	51.87
Unitedstates	unitedstraten	87	54.37

It can be clearly noticed that the recommended technique performs well under this criteria too, thus demonstrating its efficiency.

### 6.2.6 Avalanche Test Analysis for DNA-BBSG based MAC

To apply this test a single character of the input value is altered, and avalanche effect is calculated. The comparison is done using the same procedure for other schemes also. The Avalanche Test results are summarized in Table 6.35.

Table 6.35: Avalanche Test analysis DNA-BBSG based MAC

Original Input	Altered Input	No. of bits flipped	Avalanche Effect (%)
g	p	79	49.37
Sodhi	Sodhb	81	50.62
unitedstates	unitedstraten	76	47.50

It can be clearly observed that the proposed technique performs well under this criteria too, thus demonstrating its efficiency.

### 6.2.7 Avalanche Test Analysis for DNA-LCG based MAC

This test has been applied by altering a single character of the input value, it is applied on the traditional techniques and the outcomes are compared with the ones obtained for the presented technique. The Avalanche Test results are summarized in Table 6.36.

Table 6.36: Avalanche Test Results for DNA-LCG based MAC

Original Input	Altered Input	No. of bits flipped	Avalanche Effect (%)
g	p	88	51.87
Sodhi	Sodhb	79	49.37
unitedStates	unitedstraten	80	50.00

It is observed that the recommended technique performs well under this criteria too, thus demonstrating its efficiency.

### 6.3 Network Attack Analysis

The increased complexity of the technique makes it highly resistive towards various network attacks on data integrity.

#### 6.3.1 Network Attack Analysis of DNA-BRNG based MAC

A brief summary analyzing the behavior of the MAC technique framed using DNA-BRNG key and a novel Hash algorithm is presented in Table 6.37.

Table 6.37: Resistance of DNA-BRNG based MAC against Attacks

Network Attacks on integrity	Preventive Features
Salami attacks	As observed from the avalanche test analysis, a small change in the input results in a major change in the output. Hence, even the minute modification in the data would be detected.
Data diddling Attacks	Since the proposed scheme uses biological characteristics to frame the secret key, therefore it is not possible for the data to be modified by an unauthorized party.
Man-in-the-middle attacks	The proposed technique is hash based, thus it is highly resistive towards any changes in the data by an unauthorized party.
Seed Attacks	Keys generated using only the random number generator outputs are susceptible towards seed attacks, the key used in the proposed MAC is a result of fusion of DNA and random number generator output, thus increasing its resistance towards seed attacks.

The proposed MAC algorithm is highly resistive towards the attacks, thus increasing its applicability in a data sensitive environment.

#### 6.3.2 Network Attack Analysis DNA-BBSG based MAC

An analysis of the behavior of the MAC technique framed using DNA-BBSG key and a novel Hash algorithm is presented in Table 6.38.

Table 6.38: Resistance of DNA-BBSG based MAC against Attacks

<b>Network Attacks on Integrity</b>	<b>Preventive Features</b>
Salami attacks	As observed from the avalanche test analysis, a minor change in the input results in a major change in the output. Hence, the technique is resistive towards salami attacks.
<b>Data diddling Attacks</b>	Since the presented algorithm uses biological characteristics of DNA to frame the secret key, therefore it is impossible for the data to be modified by an unauthorized user.
<b>Man-in-the-middle attacks</b>	The proposed technique is hash based, thus it is highly resistive towards any unauthorized modifications in the transmitted data.
<b>Seed Attacks</b>	Keys generated using the random number generator outputs are susceptible to seed attacks, the key used in the proposed MAC is a fusion of DNA and random number generator output, thus increasing its resistance towards seed attacks.

The proposed MAC technique has higher complexity, which makes it highly resistive towards various attacks on integrity, thus increasing its applicability in networks demanding security.

### 6.3.3 Network Attack Analysis DNA-LCG based MAC

A summarized study of the behavior of the MAC technique framed using DNA-LCG key and a novel Hash algorithm is presented in Table 6.39.

Table 6.39: Resistance of DNA-LCG based MAC against Attacks

<b>Network Attacks on Integrity</b>	<b>Preventive Features</b>
Salami attacks	As observed from the avalanche test analysis, a small change in the input results in a major change in the output. Hence, even the minute modification in the data would be detected.
<b>Data diddling Attacks</b>	Since the proposed scheme uses biological characteristics to frame the secret key, therefore it is not possible for the data to be modified by an unauthorized party.
<b>Man-in-the-middle attacks</b>	The proposed technique is hash based, thus it is highly resistive towards any unauthorized alterations in the transmitted data.
<b>Seed Attacks</b>	Keys generated using only the random number generator outputs are susceptible towards seed attacks; the key used in the proposed MAC is a result of fusion of DNA and random number generator output, thus increasing its resistance towards seed attacks.

The proposed MAC algorithm is complex and therefore is highly resistive towards various attacks on integrity, thus increasing its applicability in a data sensitive network.

**CHAPTER 7****CONCLUSION AND FUTURE SCOPE**

---

Security of the data becomes most important concern in order to avoid an unauthorized or unintended access. In cryptography, a Message Authentication Code (MAC) is a short piece of information used to confirm that the message came from the stated sender and has not been altered in transit. A MAC algorithm, accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC.

In Phase-1, a new scheme is recommended which is formed by the combination of biometric features and random number generator outputs. Here DNA and Random Number generator sequences are combined to generate a security key of 256 bits. The suggested technique makes the use of unique biological characteristics along with pseudo-random generator to build a novel key generator. DNA being a unique characteristic of an individual when used in collaboration with the RNG sequence yields better results in terms of security. If used separately, biometrics may prove to be a weak authentication technique as the DNA of an individual can be obtained without making the person aware. These keys can be applied in security systems for example in confidential areas like nuclear plants, banks, military base, etc.

In Phase-2, a hash algorithm is developed which uses the basic structure of SHA-160 along with an integrated function, this is done to increase complexity. This novel hash algorithm along with the secret keys generated in Phase-1 is used in Phase-3 to frame a MAC.

The three keys, the hash algorithm and the three MAC schemes are tested using NIST tests of random numbers and avalanche criteria. Further, analysis is carried out on the basis of various network attacks on data integrity. It is observed that this algorithm has an improved performance, as compared to the existing ones as the complexity of the system is increased. Hence, the proposed scheme finds its applicability in the data sensitive environment. As a future work, other signals like audio, video etc. can be used as inputs for this algorithm other than DNA. The algorithm can also be extended for longer biometric security keys to enhance the strength of security.

**REFERENCES**

- [1] Honguntikar V, and Biradar G. S, "Optimization Techniques Incorporating Evolutionary Model in Wireless Sensor Network. A Survey", *Journal of Computer Engineering*, vol.16, no. 5, 2014, pp.19-24.
- [2] Stallings W, "Cryptography and Network Security," New York, NY: Pearson Education, 2006.
- [3] Dargie W, and Poellabauer C, "Fundamentals of Wireless Sensor Networks".
- [4] Forouzan B. A, and Fegan S. C, "Data Communications and Networking".
- [5] Eastlake D, and Hansen T, "US Secure Hash Algorithms (SHA and HMAC-SHA)", RFC, Network Working Group, 2006.
- [6] Chang H. T, Kuo C. J, Lo N. W, and Wei Z. L, "DNA Sequence Representation and Comparison Based on Quaternion Number System", *International Journal Of Advanced Chemical Science and Applications*, vol. 3, no. 11, 2012, pp. 39-46.
- [7] Goldberger A. L, Amaral L, Glass L, Hausdorff J. M, Ivanov P. C, Mark R. G, Mietus J. E, Moody G. B, Peng C. K, Stanley H. E. *Physio Bank, Physio Toolkit and Physio Net: Components of a New Research Resource for Complex Physiologic Signals Circulation*, 2000.
- [8] Ensembl Genome Browser. <http://www.ensembl.org/index.html>
- [9] NCBI databases. <http://www.ncbi.nlm.nih.gov/Entrez>
- [10] Rukhin A, Soto J, Nechvatal J, Smid M, Barker E, Leigh S, Levenson M, Vange M.I, Banks D, Heckert A, Dray J, Vo S, and Bassham III L.E, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", 2010.
- [11] Chen B, and Chandran V, "Biometric Based Cryptographic Key Generation from Faces", *Proceedings of the IEEE, 9<sup>th</sup> Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, 2007, pp. 394-401.
- [12] Hao F, Anderson R, and Daugman J, "Combining Cryptography with Biometrics Effectively", *IEEE Transactions on Computers*, vol. 55, no. 9, 2006, pp. 1081-1088.
- [13] Chen I.T, Tsai J.M, Tzeng J, "Audio Random Number Generator and its Application", *Proceedings of the IEEE, International Conference on Machine Learning and Cybernetics*, vol. 4, 2011, pp. 1678-1683.
- [14] Wei W, and Jun Z, "Image Encryption Algorithm Based on the Key Extracted from Iris Characteristics", *Proceedings of the IEEE, 14<sup>th</sup> International Symposium on Computational Intelligence and Informatics*, 2013, pp. 169-172.
- [15] Garcia B. H. A, Alarcon A. V, and Starostenko O, "A Wavelet-Based 128-bit Key Generator Using Electrocardiogram Signals", *Proceedings of the IEEE, 52<sup>nd</sup> International Midwest Symposium on Circuits and Systems*, 2009, pp. 644-647.

- [16] Hedayatpour S, and Chuprat S, “Hash Functions-based Random Number Generator with Image Data Source”, Proceedings of the IEEE, Conference on Open Systems, 2011, pp. 69-73.
- [17] Ying L, Jing Y, Shu W, and Xiao L, “Design of A Random Number Generator from Fingerprint”, Proceedings of the IEEE, International Conference on Computational and Information Sciences, 2010, pp. 278-280.
- [18] Khokher R, and Singh R. C, “Generation of Security Key using ECG Signal”, Proceedings of the IEEE, International Conference on Computing, Communication and Automation, 2015, pp. 895-900.
- [19] Liu L. W. X, Yuan S, and Xiao P, “A Novel Key Generation Cryptosystem Based on Face Features”, Proceedings of the IEEE, 10<sup>th</sup> international Conference on Signal Processing, 2010, pp. 1675 – 1678.
- [20] Pal D, and Ghoshal N, “Secured Data Transmission Through Audio Signal”, Proceedings of the IEEE, 9<sup>th</sup> International Conference on Industrial and Information Systems, 2014, pp. 1-5.
- [21] Varma P. S, and Raju K. G, “Cryptography Based on DNA Using Random key Generation Scheme”, International Journal of Science Engineering and Advance Technology, vol. 2, no. 7, 2014.
- [22] Van D. H, and Thuc N. D, “A Privacy Preserving Message Authentication Code”, Proceedings of the IEEE, 5<sup>th</sup> International Conference on IT Convergence and Security (ICITCS), 2015, pp. 1-4..
- [23] Abduljabbarl Z. A, Jinl H, Zoul D, Yassin A. A, Hussien Z. A, and Hussainl M. A, “An Efficient and Robust One-Time Message Authentication Code Scheme Using Feature Extraction of Iris in Cloud Computing”, International Conference on Cloud Computing and Internet of Things, 2014.
- [24] Lan J, Zhou J. and Liu X, “ An area-efficient implementation of a Message Authentication Code (MAC) algorithm for cryptographic systems”, Proceedings of the IEEE, 10<sup>th</sup> Conference TENCON, 2016, pp. 1977 – 1979.
- [25] Verma S, and Prajapati G. S, “Robustness and security enhancement of SHA with modified message digest and larger bit difference”, Proceedings of the IEEE, Symposium on Colossal Data Analysis and Networking (CDAN), 2016, pp. 1-5.
- [26] NCBI Genbank. <http://www.ncbi.nlm.nih.gov/nuccore/3327045?report=genbank>
- [27] NCBI Genbank. <http://www.ncbi.nlm.nih.gov/nuccore/20380066?report=genbank>
- [28] NCBI Genbank. <http://www.ncbi.nlm.nih.gov/nuccore/33874586?report=genbank>

## APPENDIX 1

## S-BOXES USED IN THE PROPOSED HASH TECHNIQUE

In the presented work, 8 Substitution boxes have been used as a part of the 'f' function, which is applied in the existing SHA-160 Algorithm. The S-boxes are used in order to convert the expanded 48 bit data into 32 bit data [2]. The eight S-boxes used are shown in Fig. A-1.

S <sub>1</sub>	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S <sub>2</sub>	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S <sub>3</sub>	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S <sub>4</sub>	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S <sub>5</sub>	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S <sub>6</sub>	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S <sub>7</sub>	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S <sub>8</sub>	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Figure A-1: S-boxes [2]



**APPENDIX 2****PROPOSED WORK PLAN WITH TIME LINE**

The proposed work plan has been completed in a time span of four months. The description of work done has been framed in the table below:

<b>Week</b> ► <b>Month</b> ▼	<b>Week 1</b>	<b>Week 2</b>	<b>Week 3</b>	<b>Week 4</b>
<b>January</b>	Understanding the Concept of Hash Schemes and Message Authentication Codes.	Studying different Research Papers on SHA-160.	Formulation of a Novel Hash Algorithm.	Testing of The Novel Hash on NIST Tests.
<b>February</b>	Writing of A Paper on a Novel Hash Algorithm for Enhanced Security.	Formulation of A Novel MAC using DNA-BRNG Key and Novel Hash Algorithm.	Testing of The Novel MAC on NIST Tests and various Network Attacks.	Writing of a Paper on MAC using DNA-BRNG Key and Novel Hash Algorithm.
<b>March</b>	Formulation of a Novel MAC using DNA-BBSG Key and Novel Hash Algorithm.	Testing of the Novel MAC on NIST Tests and Various Network Attacks.	Writing of a Paper on MAC Using DNA-BBSG Key and Novel Hash Algorithm.	Formulation of a MAC using DNA-LCG Key and Novel Hash Algorithm.
<b>April</b>	Testing of the Novel MAC on NIST Tests and various Network Attacks.	Writing of a Paper on Novel MAC using DNA-LCG Key and Novel Hash Algorithm.	Report Writing.	Report Writing.

## **APPENDIX 3**

### **AUTOBIOGRAPHY**

**Gurpreet Kour Sodhi** is currently pursuing M.Tech in Electronics and Communication Engineering from Lovely Professional University, Phagwara with Wireless communication as specialization. Her area of interest includes, Wireless Sensor Networks, Cryptography Schemes, different Hashing Algorithms and Security. Her research area includes - ‘Enhancing and Maintaining Security in Wireless Communication Systems’ and ‘Networks’. She is working in this field since 2015.

## **APPENDIX 4**

### **PUBLICATIONS**

1. DNA and Bernoulli Random Number Generator Based Security Key Generation Algorithm. (Accepted)
2. DNA and Blum Blum Shub Random Number Generator Based Security Key Generation Algorithm. (Accepted)
3. DNA and LCG Based Security Key Generation Algorithm. (Accepted)
4. An efficient Hash Algorithm to Preserve Data Integrity. (Under Review)
5. A competent MAC Scheme using DNA-BRNG Key and a Novel Hash Algorithm. (Under Review)
6. An efficacious MAC technique designed using DNA-BBSG Key and a novel Hash Algorithm. (Communicated)
7. Implementation of MAC Using DNA-LCG and A Novel Hash Algorithm. (Communicated)