JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

JSCDM

Vol. 5 No. 2 (2024) 245-263
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Enhancing Security and Randomness of DNA Cryptosystem Generated by Using Mealy Machine

Gaverchand K[1], Venkatesan R[2]*, Kavikumar Jacob[3], Yasmin A[4]

[1] *Department of Mathematics, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, 603203, Tamil Nadu, INDIA*
[2] *Department of Mathematics, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, 603203, Tamil Nadu, INDIA*
[3] *Department of Mathematics and Statistics, Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia, Pagoh, 86400, Johor, MALAYSIA*
[4] *Department of Mathematics, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, 603203, Tamin Nadu, INDIA*

*Corresponding Author: venkater1@srmist.edu.in
DOI: https://doi.org/10.30880/jscdm.2024.05.02.018

Article Info

Received: 17 June 2024
Accepted: 27 October 2024
Available online: 18 December 2024

Keywords

Encryption, decryption, mealy machine, DNA computing

Abstract

In the current digital landscape, data security is paramount due to the increasing threats posed by hackers. The Deoxyribonucleic Acid (DNA) cryptosystem offers a secure method of data transmission by converting plaintext (PT) into ciphertext (CT) using a DNA encoding table. This paper proposes a novel DNA cryptographic technique that integrates a randomly generated Mealy machine with DNA encoding to bolster security. Initially, a 256-bit secret key is generated based on the receiver's credentials. Detailed explanations of key generation, encryption, and decryption processes are provided, along with illustrative examples. Extensive testing, including frequency distribution, resistance to security attacks, and comparative analysis, affirms the effectiveness of this model. Additionally, the randomness of the CT is validated by the National Institute of Standards and Technology (NIST) test suite, which yields an average p-value of 0.72, surpassing existing schemes. Furthermore, the system achieves an average avalanche effect of 79%, ensuring robust security. These results demonstrate the superior security and performance of the proposed cryptosystem compared to existing DNA-based techniques.

1. Introduction

In modern technology, securing communication and the transfer of information over the internet has become a critical issue. Security measures are essential in today's era due to the transmission of sensitive information through insecure channels. Intruders exploit vulnerabilities by hacking systems and gaining unauthorized access to steal information. The most effective way to mitigate such threats and attacks is to develop strong and secure algorithms to safeguard crucial data. Cryptography, a method used to create encryption and decryption techniques, emerges as a cornerstone solution in this endeavor. It plays a vital role in ensuring the security of confidential data from intruders while simultaneously ensuring data integrity throughout transmission and storage processes. The benefits of cryptography extend beyond mere protection, encompassing the fostering of trust in digital transactions, guaranteeing compliance with regulatory standards, and facilitating secure global connectivity.

Moreover, cryptography enhances data privacy, enabling individuals and organizations to communicate securely across diverse networks while effectively mitigating risks associated with cyber threats. Its

indispensable role in the modern technological landscape is reinforced by its ability to empower individuals and organizations, preserve autonomy, foster innovation, and promote a resilient digital ecosystem.

Cryptography is broadly categorized into two major types, symmetric and public key cryptography, each offering distinct advantages and applications in various scenarios. Symmetric key cryptosystems, known for their simplicity and efficiency, are well-suited for secure communication within closed networks or among trusted parties. Conversely, public key cryptosystems provide a robust mechanism for secure communication over public channels, enabling the secure exchange of information between entities without the need for prior establishment of shared information. While traditional cryptographic techniques have been effective in securing digital communication for decades, the rapid progression of technology has driven the need for the formulation of more intricate and robust cryptographic algorithms. Modern cryptographic algorithms such as RSA, AES, DES, and SHA256 employ complex mathematical principles and encryption methodologies to withstand sophisticated attacks and ensure the confidentiality, integrity, and authenticity of data in diverse environment

DNA cryptography presents a revolutionary approach to data security, emerging as a promising alternative to conventional cryptographic algorithms. With heightened security levels and inherent randomness, it offers robust protection against cyber threats in modern cryptographic systems. By utilizing coding rules and mathematical operations, DNA cryptosystems minimize processing power requirements, resulting in efficient algorithms with time complexities typically at O(n). The utilization of DNA codons, comprising adenine (A), guanine (G), cytosine (C) and thymine (T), allows for the transformation of confidential data into DNA sequences or strings, offering a unique and potentially more secure method for data encryption and decryption. As technology evolves, DNA cryptography possesses the capability to significantly bolster the security and integrity of data exchange across networks in the digital age, addressing emerging security challenges and fostering confidence in digital transactions. Its interdisciplinary nature fosters collaboration across various fields, driving innovation and enabling the development of cutting-edge solutions in data security.

This paper is organized as follows: In Section 2, we discuss the literature survey related to cryptography, and in Section 3, some of the preliminaries are discussed. Section 4 unveils the proposed scheme with mathematical formulation and illustration. Section 5 discusses different analyses and attacks. We conclude the proposed model in Section 6.

2. Literature Survey

Most of the researchers focuses on DNA cryptography to attain the secure, safe, efficient and randomized cryptographic algorithms. [3, 7, 8, 17, 21, 24, 25, 26, 35, 38, 39, 43] Few of the literature related to DNA computing and cryptosystem is discussed.

Zhang et al. [2] proposed a dynamic scheme combining Block-Cipher and Index methods for encrypting data derived from DNA sequences, demonstrating a large key space and high encryption efficiency. Biswas et al. [6] propose a novel DNA cryptographic method that integrates complex coding of DNA with an asymmetry cryptosystem, enhancing data secrecy by encrypting PT chunks and merging them with random DNA sequences generated using the Fibonacci series. Elhadad [9] introduced a framework employing DNA-based re-encryption techniques for ensuring secure cloud data sharing, showing efficiency through experimental evaluations. Siarry et al. [10] enhanced PT security using index-based permutation and DNA sequence operations. Wang et al. [11] developed a data concealment model utilizing DNA sequences and recombinant DNA techniques for secure information transmission, employing a single mapping rule table. Reddy et al. [15] presented a bio-inspired DNA cryptographic system utilizing genetic encoding principles, Bidirectional Associative Memory Neural Networks, and the Whale Optimization Algorithm, proving its efficiency and resilience.

Abhishek et al. [18] introduce a secure cloud storage framework utilizing DNA-based encryption and a fuzzy-based TOPSIS model for optimized storage server selection, demonstrating high security, robustness, and effective performance across various security benchmarks. Imdad et al. [19] reassess DNA cryptography by refining the encryption process and employing a DNA encoding/decoding table to enhance security, showing improvements in various analysis, and hamming weight, demonstrating better randomization and suitability for transmitting sensitive information. Mukherjee et al. [27] proposed a genetic algorithm designed to fortify weak keys produced by Random DNA-based key generators against brute force attacks. Venugopal et al. [37] present a Blockchain- driven framework aimed at securing healthcare big data, integrating encryption and decentralized networks with scalable distributed systems, ensuring data confidentiality, supporting various applications, and offering an audit trail for enhanced access control and regulatory compliance.

Suyel et al. propose a novel DNA cryptosystem that integrates DNA cryptography with finite state machines, significantly enhancing security and randomness compared to existing systems [42]. They also introduce an approach utilizing DNA computing to improve data security in cloud computing environments, featuring a 512-bit DNA-based secret key and various encoding rules [41]. Additionally, they explore the fundamentals of DNA computing, highlighting its advantages and applications across multiple fields [40]. Furthermore, they present a DNA cryptographic scheme and access control model for IoE-based cloud computing, featuring the integration of

the station-to-station key agreement protocol and Feistel cipher algorithms, demonstrating superior performance and resistance compared to existing security schemes [43]. Their comprehensive work addresses both fundamental principles and advanced applications of DNA computing.

Recent advancements in cryptography include QR coding [29], chip integration [30], and certificateless cryptography for IoT [36]. Developments in cloud data protection [14, 20, 34], DNA encryption [12], deep learning [13] and biometric and fingerprint processing enhancements [33] have also been notable. Additionally, blockchain technology has been integrated for heightened security and authentication [23]. Mealy machines are utilized in various domains, including neural network design [32], where they assist in modeling and analyzing network behaviors. They are also applied in vending machine representation [31] to simulate state transitions and operational logic. Furthermore, Mealy machines support simulation environments [28], aiding in the modeling of complex systems and processes.

The existing research on DNA-based cryptography reveals several limitations, including increased time and space complexity, insufficient randomness test analysis, and the omission of critical characters from coding tables. These gaps lead to potential inefficiencies and vulnerabilities, with some methods being susceptible to security breaches and computational overhead. To address these issues, there is a need to enhance the integration of advanced DNA encoding, optimize complexity, and improve the randomness and robustness of cryptographic techniques.

The proposed model addresses all the previously mentioned drawbacks by utilizing a randomly generated Mealy machine and a DNA coding table to ensure heightened security, authenticity, integrity, and efficiency. The methodology behind the proposed model involves both the sender and receiver registering their credentials with the server and requesting a key. The server then transmits the private and public keys separately to each party. The receiver encrypts specific attributes using their private key and sends them to the sender. The sender, in turn, uses the receiver's public key to decrypt these attributes for authentication. Upon successful verification, the receiver requests the required data. The sender then randomly generates DNA codebooks and transition tables for constructing Mealy machines. Using these components, the sender generates a 256-bit key and encrypts the confidential data. In the final step, the sender utilizes the receiver's public key to securely transmit the encrypted data along with the necessary parameters.

This paper briefly outlines the generation of DNA codebooks and Mealy machines, explaining the procedures for key generation, encryption, and decryption algorithms, along with their mathematical formulation and simulations. An in-depth analysis of the proposed model is conducted through various evaluations, including comparative analysis, security testing, NIST tests, avalanche effects, and frequency distribution, ensuring the model's efficacy. The primary objectives of this proposed model are as follows:

- A Novel cryptosystem is introduced by employing a DNA coding table and a randomly generated Mealy Machine
- Formation of a 256-bit secret key to assure a robust level of reliability
- Evaluating the resilience as well as security of the proposed scheme through different analysis
- The NIST test suite ensures proposed system randomness and efficiency by verifying unpredictability and absence of patterns in generated CT

3. Methodology

3.1 Cryptography

Cryptography is a technique employed to ensure secure communication and data protection among authorized parties. It plays a pivotal role in upholding the CIA triad (confidentiality, integrity, and authenticity) of data, even in the face of determined attackers, technological advancements, and unforeseen vulnerabilities.

3.1.1 DNA Cryptography

DNA is a complex and fundamental molecule responsible for carrying genetic information. It comprises four nucleotides: A, C, G, and T. In DNA-based cryptosystems [22], the CT typically takes the form of a DNA sequence or DNA codons, which are utilized in the intermediate encryption process. For instance, in a DNA cryptosystem, the PT "HI" is converted into "TAGATAGT." This transformation occurs by initially converting the PT into binary values, and then each 2-bit binary segment is mapped to specific DNA codons, such as 00 to A, 10 to G, 01 to T, and 11 to C. The randomly generated DNA codebooks provided in Tables 1 and 2 are utilized to convert DNA codons to characters and vice versa during the encryption and decryption processes.

Table 1 *Random conversion of DNA codons into Alphabets (code Book 1)*

DNA Codons	Alphabets and Numbers	DNA codons	Alphabets and Numbers
TTT	A	CTT	N
TCT	B	CCT	O
TAT	C	CAT	P
TGT	D	CTC	Q
TCC	E	CCA	R
TAC	F	CAC	S
TGC	G	CGC	T
TTA	H	CTA	U
TCA	I	ATG	V
TAA	J	CAA	W
TGA	K	CGA	X
TTG	L	ACT	Y
TGG	M	CAG	Z
CTG	1	GCC	6
ATC	2	TAG	7
GTG	3	CGT	8
CCG	4	GAT	9
TCG	5	TTC	0

Table 2 *Random conversion of DNA codons into SpecialCharacters (code Book 2)*

DNA Codons	Special Characters	DNA codons	Special Characters
CGG	:	AGG	**&**
ATT	;	GCC	**(**
AAT	<	GGT	**)**
AGT	>	GTT	**SPACE**
ACA	=	GTC	*
AAC	?	GCA	+
AGC	{	GAC	,
ATA	}	GGC	-
ACG	[	GTA	.
ACC	]	GAA	/
AAA	!	GGA	@
AGA	’	GAG	~
CCC	#	GGG	|
GCT	$	AAG	%

3.2 Mealy Machine

George H. Mealy developed the Mealy machine, a specific type of Finite State Machine (FSM) [1], which holds considerable importance in computer science and digital system theory. This machine is utilized to model how a system handles inputs and produces outputs according to its present state and the received input. The Mealy machine is formally characterized by six tuples, denoted as $\left(Q, \Sigma, \delta, \Delta, \lambda, q_0 \right)$, where Q and Σ represent finite and non-empty state sets and input alphabets, respectively, $\delta : Q \times \Sigma \to Q$ is the mapping function, Δ is the finite set of output alphabets, $\lambda : Q \times \Sigma \to \Delta$ is the output function and q_0 is the initial or start state.

3.2.1 Randomly Generated DNA Mealy Machine

The randomly generated DNA Mealy machine (RGDMM) can be built as

$Q = \{0, 1, 2, 3\}$
$\Sigma = \{A, C, G, T\}$

δ, λ - randomly designed
$0 \in Q$ - initial state
Δ - {11,10,01,00}

In this proposed model, the RGDMM employed to translate DNA codons to binary sequence and vice versa. The Mealy Machine parameters are discussed in (Section 3.2.1) Here the randomly generated transition table and output table is shown in (Table 3 and Table 4).

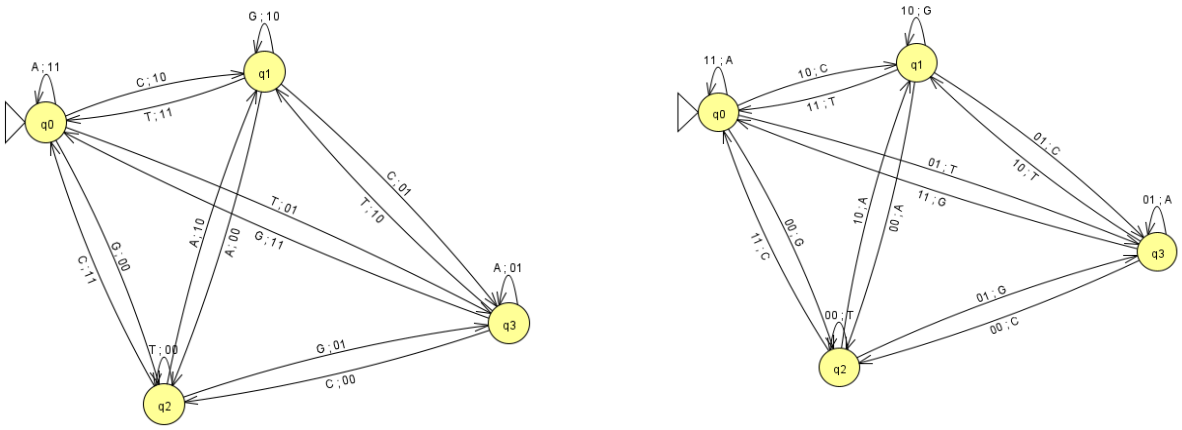
In order to randomly generate λ, an arbitrarily sequence is assigned from q_0 to q_3 distinctly to all the states in RGDMM. Subsequently, corresponding sequence for each state is inputted into the appropriate row of the state transition table. For instance, the sequence q_0, q_3, q_1, q_2 is allocated for the state q_0, and this sequence is then inputted into the corresponding row of state q_0 in the state table. Similarly, in order to generate δ an arbitrarily binary sequence 00, 10, 01, 11 is allocated to each state and an output table is created. The RGDMM visual depiction is given in Figure 1. The Figure 1(a) (RGDMM-1) can be directly created using (Table 3 & 4) and denoted as M. The idea of generating Figure 1(b) (RGDMM-2) from M using state and output table is described in the illustration of the encryption process in (section 3.8.2) and denoted as M'. From (Figure 1(a) & 1(b)) we can understand that Figure 1(b) is the invert process of Figure 1(a). Therefore, the mealy machine has the ability to perform both the forward and reverse process.

Table 3 *Mealy Machine state table*

State	Input A Next state	Input T Next state	Input C Next state	Input G Next state
q_0	q_0	q_3	q_1	q_2
q_1	q_2	q_0	q_3	q_1
q_2	q_1	q_2	q_0	q_3
q_3	q_3	q_1	q_2	q_0

Table 4 *Mealy Machine output table*

State	Input A Output	Input T Output	Input C Output	Input G Output
q_0	11	01	10	00
q_1	00	11	01	10
q_2	10	00	11	01
q_3	01	10	00	11



(a) **(b)**

Fig. 1 *RGDMM (a)RGDMM-M; (b) RGDMM-M'*

3.3 Experimental Setup

An Intel Core i5 computer operating at 1.8GHz, running Windows 10, equipped with 8GB of RAM and a 128GB SSD, serves as the hardware platform for implementing the proposed strategy. Python, a user-friendly programming language, plays a central role in this implementation. Python has gained widespread adoption across various domains due to its clarity, readability, and extensive library support. It proves valuable to both novice and experienced programmers. Notably, Python finds application in web development, machine learning,

artificial intelligence, and data analysis and visualization. Furthermore, Python software is compatible with Windows, Linux, and Unix operating systems. The results of the proposed scheme have been obtained through the execution of various experiments using distinct datasets from the 20 newsgroups [44].

3.4 Proposed Scheme

The brief summary of this proposed DNA cryptosystem is as follows:
The process begins with the sender generating a 256-bit private key using the receiver's credentials and a DNA encoding table. This is followed by the key generation, encryption, and decryption procedures. Subsequently, the mathematical formulation and algorithms are outlined, accompanied by an illustrative example of the model. The NIST test is then addressed, and the security aspects of the system are thoroughly evaluated. To further demonstrate the system's efficacy, a comparative analysis is conducted to assess the model's performance. Additionally, frequency distribution and the avalanche effect are examined to ensure randomness. The workflow of the proposed DNA cryptosystem is depicted in Figure 2.

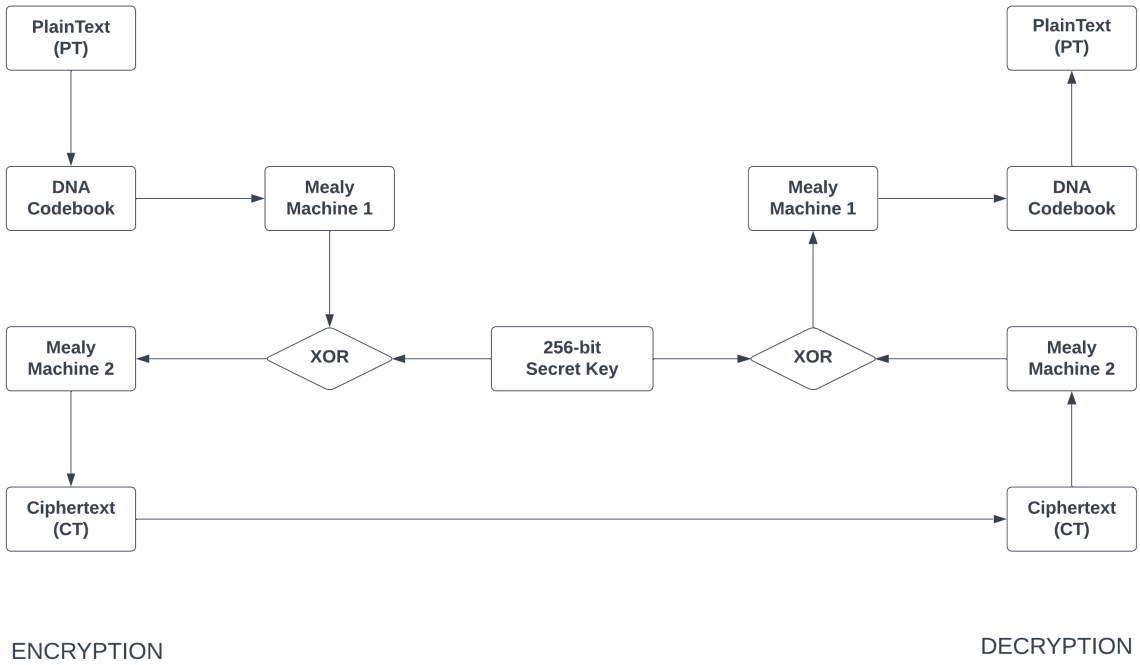


Fig. 2 *Workflow of the proposed model*

3.4.1 Key Generation

The sender arbitrarily selecting ten characters of recipient information. Each of these characters is then transformed into a DNA sequence utilizing the encoding tables (Table 1 and 2). Next, the resulting DNA sequences are transformed into ASCII values, which are subsequently concatenated. These concatenated ASCII values are further converted into their corresponding binary representation. Finally, the sender extracts the first 256 bits from this binary sequence, considering it as the secret key to be used in encryption and decryption processes.

3.4.2 Encryption

The sender initiates the encryption process by treating the secret message as PT. Each character within PT is then replaced with its corresponding DNA sequence based on the DNA code book (Table 1 and 2). Subsequently, this DNA sequence is input into M (Figure 1(a)) to yield a binary sequence. An XOR operation is executed amid the binary string and the 256-bit secret key. If the binary string exceeds 256 bits, it is divided into 256-bit blocks. The XOR operation is executed on each block individually using the secret key, and the results are concatenated. The output of the XOR operation is passed to M' (Figure 1(b)) to produce the CT represented as a DNA sequence (DNA string). This DNA string is considered as the CT.
The sender proceeds to transmit the encrypted message, along with associated parameters, to the receiver employing the receiver's public key.

3.4.3 Decryption

The recipient initiates the decryption process to retrieve the PT. To achieve this, the recipient must undertake the reverse procedure of the encryption. Initially, using their private key, the recipient acquires the CT along with its associated parameters. The CT, in the form of a DNA string, is then input into M' (Figure 1(b)) to obtain a binary string. The execution of M' is carried out using the state and output tables in an inverse manner. Subsequently, an XOR operation is executed amid the hexadecimal string and the secret key. The resulting string is then processed through M (Figure 1(a)) to obtain a DNA string. To decode the DNA string into PT, each set of three DNA codons is transformed into its corresponding character using the codebooks (Table 1 and 2). This process yields the original PT.

3.5 Mathematical Formulation

The mathematical formulation of key generation, encryption and decryption algorithm is as follows. The notations with its descriptions used in the mathematical formulation of an algorithm is listed in Table 5.

Table 5 *Notations with its descriptions used in mathematical formulation*

Notation	Description	Notation	Description
S	set of receiver's credentials	PT	Plaintext
Σ	set of DNA codons	n	Length of characters
C	Character in the confidential text	D	DNA sequence
$CB(C_i)$	DNA codons for the character C_i	KG_p	Key generation process
KG	256-bit generated secret key	M	Mealy machine 1
$ord(n)$	ASCII value of n (DNA codons)	M'	Mealy machine 2
$BR(n)$	Binary representation of $ord(n)$	$\oplus$	XOR operation
BS	Concatenated binary sequence	$M'(D)$	DNA string produced by M'
$M(B)$	Binary sequence produced from M	$\oplus(M(B), KG)$	XOR b/w binary sequence and secret key

3.5.1 Key generation

Initially, consider 10 characters receiver credentials in the concatenated form for key generation process. Let S= {Name, DoB, Email id, SSN, Passport id} and Σ= {A, C, G, T}.

$$KG_p = C_1C_2 \ldots c_{10} \quad \forall C_i \in S \tag{1}$$

$$KG_p = (x_1y_1z_1)(x_2y_2z_2) \ldots (x_{10}y_{10}z_{10}) \quad \forall (x_iy_iz_i) \in \Sigma \tag{2}$$

Where C_i converted (x_i, y_i, z_i) by codebook 1 & 2 and $x_i=y_i=z_i$; $x_i \neq y_i=z_i$; $x_i=y_i \neq z_i$; $x_i=z_i \neq y_i$

$$KG_p = ord(x_1)\,ord(y_1)\,ord(z_1)\,ord(x_2)\,ord(y_2)\,ord(z_2) \ldots ord(x_{10})\,ord(y_{10})\,ord(z_{10}) \tag{3}$$

$$KG_p = BR(ord(x_1))\,BR(ord(y_1))\,BR(ord(z_1))\,BR(ord(x_2))\,BR(ord(y_2))\,BR(ord(z_2)) \ldots BR(ord(x_{10}))\,BR(ord(y_{10}))\,BR(ord(z_{10})) \tag{4}$$

In Equ (4), $BR(n)$ can be computed by $BR(n) = \sum_{i=0}^{\infty} \left(\frac{n}{2_i} mod 2 \right) \times 10^i$

$\frac{n}{2_i}$= shows the results of division of $\frac{n}{2_i}$, $\frac{n}{2_i} mod 2$ = extracts the remainder of the division (the values will be either 0 or 1) and 10^i=used to position the extracted remainder within the binary representation.
Now consider, Equ (4) is equivalent to BS

$$BS = BR(ord(x_1))\,BR(ord(y_1))\,BR(ord(z_1))\ldots BR(ord(x_{13}))\,BR(ord(y_{13}))\,BR(ord(z_{13}))$$

$$KG_p = BS[0:256] = KG \tag{5}$$

Equ (5) extracts the first 256-bit binary sequence from Equ (4) which is the required secret key (KG).

3.5.2 Encryption

Let us assume PT with n characters

$$PT = C_1 C_2 \ldots C_n \tag{6}$$

$$D = CB(C_1)\,CB(C_2)\ldots CB(C_n) \tag{7}$$

Where, $CB(C_i) = (x_i y_i z_i) = d_i,\ \forall d_i \in \Sigma$ and $x_i = y_i = z_i\ ;\ x_i \neq y_i = z_i\ ;\ x_i = y_i \neq z_i\ ;\ x_i = z_i \neq y_i$
Therefore, Equ (7) can be written as

$$D = d_1 d_2 \ldots d_n \quad \forall d_i \in \Sigma \tag{8}$$

Now, consider M with transition and output functions T and O respectively.

$$M(B) = O(d_1)\,O(d_2)\ldots O(d_n) \tag{9}$$

$$= b_1 b_1' b_2 b_2' \ldots b_n b_n' \tag{10}$$

where, $O(d_i)$ - output produced by M for the i^{th} DNA codon d_i in the DNA sequence D. When passing d_i as input to M the output will be 2-bit binary value i.e., $d_i = b_i b_i',\ \forall d_i,\ i = 1\ to\ n\ \exists\ b_i b_i',\ i = 1\ to\ n$.

$$XOR(M(B), KG) = \left(b_1 \oplus k_1\ b_1' \oplus k_2 \ldots b_n \oplus k_{2n-1}\ b_n' \oplus k_{2n}\right) \tag{11}$$

$$= x_1 x_1' x_2 x_2' \ldots x_n x_n' \tag{12}$$

where, b_i denotes i^{th} bit of the resulting binary sequence $M(B)$, k_i is the i^{th} bit of the 256-bit secret key and x_i be the resultant of XOR between $M(B)$ and KG. Now consider M', M' be the M with interchanging corresponding input and output function T' and O' respectively. On passing each 2-bits of XOR, i.e., $x_i x_i'$ in M', we get the DNA sequence d_i.

$$M'(D) = O'(x_1 x_1')\,O'(x_2 x_2')\ \ldots O'(x_n x_n')$$

$$= d_1' d_2' \ldots d_n' \tag{13}$$

$$= D' \tag{14}$$

$$= CT$$

Where d_i'- output produced by M' for the i^{th} XOR bits $x_i x_i'$ in the resultant of XOR.

3.5.3 Decryption

The mathematical formulation of the decryption algorithm is invert process of encryption.

3.6 Algorithms

Generation of 256-bit secret key is given as follows.

Input: Recipient Information (10 characters)
Output: 256-bit Secret Key (K)

1. Select 10 characters from the recipient information
2. Convert each selected character into a DNA sequence using the specified encoding tables (Tables 1 and 2)
3. Transform each DNA sequence into its associated ASCII values
4. Concatenate the ASCII values to form a unified sequence
5. Translate the concatenated ASCII sequence into binary representation
6. Extract the first 256 bits from the binary sequence
7. Utilize the extracted 256-bit sequence as the secret key

The encryption process is given as follows.

Input: Secret Message (PT)
Output: Ciphertext (CT)

1. Start with the PT
2. Replace each character in PT with its corresponding sequence using the DNA code books to get D
3. Pass the D through M to obtain a binary sequence $M(B)$
4. Perform an XOR operation between $M(B)$ and k, then concatenate the result
5. Direct the XOR output through M' to generate a D'
6. D' is the CT
7. End

The decryption process is given as follows.

Input: CT with parameters
Output: PT

1. Begin with the CT D'
2. Decode D' to binary string through M'
3. Perform XOR operation between binary string and k to get $M(B)$
4. Translate the $M(B)$ into D using M
5. Convert D into PT with the help of Table 3 and 4
6. PT
7. End

3.7 Simulation of the Proposed Model

In this part, we have provided an illustration of proposed model.

3.7.1 Key Generation

The sender must adhere to the subsequent rules for generating the secret key.

a) The sender is having the recipient credentials.
Name: PQRST, Phone no: 1234567890, Email id: Y@hotmail.com, Passport id: BHKF8512, SSN: 654812783, using these credentials, sender forms a string of length 10.
PQ12Y@BH65

b) Pass a) in to code book 1 and 2 to get the corresponding DNA sequence.
CATCTCCTGATCACTGGATCTTTAGCGTCG

c) Modify b) to its ASCII value
67658467846767847165846765678471716584678484846571677184677 1

d) Transform c) into equivalent binary value
0011011000110111001101100011010100111000001101000011011000110111001110000011010000110110001101110011011000110111001110000011010000110111001100010011011000110101001110000011010000110110001101110011011000110101001101100011011100111000001101000011011100110001001101110011000100110110001101010011100000110100001101100011011100111000001101000011100000110100001110000011010000110110001101010011011100110001001101100011011100110111001100010011100000110100001101100011011100110111001100010011100000110100001101100011011100110111001100011011100110001
The length of string d) has 480-bits
e) Extract the first 256-bit by removing the additional binary value.
0011011000110111001101100011010100111000001101000011011000110111001110000011010000110110001101110011011000110111001110000011010000110111001100010011011000110101001110000011010000110110001101110011011000110101001101100011011100111000001101000011011100110001
The aforementioned 256-bit binary value is the integral secret key required for the encryption and decryption procedure.

3.7.2 Encryption

a) The sender encrypts the confidential data i.e., Plaintext (PT)
PT=**HELLO WORLD**
b) To obtain corresponding DNA sequence of PT pass a) into code book 1 and 2
D=**GTTTTATCCTTGTTGCCTGTTCAACCTCCATTGTGT**
c) Convert D into binary string by considering D is input to M Figure 1(a) and that $M(B)$ is the output of M consisting of binary sequence.
$M(B)$ = **00000000001011100110110000000010011011101100101010011010011110110101100000**
d) Split k from KG, where k is the length of $M(B)$.
k=**00110110001101110011011000110101001110000011010000110110001101110011 1000**
e) Execute XOR between $M(B)$ and k.
XOR=00110110000110010101101000110001111001011010000100000010110000011000 1000
f) In order to obtain DNA sequence D', the recipient has to perform the inverse process of M using (Table 1 and To generate M'. The sample is provided below.
1. In the resultant of the XOR, each 2-bit of binary value is transformed in to DNA sequence. Consider initial 2 XOR bits, i.e., 00. q_0 is the initial state. As the output 00 is given in the input column G of the output Table 4, 00 is replaced by G the initial character of $D' = G$.
2. Now, use state Table 3 to find the following state. In Table 3 for state q_0, q_2 is the next state for the input G. Therefore, the state q_0 changes to state q_2.
3. The 2nd 2-bit binary value of the XOR is 11. Again, by using Table 4, 11 is replaced by C, where in state 2 output 11 is presented in input column C. Now $D' = GC$.
4. In Table 3 for input C of the state q_2, the next state as q_0. The state q_2 changes to q_0.
5. The 3rd 2-bit is 01. In Table 4, 01 is identified in the column, is inputted as T for state q_0 resulting in the replacement of 01 with T. Now, $D' = GCT$.
6. Table 3 shows that when the state q_0 receives the input T, it transitions to the state q_3. By following the above process for the last bit of resultant of XOR, we get
D'=**GCTTAGTCAATGACGGGCCATGAGCTTATGTGTAAA**
D' is considered as CT.

3.7.3 Decryption

To recover the PT, the receiver uses the reverse encryption process.
a) The recipient has the CT
$CT = D' =$ **GCTTAGTCAATGACGGGCCATGAGCTTATGTGTAAA**
b) Pass D' to M' to get the binary sequence (XOR value).
XOR = 001101100001100101011010001100011110010110100001000000101100000110001000
c) Perform the XOR operation between secret key k and b) to get another binary string.
$M(B) =$
00000000001011100110110000000010011011101100101010011010011110110101100000

d) Using M convert $M(B)$ into D. Each 2 bit of $M(B)$ is converted into DNA codons. The similar working process as given in the (section 4.5.2). Therefore, we get

$D =$ GTTTTATCCTTGTTGCCTGTTCAACCTCCATTGTGT

e) Eventually, D is converted in to PT using code book 1 and 2.

$PT =$ **HELLO WORLD**

The output of this illustration of proposed model using python is given in the Figure 3.

```
PS G:\multiplexer> python -u "g:\multiplexer\Key Generation.py"
User Data's = PQ12Y@BH65
ASCII string =  676584678467678471658467656784717165846784848465716771846771
Binary String =  0011011000110111001101100011010100111000001101000011011000110111001101100011011100110000
1101000011011100110001001101100011010100111000001101000011011000110111001101100011011100110000100110
1110011000100110110001101010011100000110100001101100011011100110000011010000110111001100010011011100
0011011000110111001101110011000100111000001101000011011000110111001101110011011100110001
0100001101110011000100110110001101010011100000110100001101100011011100110101001101100011011100110001
PS G:\multiplexer> python -u "g:\multiplexer\DNA Encryption.py"
PT= HELLO WORLD
D = TTATCCTTGTTGCCTGTTCAACCTCCATTGTGT
M(B) = 0110000011101101110110100100000110111000100100000111000000001101011
XOR = 0101011011011010111011000111010010000000101001001101011000101101111
D' = CT = TAATTTTGTCTGGGTCAATTAGCCCTATAATTG
PS G:\multiplexer> python -u "g:\multiplexer\DNA Decryption.py"
CT = D' = TAATTTTGTCTGGGTCAATTAGCCCTATAATTG
XOR = 0101011011011010111011000111010010000000101001001101011000101101111
M(B) = 0110000011101101110110100100000110111000100100000111000000001101011
D = TTATCCTTGTTGCCTGTTCAACCTCCATTGTGT
PT = HELLO WORLD
```

Fig. 3 *Output of the proposed model using python*

Security Analysis

The security analysis evaluates the proposed model's robustness against various attacks by utilizing key components such as a 256-bit key, XOR operations, and the Mealy machine. These mechanisms effectively counter brute-force, ciphertext-only, and known plaintext attacks while secure key exchanges mitigate man-in-the-middle and phishing threats, providing valuable insights into the model's strengths and ensuring its reliability in safeguarding sensitive data.

- Brute – force attack: During this form of assault, the adversary attempts to explore all possible variations of the secret key used in the encryption algorithm. However, the size of the secret key in the proposed model is 256 bits, rendering it exceptionally challenging for the attacker to explore. This is due to the immense computational effort required, as the attacker would need to test a staggering 2^{256} combinations to uncover the correct information.
- Ciphertext-only attack: In this cryptographic assault, the assailant endeavors to decipher the encrypted data without knowledge of the encryption algorithm. In this type of attack, the assailant seeks to deduce the PT by identifying repeated patterns within the CT. However, in the proposed approach, the same PT character yields unique CT characters. As a result, this model is resistant to attacks based solely on ciphertext analysis.
- Kown plaintext attack: An attacker employing a known plaintext attack attempts to deduce the encryption technique or key. In such an attack, the assailant has the ability to obtain both the CT and the associated PT. However, in the proposed methodology, we employ the XOR operation, the Mealy machine, and codebooks as key components. Furthermore, in this model, distinct CT characters are generated for the same PT and vice versa. As a result, the proposed technique proves resilient against known plaintext attacks.
- Man-in-the-middle attack
- In this form of assault, the adversary intercepts the transmission of information and attempts to manipulate information exchanged between the sender and the receiver. This attack is commonly employed for data corruption and covert surveillance of individuals' activities, among other purposes. However, in the proposed model, the personal attributes and confidential information of the receiver are transformed into a scrambled form using the receiver's private key and the sender's public key. This transformation is executed securely, and the resulting data is shared between the parties in a protected manner. Consequently, intruders face unsuccessful attempts to access the information. Thus, the model demonstrates resilience against this type of attack.
- Phishing attack: In a phishing attack, the intruder employs a simple trick to target users by sending harmful links. Users may unwittingly fall into the trap by clicking on these malicious links, resulting in the loss of personal data and confidential information to hackers. However, in the methodology, all details are shared

in a ciphered format utilizing the private and public keys of the sender and recipient. Consequently, this proposed scheme is resilient against such attacks.

3.8 NIST Test

The randomness of the CT significantly influences the reliability of a cryptosystem. The NIST test suite is employed to evaluate this randomness. This suite assesses various statistical properties of cryptographic algorithms, including the uniformity and unpredictability of the generated data, ensuring that the encryption algorithms meet rigorous security and reliability standards. The suite includes tests such as the monobit test, block frequency test, and run test, which analyze the binary representation of the CT to determine its randomness. A p-value that is higher than 0.01 indicates that the CT meets the required randomness criteria. Table 6 provides the notations and descriptions for these tests, and the following section offers a brief overview of these tests.

Table 6 *Notations and descriptions of NIST Test analysis*

Notation	Description	Notation	Description
l	Length of binary string	S_k	Absolute value
N	Number of blocks in the input	$erf\,c$	Complementary error function
M	Number of bits in a block	S_{obs}	Test static in frequency test
$.\pi$	Proportion of ones in binary string	ε	Sequence of bits
π_k	The proportion of ones in the i^{th} block	$igamc$	Incomplete gamma function
P_{value}	Probability of test static	χ^2_{obs}	The test statistic in block frequency test
Vn_{obs}	Run test statistics	k	length of run

3.8.1 Monobit Test

The monobit test, commonly referred to as the frequency test, evaluates the parity between 1s and 0s in the binary sequence, providing a fundamental gauge of randomness. Deviations from an even distribution may indicate potential irregularities in the sequence's randomness properties. The P-values of the monobit test is computed using Equations 16 to 18.

$$S_l = \sum_{j=1}^{l} X_j \;\; \forall\, X_j = 2\varepsilon - 1 \tag{16}$$

$$S_{obs} = \frac{|S_l|}{\sqrt{l}} \tag{17}$$

$$P_{value} = erf\,c\left(\frac{S_{obs}}{\sqrt{2}}\right) \tag{18}$$

3.8.2 Frequency Test Within a Block

Testing the frequency within a block assesses the distribution of ones in M-blocks, targeting a frequency close to M/2. Equations 19 through 21 are used to determine the P-values of this test.

$$\pi_k = \frac{\sum_{j=1}^{M} \varepsilon_{i=1} M + j}{M} \;\; \forall\, 1 \le k \le N \tag{19}$$

$$\chi^2_{obs} = 4M \sum_{k=1}^{N} \left(\pi_k - 1/2\right) \tag{20}$$

$$P_{value} = igamc\left(\frac{N}{2}, \frac{\chi^2_{obs}}{2}\right) \tag{21}$$

3.8.3 Runs Test

The run test, examines sequences of similar bits. It traverses through the entirety of the binary string, calculating the quantity of bits and comparing the count of ones to values from a random sequence, additionally assessing the speed of zero-to-one transitions. Equations 22 to 24 are used to compute the P-values for the runs test.

$$\pi = \frac{\sum_i \varepsilon_i}{n} \tag{22}$$

$$Vn_{obs} = \sum_{k=1}^{n-1} r(k) + 1, \begin{cases} r(k) = 0 \ \ if \ \ \varepsilon_k = \varepsilon_{k+1} \\ r(k) = 1 \ \ otherwise \end{cases} \tag{23}$$

$$P_{value} = erfc\left(\frac{\left|Vn_{obs} - 2n\pi(1-\pi)\right|}{2\sqrt{2n}\,\pi(1-\pi)}\right) \tag{24}$$

4. Results and Discussion

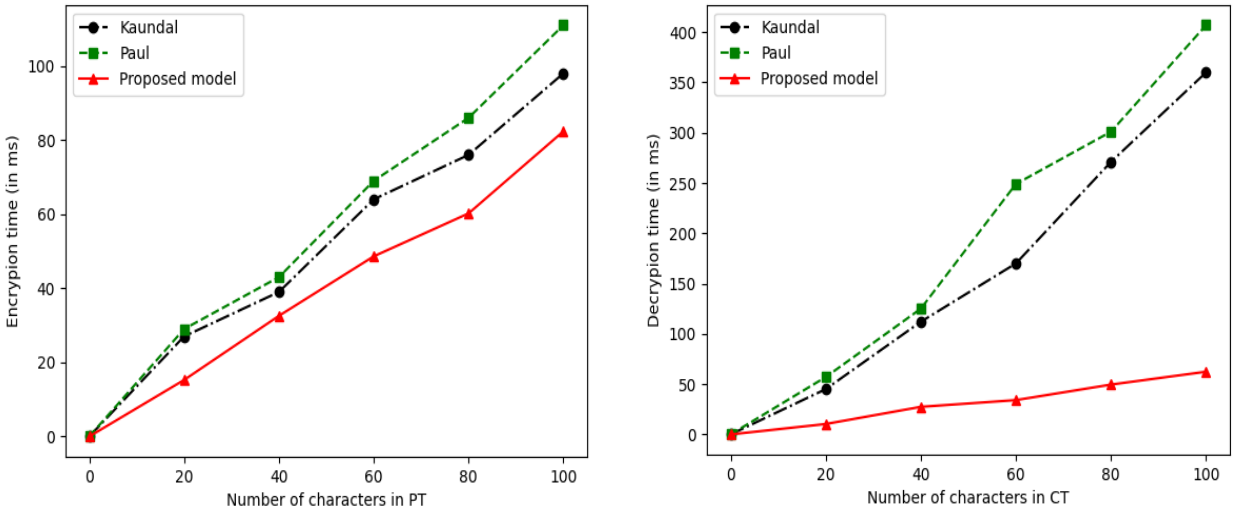
4.1 Comparative Analysis

In this section, we critically assess the proposed DNA cryptographic model against existing DNA computing approaches by Kaundal et al. [4] and Paul et al. [5] to highlight its superior efficiency and security. Kaundal model utilizes a One-Time Pad (OTP), leading to extended encryption and decryption times due to its complex processing requirements. Conversely, the Paul model needs a limited key size, making it impractical to encrypt large messages and vulnerable to security breaches. The proposed model enhances these existing DNA computing approaches by integrating a randomly generated Mealy machine and a DNA encoding table, thereby improving both security and operational efficiency.

To evaluate a time complexity, we've considered five different datasets of varying sizes, performing each operation up to 50 times and then computing the average. The comparison between the existing models and the average execution time of the proposed model is presented in Table 6. The graphical representations of Figure 4(a) illustrate that the proposed model exhibits the lowest time complexity for encrypting data compared to Kaundal's model, with reductions of 24.54%, and Paul's model, with reductions of 31.43%. Similarly, Figure 4(b) demonstrates that the proposed model displays the lowest time complexity for decrypting data compared to Kaundal's model, with reductions of 97.53%, and Paul's model, with reductions of 98.91%. The significant enhancements in time complexity not only showcase the efficacy and efficiency of the proposed method in DNA cryptography but also indicate its potential for widespread adoption across diverse applications, showcasing its superiority over existing models.

Table 6 *Comparative analysis for encrypting and decrypting the data of different length (in ms)*

PT length (char)	Kaundal's model [4]		Paul's model [5]		Proposed model	
	Encryption time	Decryption time	Encryption time	Decryption time	Encryption time	Decryption time
20	27	45	27	57	15.25	10.35
40	41	112	43	125	32.56	27.42
60	60	170	61	249	48.62	34.12
80	79	271	86	301	60.21	49.56
100	99	360	111	407	82.36	62.37



(a) **(b)**

Fig. 4 *Comparative analysis (a) Encryption; (b) Decryption*

4.2 Frequency Distribution

Frequency analysis plays a crucial role in cryptography to ascertain randomness. Intruders often attempt to discern patterns by matching the frequency of observed CT with that of English alphabets in PT to uncover the PT. In the proposed model, we adopt a proactive approach to counteract this type of threat. To evaluate the frequency analysis within this model, we employ four distinct sets of data in the PT. Table 7 demonstrates the distribution of frequency, specifically the occurrences of DNA codons within the CT for each PT. Additionally, Figure 5(a) graphically illustrates the frequency histogram distribution of DNA codons in the CT for each dataset, emphasizing the robustness of the model in preserving the security of the data.

Table 7 *Result of frequency analysis*

Dataset	PT	A	G	C	T
1	50A's	41	32	35	52
2	50G'S	34	26	40	50
3	50C'S	30	21	44	55
4	50T'S	30	40	46	34

4.3 Avalanche Effect

The Avalanche effect (A.E.) is a significant metric used to evaluate the security of an encryption technique in cryptography. It measures how alterations in either the PT or the secret key result in modifications to the CT. Typically, this effect involves flipping just a single bit in the PT or the secret key to observe how many bits change in the CT. Equ (15) is commonly used to quantify the Avalanche effect, with a threshold of 50% often considered an encryption technique is deemed secure. If the Avalanche effect exceeds this threshold, the scheme is considered robust against attacks that rely on statistical analysis, making it harder for intruders to predict the PT.

To compute the A.E. within the proposed model, we employ five distinct datasets of varying sizes and introduce a 1-bit change in the PT for each set. The results of the Avalanche effect are presented in Table 8. Additionally, Figure 5(b) provides graphical representation, shows the robustness of the proposed scheme with an average Avalanche effect of 79%. This outcome signifies the model's resilience against statistical attacks and its ability to maintain the security of the data.

$$A.E = \frac{The\ no.\ of\ \ characters\ changed\ in\ the\ cipher\ text}{Total\ no.\ of\ \ characters\ in\ the\ cipher\ text} \times 100 \quad (15)$$

J. of Soft Computing and Data Mining Vol. 5 No. 2 (2024) p. 245-263

Table 8 *Avalanche effect analysis by changing 1-bit in the PT*

Size of PT	Alteration in CT
50	72%
100	77%
150	84%
200	79%
250	73%

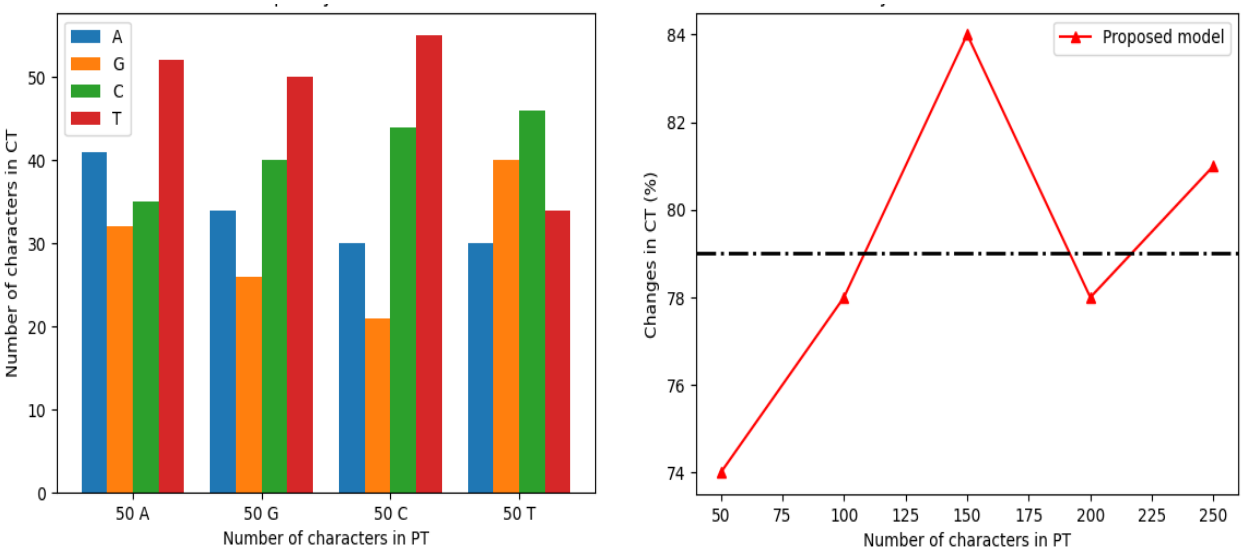


Fig. 5 *(a) Frequency distribution chart; (b) Analysis of Avalanche effect*

4.4 NIST Test

The randomness of the CT produced by the proposed model is analyzed using the Monobit Test, Block Frequency Test, and Runs Test from the NIST test suite, as described in Section 3.9. The p-values of the proposed DNA cryptosystem is compared with those of an existing DNA cryptosystem implemented in Python. To ensure consistency, the CT from both the proposed and existing models is converted into binary. For each test, the same set of PT's is used for evaluation, with 10 experiments conducted and the corresponding p-values calculated. The average p-values between the proposed and existing models CT's are displayed in Table 10, with graphical representations in Figures 6, 7, and 8. The results show that the proposed DNA cryptosystem achieves the highest p-value, demonstrating superior security compared to existing techniques.

Table 10 *Average P-value of the NIST evaluation*

Model	Monobit test	*Frequency test within a block*	*Runs test*
Kaundal's model [4]	0.531	0.658	0.618
Paul's model [5]	0.577	0.711	0.691
Proposed model	0.681	0.758	0.728

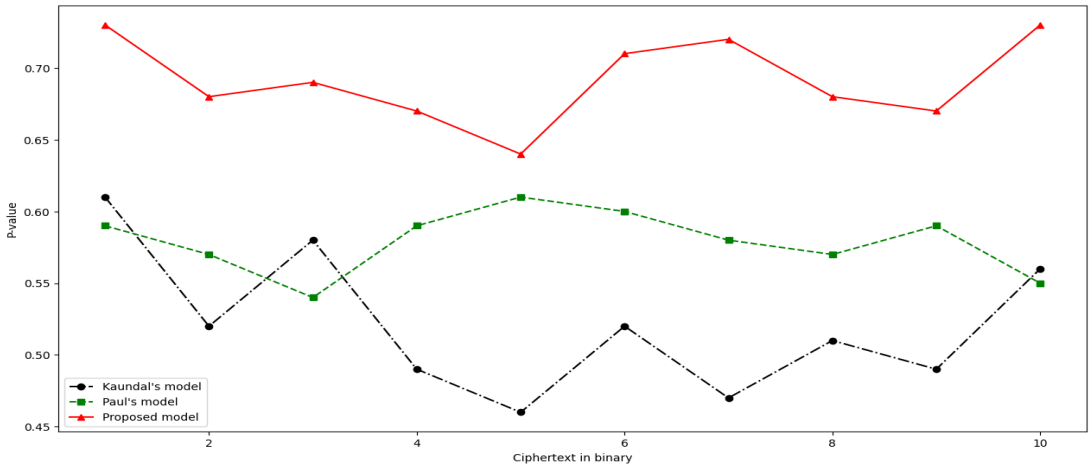


Fig. 6 *Monobit test*

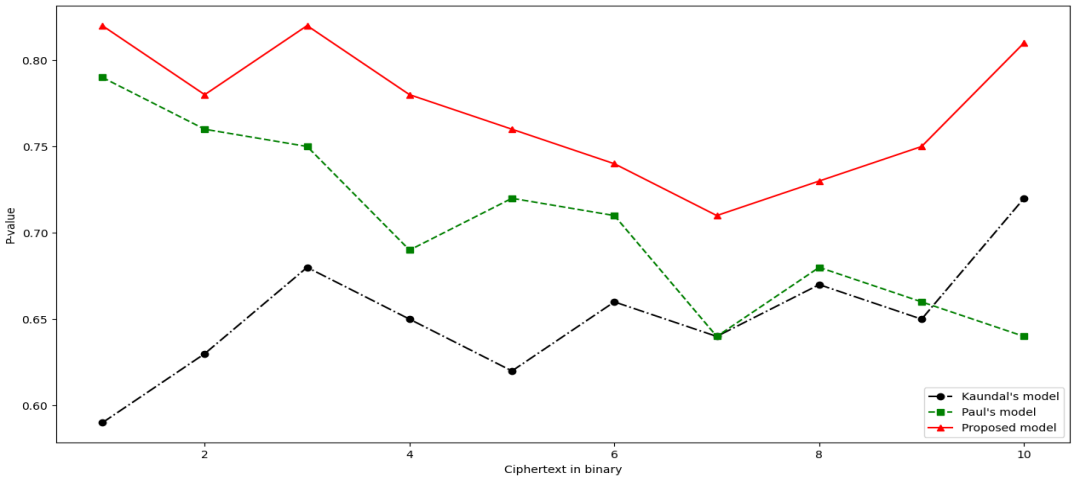


Fig. 7 *Frequency test within a block*

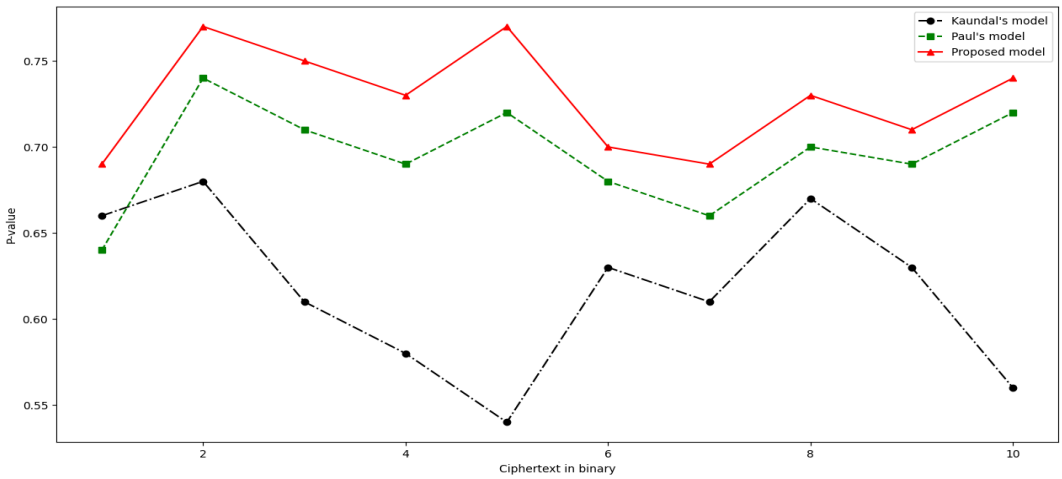


Fig. 8 *Runs test*

5. Conclusions

This study introduces an innovative DNA cryptosystem employing Mealy machine. In this innovative model, we generate a highly secure 256-bit secret key using the receiver's attributes. The proposed model incorporates four layers of security. Initially, a DNA sequence is generated from the PT, which is then interpreted through a Mealy machine. Following this, an XOR operation is conducted, and the resulting data is transmitted into another Mealy machine, ultimately producing the CT in the form of a DNA sequence. We provide a mathematical implementation and illustrative examples to elucidate the process. Furthermore, we discuss various aspects to ensure the model's efficacy, including frequency distribution, security analysis, comparative assessments, and an avalanche effect. The evaluation process includes three randomness tests from the NIST test suite, comparing them with existing schemes, and observing higher P-values in these tests. The overall findings and discussions demonstrate the superiority of the scheme compared to existing models. In the future, we aspire to conduct the remaining NIST tests to refine the randomness of this model further. Furthermore, we aim to explore practical applications utilizing this scheme.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm their contribution to the paper as follows: ***study conception and design:*** *Gaverchand K;* ***data collection:*** *Yasmin A;* ***analysis and interpretation of results:*** *Venkatesan R, Kavikumar Jacob;* ***draft manuscript preparation:*** *Gaverchand K, Venkatesan R. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Hopcroft, J. E., Motwani, R., Ullman, J. D. (2001) Introduction to automata theory, languages, and computation, ACM SIGACT News 32(1), 60-65, https://doi.org/10.1145/568438.568455
[2] Wang, Z., Yu, Z. (2011) Index- based symmetric DNA encryption algorithm, In: Proceedings of the 4th International Congress on Image and Signal Processing, Shanghai, IEEE, 2290-2294, https://doi.org/10.1109/CISP.2011.6100690
[3] Gupta, R., Singh, R. (2015) An improved substitution method for data encryption using DNA sequence and CDMB, In: Proceedings of the 3rd International Symposium, India, 197-206, https://doi.org/10.1007/978-3-319-22915-7_19
[4] Kaundal, A. K. and Verma, A. K. (2015) Extending Feistel structure to DNA Cryptography, Journal of Discrete Mathematical Sciences and Cryptography, 18(4), 349-362, https://doi.org/10.1080/09720529.2014.995975
[5] Paul, S., Anwar, T. and Kumar, A. (2016) An innovative DNA cryptography technique for secure data transmission, International Journal of Bio-informatics Research and Applications, 12(3), 238, http://dx.doi.org/10.1504/IJBRA.2016.078235
[6] Biswas, M. R., Alam, K. M. R., Akber, A. and Morimoto, Y. (2017) A DNA cryptographic technique based on dynamic DNA encoding and asymmetric cryptosystem. In: Proceedings of the 4th International Conference on Networking, Systems and Security, IEEE, 1-8, https://doi.org/10.1109/NSYSS2.2017.8267782
[7] Malathi, P., Manoaj, M., Manoj, R., Vaikunth Raghavan and Vinodhini, R. E. (2017) Highly improved DNA based steganography, Procedia Computer Science, 115, 651-659, https://doi.org/10.1016/j.procs.2017.09.151
[8] Pujari, S. K., Bhattacharjee, G. and Bhoi, S. (2018) A hybridized model for image encryption through genetic algorithm and DNA sequence. Procedia Computer Science, 125, 165-171, https://doi.org/10.1016/j.procs.2017.12.023
[9] Elhadad, A. (2019) Data sharing using proxy re-encryption based on DNA computing, Soft Computing, 24, 2101-2108, https://doi.org/10.1007/s00500-019-04041-z
[10] Enayatifar, R., Guimaraes, F. G. and Siarry, P. (2019) Index-based permutation diffusion in multiple-image encryption using DNA sequence, Opt. Lasers Eng. 115, 131-140, https://doi.org/10.1016/j.optlaseng.2018.11.01
[11] Wang, Y., Han, Q., Cui, G. and Sun, J. (2019) Hiding messages based on DNA sequence and Recombinant DNA technique, IEEE Transactions on Nanotechnology, 18, 299-307, https://doi.org/10.1109/TNANO.2019.2904842
[12] Namasudra, S. (2020) Fast and secure data accessing by using DNA computing for the cloud environment, IEEE Transactions on Services Computing, 15(4), 2289-2300, https://doi.org/10.1109/TSC.2020.3046471

[13] Jin, X., Xiao, Y., Li, S. and Wang, S. (2020) Deep learning-based side channel attack on HMAC SM3, International Journal of Interactive Multimedia and Artificial Intelligence, 6(4), 113-120, https://doi.org/10.9781/ijimai.2020.11.007
[14] Namasudra, S., Chakraborty, R. and Bharat, S. (2020) FAST: Fast Accessing Scheme for data transmission in cloud computing, Peer to Peer Networking and Applications, 14, 2430-2442, https://doi.org/10.1007/s12083-020-00959-6
[15] Reddy, M.I., Kumar, A.P.S. and Reddy, K.S. (2020) A secured cryptographic system based on DNA and a hybrid key generation approach, Biosystems, 197, https://doi.org/10.1016/j.biosystems.2020.104207
[16] Namasudra, S., Devi, D., Seifedine, K., Revathi, S. and Shanthini, A. (2020) Towards DNA based data security in the Cloud Computing environment, Computer Communications, 151, 539-547, https://doi.org/10.1016/j.comcom.2019.12.041
[17] Namasudra, S., Chakraborty, R., Majumder, A. and Moparthi, N.R. (2020) Securing multimedia by using DNA-based encryption in the cloud computing environment, ACM Transactions on Multimedia Computing, Communications and Applications, 16(3), 1-19, http://dx.doi.org/10.1145/3392665
[18] Majumdar, A., Biswas, A., Majumder, A., Sood, S. K. and BAISHNAB, K. (2021) A novel DNA-inspired encryption strategy for concealing cloud storage, Front. Comput. Sci. 15(3), https://doi.org/10.1007/s11704-019-9015-2
[19] Imdad, M., Ramli, S.N. and Mahdin, H. (2021) Increasing randomization of ciphertext in DNA cryptography, International Journal of Advanced Computer Science and Applications, 12(10), 423-429, https://doi.org/10.14569/IJACSA.2021.0121047
[20] Namasudra, S. (2021) Data Access Control in the Cloud Computing Environment for Bioinformatics, International Journal of Applied Research in Bioinformatics, 11, 40-50, https://doi.org/10.4018/IJARB.2021010105
[21]
[22] Sadkhan, S. B. (2021) A Proposed Genetic Algorithm Attack for Public Key Cryptosystem, IEEE 7th International Engineering Conference Research and Innovation amid Global Pandemic, 194-199, https://doi.org/10.1109/10.1109/IEC52205.2021.9476146
[23] Akiwate, B., Parthiban, L. (2021) A DNA Cryptographic Solution for Secured Image and Text Encryption, International Journal of Advanced Computer Science and Applications, 12, https://doi.org/10.14569/IJACSA.2021.0120250
[24] Parsa Sarosh, Shabir A. Parah, Mohiuddin Bhat, Khan Muhammad. (2021) A Security Management Framework for Big Data in Smart Healthcare, Big Data Research, 25, 100225, https://doi.org/10.1016/j.bdr.2021.100225
[25] Pavithran, P., Mathew, S., Namasudra, S. and Lorenz, P. (2021) A Novel Cryptosystem based on DNA cryptography and randomly generated mealy machine, Computer and security, 104, https://doi.org/10.1016/j.cose.2020.102160
[26] Pavithran, P., Mathew, S., Namasudra, S. and Srivastava, G. (2022) A novel cryptosystem based on DNA cryptography, hyperchaotic systems and a randomly generated Moore machine for cyber physical systems, Computer Communications, 188, 1-12, https://doi.org/10.1016/j.comcom.2022.02.008
[27] Namasudra, S. (2022) A Secure Cryptosystem using DNA cryptography and DNA Steganography for the cloud-based IoT infrastructure, Computers and Electrical Engineering, 104, 108426, https://doi.org/10.1016/j.compeleceng.2022.108426
[28] Mukherjee, P., Garg, H., Pradhan, C., Ghosh, S., Chowdhury, S. and Srivastava, G. (2022) Best Fit DNA-Based Cryptographic Keys: The Genetic Algorithm Approach, Sensors (Basel, Switzerland), 22(19), 7332, https://doi.org/10/3390/s22197332
[29] Zhang, K., Tian, J., Han, J., Yuan, Q. (2022) A Lightweight Model-driven MES Simulation Framework Based On Probabilistic Finite Automata, IEEE 7th International Conference on Intelligent Computing and Signal Processing, 1093-1096, https://doi.org/10.1109/ICSP54964.2022.9778640
[30] Bhardwaj, C., Garg, H., Shekhar, S. (2022) An Approach for Securing QR code using Cryptography and Visual Cryptography, IEEE International Conference on Computational Intelligence and Sustainable Engineering Solutions, 284-288, https://doi.org/10.1109/CISES54857.2022.9844332
[31] Lee, B., Lee, I., Kim, M. (2022) Design and Implementation of Secure Cryptographic System on Chip for Internet of Things, IEEE Access, 101, https://doi.org/10.1109/ACCESS.2022.3151430
[32] Sathiyapriya, K., Nirmala, P., Deepika, M., Revanth, J., Saravanan, M., Mohankumar, M. (2022) Depiction of FPGA Based Vending Machine Using Mealy Model, IEEE Third International Conference on Smart Technologies in Computing, Electrical and Electronics, 1-5, https://doi.org/10.1109/ICSTCEE56972.2022.10099962
[33] Anirban Bhowmik, Sunil Karforma, Joydeep Dey. (2022) Symmetric Key and Artificial Neural Network with Mealy Machine: A Neoteric Model of Cryptosystem for Cloud Security, Machine Learning Techniques and Analytics for Cloud Security, 81-101, https://doi.org/10.1002/9781119764113.ch5

[34] Hashem, M. I., Alibraheemi, K. (2022) Literature Survey: Biometric Cryptosystems Based on Fingerprint Processing Techniques, IEEE International Conference on Data Science and Intelligent Computing, 198-201 https://doi.org/10.1109/ICDSIC56987.2022.10076184
[35] Namasudra, S. (2022) Fast and Secure Data Accessing by Using DNA Computing for the Cloud Environment, IEEE Transactions on Services Computing, 15(4), 2289-2300, https://doi.org/10.1109/TSC.2020.3046471
[36] Kim, S. R., Kyung, R. (2023) Study on Modified Public Key Cryptosystem Based on ElGamal and Cramer-Shoup Cryptosystems, IEEE 13th Annual Computing and Communication Workshop and Conference, 0280-0284, https://doi.org/10.1109/CCWC57344.2023.10099297
[37] Fares, N., Wang, B., Bakiras, S. (2023) Design and implementation of certificateless cryptography for IOT applications, IEEE International Midwest Symposium on circuits and systems, 933-937, https://doi.org/10.1109/MWSCAS57524.2023.10405921
[38] Venugopal, L. K., Rajaganapathi, R., Birjepatil, A., Raja, S. E., Subramaniam, G. (2023) A Novel Information Security Framework for Securing Big Data in Healthcare Environment Using Blockchain, Engineering Proceedings, 59(1), 107, https://doi.org/10.3390/engproc2023059107
[39] Fawad Ahmed, Muneeb ur Rehman, Jawad Ahmed, Muhammad Shahbaz Khan, Wadii Boulila, Gautam Srivastava, Jerry Chun-Wei Lin and William J. Buchanan. (2023) A DNA Based Color Image Encryption Scheme using A Convolutional Autoencoder, ACM Transactions on Multimedia Computing Communications and Applications, 19(3s), 1-21, https://doi.org/10.1145.3570165
[40] Gaverchand, K. and Venkatesan, R. (2023) A Novel Approach of 1-D Cellular Automata in Cryptosystem, Mathematical Modelling of Engineering Problems, 10, 2121-2126, https://doi.org/10.18280/mmep.100623
[41] Namasudra, S. (2023) Perspective of DNA Computing in Computer Science, Advances in Computers, Elsevier 129, 1-387, https://doi.org/10.1016/bs.adcom.2022.08.001
[42] Namasudra, S., Kumar, T., Kumar, P. (2023) Providing Data Security using DNA Computing in the Cloud Computing Environment, International Journal of Web and Grid Services. 19(4), 463-486, https://doi.org/10.1504/IJWGS.2023.10060351
[43] Pavithran, P., Mathew, S., Namasudra, S. and Singh, A. (2023) Enhancing randomness of the ciphertext generated by DNA-based cryptosystem and finite state machine, Cluster Computing, 26, 1035-1051, https://doi.org/10.1007/s10586-022-03653-9
[44]
[45] Singh, A., Kumar, A. and Namasudra, S. (2024) DNACDS: Cloud IoE big data security and accessing scheme based on DNA cryptography, Frontiers of Computer Science, 18, 181801, https://doi.org/10.1007/s11704-022-2193-3
[46] Lang, K: 20 Newsgroups. Accessed on December 2023 https://qwone.com/jason/20Newsgroups