

A Major Project-II Report
On
**SYMMETRIC KEY ENCRYPTION BASED ON DNA
CRYPTOGRAPHY**

Submitted in Partial fulfilment of the Requirement for the Degree of
Master of Technology
in
Software Engineering

Submitted By
Anukriti Kaushal
2K17/SWE/05
Under the Guidance of
Ms Divyashikha Sethia

Assistant Professor
(COE Department)



DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahabad Daulatpur, Main Bawana Road, Delhi-110042

June 2019

DECLARATION

I hereby declare that the Major Project-II work entitled “**SYMMETRIC KEY ENCRYPTION BASED ON DNA CRYPTOGRAPHY**” which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of the degree of Master of Technology (Software Engineering) is a bona fide report of Major Project-II carried out by me. I have not submitted the matter embodied in this dissertation for the award of any other degree or diploma.

Place: Delhi

Anukriti Kaushal

Date:

Roll No. 2K17/SWE/05

CERTIFICATE

This is to certify that **Anukriti Kaushal** (2K17/SWE/05) have completed the major II project titled **“SYMMETRIC KEY ENCRYPTION BASED ON DNA CRYPTOGRAPHY”** under my supervision in partial fulfilment of the master of technology degree in software engineering at Delhi Technological University.

PLACE: DELHI

DATE:

SUPERVISOR

Ms Divyashikha Sethia

Assistant Professor

Delhi Technological University

Bawana Road, Delhi -110042

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor **Ms Divyashikha Sethia** for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

Secondly, I am grateful to **Dr. Rajni Jindal**, HOD, Computer Science & Engineering Department, DTU for her immense support. I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Anukriti Kaushal

Roll No – 2K17/CSE/03

M. Tech (Computer Science & Engineering)

Delhi Technological University

ABSTRACT

Nowadays large portion of significant records and personal information transmitted over unreliable channels or present over insecure platform, such as, social application, cloud, remote server. These documents or files can be used by intruder for malicious purpose, in such case we required an encryption scheme that is more efficient and ensure security end to end. The previous works based on DNA symmetric encryption were proposed but the necessity to give exceedingly improved and proficient calculation that give the encryption, steganography for all kind of archive to that it can undoubtedly move and kept over unreliable platform.

Proposed an improved DNA symmetric encryption scheme that provides data encryption and steganography for data hiding in DNA nucleotides so that it can easily transmit over insecure channel and kept over unreliable platform. Also, the key is randomly generating from a large database of NCBI nucleotide hence, enhancing automation for key generation process, the key maintained very high cracking probability as every time generated new 64 length of nucleotide and the algorithm uses Shannon principle of confusion and diffusion protecting documents from numerous possible attacks and efficient in term of execution. This work reduces the AND, or and shift operations used in AES cryptography by using DNA properties.

LIST OF CONTENT

DECLARATION	ii
CERTIFICATE	iii
ACKNOWLEDGMENT	vi
ABSTRACT	v
TABLES OF CONTENT	vi
List of Figures	vii
List of Tables	vii
List of Abbreviations	ix
Chapter 1: INTRODUCTION	1
1.1 Background	1
1.2 Motivation	2
1.3 Problem Statement	3
1.4 Research and objective	3
Chapter 2: LITERATURE REVIEW	5
2.1 Cryptography	5
2.2 DNA Theory	5
2.3 DNA cryptography	6
2.4 Symmetric key cryptography	8
2.5 Types of Attack	9
2.6 Basic Architecture of DNA cryptography	10
2.7 Related work	11
Chapter 3: DNA TABLES AND PROPERTIES	13
3.1 HDNA table	13
3.2 DNA XOR table	14
3.3 DNA inversion	16
3.4 DNA s-box	17

3.5	DNA inverse S-box	18
3.6	D-Fusion matrix	19
Chapter 4	PROPOSED ALGORITHM	21
4.1	Document pre-processing	22
4.2	Key generation	23
4.3	DNA encryption	24
4.4	DNA steganography DATA post processing	31
4.5	DNA decryption	32
Chapter 5:	Security analysis	35
5.1	Cracking probability	35
5.2	Security	37
Chapter 6:	Results and Performances	38
6.1	Time and complexity analysis	38
6.1	Experimental results	42
Chapter 7:	Conclusion and future work	47
References		x

LIST OF FIGURES

1	Architecture of DNA cryptography	10
2	DNA matrix for encryption process	19
3	The 16 possible DNA XOR positions	30
4	flow diagram of proposed algorithm	21
5	Example of key representation in a matrix	24
6	Division matrix into 4 coordinates UL, UR, LL and LR	25
7	representation of redundant XOR result region	26
8	case1 for upper left UL coordinates	27
9	case2 for upper right UR coordinates	28
10	case3 for lower left LL coordinates	29
11	case 4 for lower left LL coordinates	30
12	The stages of adding guessing probability	35
13	Average encryption time of document files	44
14	Changing probability of document files	45
15	graph of comparison table	46

LIST OF TABLES

1	Amino acid	6
2	Dictionary rule table	7
3	HDNA-DNA table	13
4	DNA XOR table	14
5	DNA S-box table	17
6	AES and proposed work encryption comparison	42
7	AES and proposed work decryption comparison	42
8	The execution time, average execution time and changing probability	44
9	comparison table of average encryption time of silent mutation and proposed work	45

LIST OF ABBREVIATIONS

DNA_{seq_1}	Fixed size sequence of nucleotide
A	DNA Base stands for Adenine
C	DNA Base stands for Cytosine
G	DNA Base stands for Guanine
T	DNA Base stands for Thymine
$(DNA_{seq_1} \oplus DNA_{seq_2})$	nucleotide wise XOR between 2 DNA sequences
$a \oplus b$	Bitwise XOR operation between two binary strings
HDNA	hexadecimal to DNA sequences mapping table
DNA s-box	DNA substitution box
DNA reverse s-box	DNA inverse substitution box
D-fusion matrix	DNA diffusion matrix
UL	Upper left coordinate of matrix
UR	Upper right coordinate of matrix
LL	Lower left coordinate of matrix
LR	Lower right coordinate of matrix
D_{N_b}	Number of DNA blocks

D_{N_k}	Number of key for rounds
D_{N_r}	Number of DNA rounds
DNA_{block}	DNA block of size 4 or 64
$D(i,j)$	DNA quadruple position in 4x4 matrix
$N(\text{base})$	Counting of base individually
GF	stands for Galois field Arithmetic
$P_{changing}$	Changing probability of DNA sequences
ET	Execution time
$p(\text{starting})$	Probability of finding starting position
$p(e)$	Probability conversion of bits to nucleotide
$p(k)$	Key guessing probability
$p(S)$	Probability of successful guessing
Avg_{ET}	Average execution time

CHAPTER 1: INTRODUCTION

With the moving advances in the field of DNA cryptography, numerous speculations directed for actualising DNA computers which are DNA based and appropriate natural capacities to achieve the ideal tasks. DNA cryptography is a cryptographic field that emerged with the examination of DNA computing in which DNA is used as data transporter also the modern natural innovation utilised as a usage instrument [1]. The high parallelism, extraordinary vitality productivity, and unusual data thickness characteristic in DNA sequences investigated for cryptographic purposes such as encryption, authentication, signature, thus on. As from the synthetic DNA capability of storing information and high durability give attention to the digital DNA system. Also, one of the properties of the DNA sequences is to manufacture or amplify into desirable length this property prepared for the efficient encryption process.

1.1 Background

In 1994, 1st DNA computation to solve the NP-complete problem, namely Directed Hamilton Path Problem with seven nodes by basically manipulating the DNA strands. The cryptanalysis of DES algorithm using Bio-molecular computing [13]. The Steganographic area of secret messaging by first converting the original message into DNA form and then hiding it between random DNA strands.

Classification of DNA in four parts:

- Symmetric DNA cryptography using one-time pad sequence
- Asymmetric DNA cryptography using the hybridisation method.
- Pseudo-DNA cryptography
- DNA Steganography

By using DNA, which is very dense, we can store approximately 400 million GB data in just a small microchip, and the DNA is very durable; its span life is over 500 years. It is also possible that one day can store all information currently on the internet into the space of a single laptop.

In 1999, the scientist of New York published an article that protects messages from espionage by first applying DNA cryptography and then concealed with a sample of microdots [12]. In this, every English alphabet, punctuation symbols, and numbers were

encoded using 64 possible DNA codons one specific for each. Then they spliced a 22-character message into a long strand of DNA and surrounded it with specific genetic markers. Further, the DNA encoding applied for binary sequences 0's and 1's using DNA codons. Digital data encoding and programmed into synthetic DNA and then decoded back into its original binary form. A breakthrough, 13 years later in 2012 scientist in the UK, encoded 739 kb of computer files into DNA strands. It contained all 154 Shakespeare sonnets, and an excerpt from Martin Luther King's, " I have a dream speech". Four years later, Microsoft and the University of Washington were able to encode 200 MB of data [14].

1.2 Motivation

These days a large portion of valuable records and personal information stored over the insecure platform, for example, social application, cloud, remote server. In such a case, we required an encryption scheme that is more efficient and ensures security. The cryptography algorithm such as AES, DES is extensively used for encrypting a large mass of data. A Conventional cryptographic framework has great inheritance and occurs based on a solid numerical and hypothetical premise. Conventional security frameworks like RSA, DES or NTRU, are too found activities continuously. Along these lines, a significant discernment built up that the DNA cryptography is not to refute the convention, yet to make an extension between existing furthermore, innovation. DNA cryptography executed by utilising modern organic strategies as apparatuses what is more, vital difficult issues as a fundamental security premise to thoroughly apply the extraordinary points of interest. The intensity of DNA cryptography will fortify the current security frameworks by opening up a new probability of a half and half cryptographic framework.

In the literature overview, numerous DNA symmetric key cryptography calculations proposed where paper [1], [2] pursue the comparable advance of AES for encryption, in paper [4] appropriate natural interpretation, reversal and translation process for encryption. Some utilisation of the DNA groupings to shroud figure content [8]. The necessity to give exceedingly improved and proficient calculation that gives the encryption, steganography

for all kinds of the archive to that it can undoubtedly move and kept over the unreliable platform.

1.3 Problem statement

These days a large portion of information records, reports, individual data, and pictures are put away and required to transmit. The transmission procedure is the real test, even though the scrambling calculation is ideal, yet there is a dependably probability of assault. In such a case, we required an increasingly proficient encryption plot and guarantee security. The cryptography calculation, for example, AES, DES is generally used for scrambling tremendous measures of information. We required flexible encryption that encrypts the large volume of data in a few minutes as well as protected data from intruder and transmitted over insecure channels. The past work [1], [2] on DNA symmetric encryption make such a large number of moves to attain a similar dimension of security as in AES effectively present additionally, required the framework becomes increasingly complex and required more execution time and some work are insufficient in providing all security aspects in papers [5], [14].

1.4 Research and objective

In the thesis, the objective is to give an efficient DNA symmetric encryption technology that provides data encryption and steganography so that applied to various type of document of any size. Therefore, proposed an Advance DNA symmetric encryption (ADSE), combined encryption and steganography techniques to hide data into DNA sequences.

Also, it expected more automation in the encryption process, so the involvement of the user is less required, and system efficiency increased. In the ADSE algorithm, the key is randomly generating from an extensive database of NCBI automatically during the encryption process such that the encryption algorithm has a high cracking probability.

Moreover, introduced HDNA mapping that converts the original form in into DNA sequences encoding. Likewise, Effectively use of DNA properties to reduce steps of actual work. In ADSE, DNA-XOR property is used to give a diffusion principle and produce a

highly shuffled matrix by applying the proposed D- fusion algorithm. Results of performance time and comparison with previous work given in section 7.1 and 7.2.

CHAPTER 2: LITERATURE REVIEW

2.1 Cryptography

Each work needs evidence, and the verification given by the Documentation procedure, a few archives are open and can be composed, get to and adjusted by anybody, yet are private and required confirmation before getting. There are a few interlopers that are consistently endeavouring to access such records used for the wrong aim. Cryptography is a mechanism that gives a component to verify information exchange over the unbound channel. It encodes the information in a configuration that does not bode well and must decide by the individual who has permitted to get to or can ready to verified for getting to the information. Beforehand, cryptography was relying on its algorithm, yet soon it was demonstrated that even a calculation is not verifying there is a prerequisite for the key. Key can be private or open, yet the scrambled information is relying on algorithm alongside key. The cryptography partitioned into two symmetric key cryptography and asymmetric key cryptography.

2.2 DNA Theory

DNA is a molecule this is Deoxyribonucleic acid, in the cellular offers double stand DNA wherein the double stand interconnects to shape double helix-like structure additionally called B-shape DNA. On simplifying to look chemical structure, here every position is a polynucleotide. A nucleotide has three additives five-carbon sugar, phosphate institution and four viable nitrogenous bases which can be Adenine [A], Guanine[G], Cytosine[C], and Thymine[T]. The nitrogenous bases usually connected to the 1' prime carbon of sugar and counted from there it is seen that there is phosphate between the 5' carbon of the one sugar and 3' carbon of the neighbouring sugar. The sugar is referred to as deoxyribose due to this nucleotide in DNA and Deoxyribonucleic acid is referred to as deoxynucleotide nucleotides connect to each different in DNA stands through a phosphodiester bond. On explaining that makes pinnacle stand orientation 5'-3' prime (Watson) and bottom stand from 3'-five' top (crick). Nucleotides come together via covalent bonds in the backbone. The two DNA strands have interaction with non-covalent hydrogen bonds among the given bases A, G, C, and T. Each

base form a couple of hydrogen bonds with its inverse base inside the different base bond sequences together by using a hydrogen bond each unit is known as a base pair. Adenine pairs with Thymine and Guanine pairs with Cytosine.

Properties of DNA molecule: DNA is composed of two identical Nucleotides stands where each is complementing each other. Replication is the process of spiting this stands so; either side could be used to construct the other side. Their present RNA that stands for ribonucleic acid, where RNA is plays a role of messenger between a particular section of DNA called mRNA. So, that translated into an actual protein. The process of making RNA from DNA is called Transcription, which is similar to replication where Adenine pairs with Uracil(U).

An inverse T

C inverse G (1.2)

Proteins are made from a series of Amino acids wherein three bases codons are used to code into particular amino acid there are sixty-four possible codons and 20 possible amino acids given in table 1.

Amino acid	Codons
Isoleucine	ATA, ATC, ATT
Leucine	CTA, CTC, CTG, CTT, TTA, TTG
Valine	GTA, GTC, GTG, GTT
Phenylalanine	TTC, TTT
Methionine	ATG
Cysteine	TGC, TGT
Alanine	GCA, GCC, GCG, GCT
Glycine	GGA, GGC, GGG, GGT
Proline	CCA, CCC, CCG, CCT
Threonine	ACA, ACC, ACG, ACT
Serine	AGC, AGT, TCA, TCC, TCG, TCT
Tyrosine	TAC, TAT
Tryptophan	TGG
Glutamine	CAA, CAG
Asparagine	AAC, AAT
Histidine	CAC, CAT
Glutamic acid	GAA, GAG
Aspartic acid	GAC, GAT
Lysine	AAA, AAG
Arginine	AGA, AGG, CGA, CGC, CGG, CGT

Table 1. of Amino acid [2]

2.3 DNA Cryptography

DNA cryptography is the brand-new developing encryption technique inside the cryptography,

as DNA gives the foreign nation of records safety and protection through contending with the high-quality cryptography calculation, for example, AES, DES or RSA. The little bit of leeway is that DNA cryptography is less puzzling and excessive versatile that provides calculation only as natural security.

In the data science, the superior coding technique paired computerised coding, which is anything that may be encoded by two states zero or 1 and a combination of zero and 1. DNA represents deoxyribonucleic corrosive; the structure of DNA includes a significant degree of genetic data structure skin tone to eye shading it custom designed our whole frame.

As some distance as paired, we can communicate with particles as zero's and 1's. As we have four debris, can speaks to using 2 bits 00,01,11,10 to offer 4! Conceivable DNA designs [5].

As consistent with the professionals the DNA cryptography is characterised into four sub-elements:

1. Symmetric key DNA cryptography (as soon as cushion succession)

2. Deviated key cryptography (mixture method)

Three. Pseudo-DNA encryption

4. DNA steganography

The DNA cryptography is available in numerous versions, and a definitive association is the DNA successions this is something however robust to keep an alternate over the machine. We have grouped the roles and use of DNA sequencing in two types

1. DNA encoding and disentangling

2. DNA stockpiling and packed file age

In Digital DNA there is a mapping exist among the DNA succession, and double bits and this mapping are depending upon the analyst that fluctuates calculation to calculation, one such desk is given in Table 2 [11].

Bits	0	1
0	A	G
1	C	T

Table 2 Dictionary rule table [11]

Additionally, how DNA information put away and safely transmitted between information sender and the recipient are depicted in figure 1 preceding this, there is a standard validation required between the sender and the collector. This procedure finished by considering every conceivable weakness resents amid correspondence. The information transmitted over uncertain channel and interloper has full access over the channel. This DNA cryptography provides the biological complexity as the data hidden inside the DNA nucleotides and DNA can store more abundant information.

2.4 Symmetric key cryptography

The symmetric key is as the name propose is that the equivalent key utilised for both encryption and decoding. A key is trade over a protected channel and encodes the information with the assistance of the emitting key, and the ciphertext can transmit freely over the insecure channel. In Asymmetric key cryptography, there are two keys utilised one for encryption that is open key, and the other used for Decryption that is a private key as the Asymmetric key relies on the long scientific calculation, so it is moderate as the contrast with symmetric key cryptography. The symmetric key gives secrecy to the enormous measure of information and execution is high. Commonplace calculations for Symmetric key cryptography are RC2, AES, DES, 3DES, RC4.

The symmetric key cryptography consists of five main parameters are Message (M), ciphertext (C), key (K), the encryption algorithm (E), and Decryption algorithm (D). The M and K is input to the E system, and then output is the C, now for the decryption process input is C and K in system D and the output is M. in both cases the universal input is K, so the M, C, D, E is public, and key must be private in order to preserve secrecy.

The process of Encryption given as:

$$E(K,P)=C$$

The decryption process is shown as:

$$D(K,C)=P$$

Block cipher task modes concerning various sort of Data we required distinctive kind of handling, there are different block cipher operation modes are given underneath:

- 2.4.1 **Electronic codebook mode (ECB):** this is the least complex model where the manifest content is separate into the fixed-size block then the same key is utilised to scramble each block. This mode is valuable for the small size of information; however, not for the massive size of data that have rehashed block, produce the same cipher content.
- 2.4.2 **Cipher block chaining mode (CBC):** This mode defeats the security issues in the ECB mode, here the initialize vector (IV) is used that is first XOR with the first block of plain text and after that next plain content block is XOR with the previous cipher content block (note: blocks fixed in size) before applying key.
- 2.4.3 **Cipher feedback mode (CFB):** There is a register of size n is present that initially contains the initialise vector and applies the algorithm along with a key to produce ciphertext. The size of the block fixed of size s where $1 < s < n$. Then from the first step choose s bits and then XOR with the block size of s plain text, then feed s cipher bits in a register after shifting bits to the left. Repeat the step for all other rounds during the encryption process until the last plain text block lasts. Also, it uses the same encryption algorithm for decryption.
- 2.4.4 **Output feedback mode (OFB mode):** It is similar to CFM the difference is the output from the encryption is feed to the register.
- 2.4.5 **Counter mode (CTR):** here counter, and the nonce is present; the counter is similar to the size of the plain block, apply the encryption. The counter is incremental, and the nonce sets the increment value.

2.5 Type of attacks

Numerous attacks are conceivable to get the understandable content from the ciphertext.

- 2.5.1 **Known plain content attacks:** In this, the interloper knows some piece of the plain content and furthermore the ciphertext and ready to guide and locate the total plain material.
- 2.5.2 **Chosen cipher attacks:** Here the invader sends a message to sit tight for the ciphertext

from the collector then investigates the cipher content from the known plain content and known cipher message, this attack is conceivable when a similar key utilised encode different messages moreover.

2.5.3 **Brute force:** In this savage power is utilised to discover key by applying all conceivable keys; this assault is conceivable when the key size is little.

2.6 Basic Architecture of DNA cryptography

As other cryptographic calculation, there is the trading of the figure messages among the sender and collector through an insecure channel. The trade depicted beneath:

- Encryption begins with the trading of two groundworks (forward and turn around) among Alice and Bob using a secure channel.
- For Encryption, pre-handling should be possible, that the entire calculation like RSA can be connected first.
- Then cipher content can be changed over into DNA arrangement by the coding plan.
- By playing out this, an extraordinary ciphertext can acquire. In writing, cipher DNA alludes to the term figure content, which is a DNA arrangement, and plain content DNA means the understandable content which is as DNA.

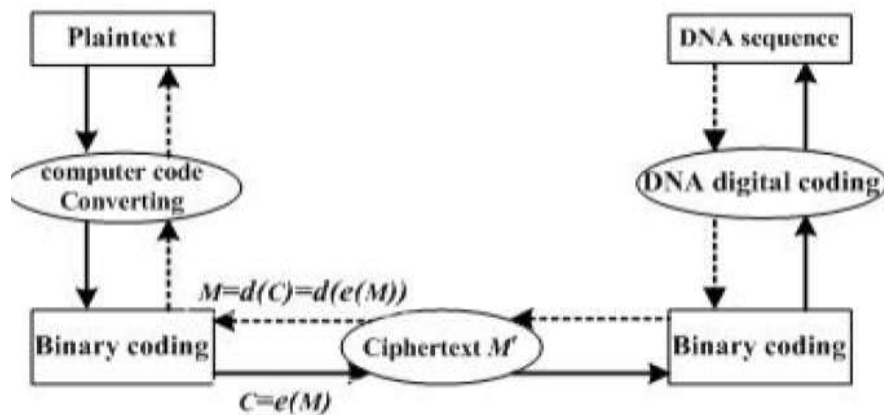


Figure 1 Architecture of DNA cryptography [16]

CHAPTER 3: RELATED WORK

Sabry et al. [1] perform all the traditional steps of AES in terms of DNA sequencing for preserving security also proposed S-box for DNA substitution. The drawback in the proposed system is Involves many steps to Attain security. S Basu set al [4] presented a Bidirectional Associative Memory Neural Network (BAMNN) for key generation and encrypted using a simple DNA process of replication, translation, and transcription. In this paper, the BAMNN is costly for a lesser number of Byte data in terms of memory. The cracking probability is $1/(n * 2^{16})$ where n is the number of blocks, several keys generated depend upon the size block data.

In the paper, Increasing robustness of data encryption standard by integrating DNA cryptography proposed a new cryptography algorithm that is a combination of DNA and DES. This proposed method is more powerful than the traditional DES and provides security as 3DES.

Yushu Zhang et al. [5] is the image encryption based upon the chaos system, which is similar to the cryptography algorithm, and that is iterative. However, the proposed system is not secure from brute force attack due to is a lesser cracking probability. The DNA cryptography is stronger in the field of image encryption as the previous block symmetric cipher algorithm are lesser robust for image encryption.

DNA steganography is likewise a developing examination point where the data can be stowed away into the DNA grouping. The branch of NCBI has given 163 million DNA nucleotide openly, so the concealing information inside the DNA groupings gives security of information as the breaking likelihood given below.

$$p(DNA_{ref}) = \frac{1}{1.63 * 10^8} \quad [7] \quad (3.1)$$

Ghada Hamed et al. [6] proposed an algorithm for hiding data inside DNA sequences also give two levels of security by encrypting using Playfair cipher. Also given the comparison table of various DNA steganography techniques.

Malathi pa et al. [8] is paper for based on DNA steganography and proposed an algorithm for that encodes plain text into DNA sequences, preserving security by inversion, complementary and substitution method. Having high cracking probability and given performance analysis by calculating payload, capacity, and BPN.

From the above-related work, we know that the DNA is a vast concept so as DNA cryptography, some algorithms based upon the DNA biological process some are based on hybrid DNA cryptography where DNA mixed with Another modern cartographic algorithm, sometimes it is emerging in terms of steganography. It used for implementing Symmetric key encoding or Asymmetric key cryptography. Sometimes it is proved useful for encrypting Image data. From the studies, it concluded that we required a DNA cryptography algorithm that more secure and restricted all possible attacks, able to encrypt all size and type of data. Also, it is fast and robust.

From the previous research work DNA work is versatile and flexible that used for steganography, cryptography, image encryption, data compression using Huffman encoding, solve NP-complete problems. The comparison paper of the previous related work is given below in table 3. The papers are shown the symmetric encryption on data, the system cracking probability of the work, attacks possible where Attack 1 chosen plain text attack, attack 2 is ciphertext-only, attach 3 is ciphertext attack and attack 4 is differential cryptanalysis attack. Also, some papers compared and given results of the encryption time also defined in the paper. Now, the requirement is to produce a standard algorithm that secures from the possible defined attacks, able to encrypt all kinds of document files, in less execution time.

CHAPTER 4: DNA TABLES AND PROPERTIES

4.1 HDNA table

Previously in papers [1], [2], [5] the mapping between bits and DNA sequences are done using dictionary rule that is 00 is mapped to A, 01 mapped to G, 10 mapped to C, 11 assigned to T given in table 2. In the ADSE work using an HDNA table that mapped hexadecimal number to DNA sequences. The "Hexadecimal" framework utilises the Base of 16 frameworks and is a well-known decision for speaking too long parallel qualities because their organisation is very minimal, it is fast and flawless a lot clearer contrasted with the long double strings of 1's and 0's. The proposed Hexadecimal table given in table 3 is more rapid in the conversion of plain text to DNA sequence also do not base upon the dictionary rule but, the cracking probability is the same as Dictionary rule.

Hexadecimal number	Nucleotides
0	AA
1	AG
2	AC
3	AT
4	GA
5	GG
6	GC
7	GT
8	CA
9	CG
A	CC
B	CT
C	TA
D	TG
E	TC
F	TT

Table 3. DNA-DNA table

4.2 DNA XOR table

XOR is a Boolean operator that stands for exclusive or. It follows some mathematical properties apply to two samples of inputs to produce output are commutative, identity, self-inverse and associative. It vastly used in the cryptography algorithm due to its high-speed computation and easy hardware implementation. The DNA XOR shown in paper [3], follow the bitwise property XOR given in table 4.

\oplus	A	C	G	T
A	A	C	G	T
C	C	A	T	G
G	G	T	A	C
T	T	G	C	A

Table 4. DNA XOR table[3]

Here the conditions followed to give DNA XOR are given below:

if the XOR with A then result would be the same

if the XOR with T then result would be inverse

if the XOR with the between same nucleotides then result would be A

If the XOR with the between inverse nucleotides then result would be T.

By following the above rules that follow commutatively, identity, self-inverse; and associative in terms of DNA XOR table is produce given in table 3 above.

Example:

INPUT 1	INPUT 2	OUTPUT (\oplus)
AGCC	CCGT	CTTG

4.2.1 Proof of the properties of DNA XOR given below:

4.2.1.1 DNA XOR property 1 "commutative."

$$DNA_{seq_1} \oplus DNA_{seq_2} = DNA_{seq_2} \oplus DNA_{seq_1} \quad (4.1)$$

Example:

INPUT 1	INPUT 2	OUTPUT (\oplus)
AGGT	GGCC	GATG
GGCC	AGGT	GATG

4.2.1.2 DNA XOR property 2 "Associative"

$$(DNA_{seq_1} \oplus DNA_{seq_2}) \oplus DNA_{seq_3} = DNA_{seq_1} \oplus (DNA_{seq_2} \oplus DNA_{seq_3}) \quad (4.2)$$

Example:

INPUT 1	INPUT 2	INPUT 3	OUTPUT (\oplus)
AGGT	GGCC	CTCT	GATG
GGCC	AGGT	CTCT	GATG

4.2.1.3 DNA XOR property 3 "Identity element"

Here considers "A" as 0 and "T" as 1

$$DNA_{seq_1} \oplus A = DNA_{seq_1} \quad (4.3)$$

Example:

INPUT 1	INPUT 2	OUTPUT
A	A	A
G	A	G
C	A	C
T	A	T

4.2.1.4 DNA XOR property 4 “Self-inverse”

$$\text{DNA}_{\text{seq}_1} \oplus \text{DNA}_{\text{seq}_1} = A \quad (4.4)$$

Example

INPUT 1	INPUT 2	OUTPUT
A	A	A
G	G	A
C	C	A
T	T	A

4.3 DNA inversion

It far primarily based upon the organic system that was in double helix stand based totally upon the more than one hydrogen bond between the two complementary bases. As in binary bits like 1 inverse is 0, in chance True inverse is False. There are existing four bases, so from the biological conclusion, the inversion of A is T and G is C additionally given in equation:

$$A \Leftrightarrow T \quad G \Leftrightarrow C \quad (4.5)$$

4.4 DNA S-box

It is simpler put into effect cryptography rounds consisting of transposition, transferring row, mixing column, XOR, Add spherically. However, substitution is more enormous complex as, required as-box for DNA Sequences. The S-field utilised in AES cryptography as a way to keep Shannon confusion property. In paper [1], given DNA s-container this is inspired by the AES s-field, in which 256 possible substitutions which are mapping among the hexadecimal numbers. Here within the ADSE algorithm the constitute chunks of four nucleoids, the DNA s-box starting from left the two used to map the row and the following two to map column then the block perpendicularly bisects the row and column is the result and replaceable for the given bite of nucleotides.

The given S-container is maintaining the property of Shannon Confusion and additionally adjustments the nucleotides to maintain the substitution; this manner is maximum green substitution as already seen in AES cryptography. In the ADSE method every matrix contains 128 bits this is the same to the 64 nucleotides, every conceals two binary bits. The DNA s-container is given below in table 5.

		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
		AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU
0	AA	CGAU	CUJA	CUCU	CUGU	UUAG	CGGU	CGUU	UACC	AUAA	AAAC	CGCU	AGGU	UUUG	UCCU	GGGU	CUCG
1	AC	UAGG	GAAG	UAGC	CUUC	UUGG	CCGC	CACU	UUAA	GGUC	UCCA	GGAG	GGUU	GCUA	GGCA	CUAG	UAAA
2	AG	GUCU	UUUC	GCAU	AGCG	AUCG	AUUU	UUUC	UAUA	AUCA	GGCC	UGCC	UUAC	CUAC	UCGA	AUAC	ACCC
3	AU	AACA	UACU	AGAU	UAAU	ACGA	GCCG	AACC	GCGG	AACU	ACAG	GAAA	UGAG	UGGU	AGCU	GUAG	CUCC
4	CA	AAGC	GAAU	AGUA	ACGG	ACGU	CGUG	CCGG	GGAA	CCAG	AUGU	UCCG	GUAU	AGGC	UGAU	AGUU	GACA
5	CC	CCAU	UCAC	AAAA	UGUC	AGAA	UUUA	GUAC	CCGU	CGGG	UAGU	GULG	AUGC	CAGG	CAUA	CCGA	UAUU
6	CG	UCAA	UGUU	GGGG	UUGU	CAAU	CAUC	AUAU	GACC	CACC	UUGC	AAAG	CUUU	CCAA	AUUA	GCUU	GGGA
7	CU	CCAC	GGAU	CAAA	GAUU	GCAG	GCUC	AUGA	UUCG	GUUA	GUCG	UCGG	AGAC	ACAA	UUUU	UUAU	UCAG
8	GA	UAUC	AAUA	ACAU	UGUA	CCUU	GCCU	CACA	ACCU	UACA	GGCU	CUUG	AUUC	CGCA	CCUC	ACGC	CUAU
9	GC	CGAA	GAAC	CAUU	UCUA	AGAG	AGGG	GCAA	GAGA	CACG	UGUG	GUGA	ACCA	UCUG	CCUG	AAGU	UCGU
a	GG	UGAA	AUAG	AUGG	AAGG	CAGC	AACG	AGCA	CCUA	UAAG	UCAU	GGUA	CGAG	GCAC	GCCC	UGCA	CUGC
b	GU	UGCU	UAGA	AUCU	CGUC	GAUC	UCCC	CAUG	GGGC	CGUA	CCCG	UUCA	UGGG	CGCC	CUGG	GGUG	AAGA
c	UA	GUGG	CUGA	AGCC	AGUG	ACUA	GGCG	GUCA	UACG	UGGA	UCUC	CUCA	ACUU	CAGU	GUUC	GAGU	GAGG
d	UC	CUAA	AUUG	GUCC	CGCG	CAGA	AAAU	UUCG	AAUG	CGAC	AUCC	CCCU	GUGC	GACG	UAAC	ACUC	GUCG
e	UG	UGAC	UUGA	GCGA	ACAC	CGGC	UCGC	GAUG	GCCA	GCGU	ACUG	GACU	UGGC	UAUG	CCCC	AGGA	UCUU
f	UU	GAUA	GGAC	GAGC	AAUC	GUUU	UGCG	CAAG	CGGA	CAAC	GCGC	AGUC	AAUU	GUAA	CCCA	GUGU	ACCG

Table 5. DNA S-box table [1]

4.5 DNA -inverse S-box

The inverse DNA s-box matrix is to Decrypt the data documents. In the proposed set of rules, the inverse DNA s-field has given in [1] used and necessary with a view to back up the authentic shape. The DNA encoding primarily based on a symmetric block cipher, the decryption technique is simply the opposite of the encryption procedure, so there is inverse s-box needed. At this stage of Decryption, we get the original nucleotides in the shuffling format.

4.6 D-Fusion matrix

The new diffusion matrix is proposed to preserve the Shannon diffusion assets, wherein the fusion matrix is the mixture of the moving and the XOR operation. In many blocks, ciphers observe row transferring and column blending for transposition. The ADSE algorithm primarily based on DNA bases, and the XOR characteristic between the two unusual positions of the 4x4 matrix provides the fusion in addition to the row shift, and column mixing called as D-fusion matrix.

For 16 positions in DNA $M[4 \times 4]$ matrix, the 16 possible cases are given below, where each DNA position is replaced by performing XOR operation between the given two distinct position DNA nucleotide and gives a complete shuffling along with substitution by performing a fusion of matrix. If the XOR operation of the given two other position DNA nucleotide is the same with the current position DNA nucleotide, then it will replace with its inverse to maintain changing in every position nucleotide. As DNA is a double helix structure that stores A inverse is T and C inverse is G, from the DNA inversion property.

This single step combines the multiple steps of row rotation, column mixing and Add the round constant of AES cryptography. The matrix is 4×4 where each cell stores group of 4 nucleotides, the resultant matrix contains $4 \times 4 \times 4 = 64$ nucleotides.

Represent 64 nucleotides as 4x4 matrix and represents the position as i, j



Such that:

$$a_{ij}, \text{where } i \in [1, 4], j \in [1, 4] \quad (4.6)$$

D(1,1)	D(1,2)	D(1,3)	D(1,4)
D(2,1)	D(2,2)	D(2,3)	D(2,4)
D(3,1)	D(3,2)	D(3,3)	D(3,4)
D(4,1)	D(4,2)	D(4,3)	D(4,4)

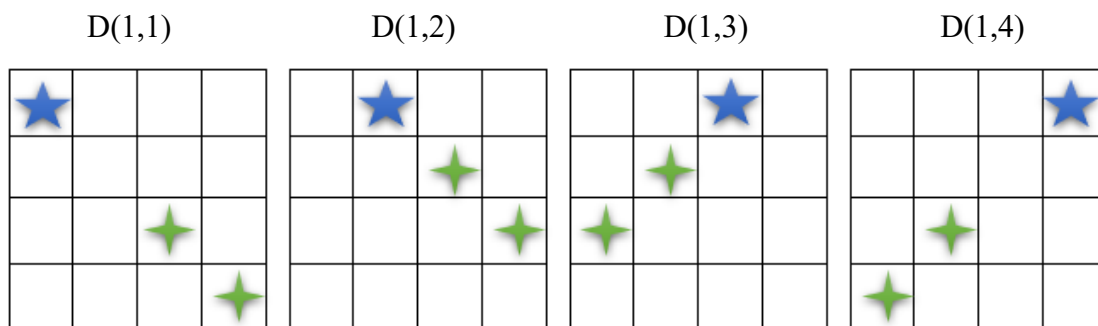
Figure 2 DNA matrix for the encryption process

Starting from the very first position it than the possible case to produce distinct XOR inputs position to give a rotation without any change in nucleotides. The possible case for each cell given below:

-  Represents the diffusion position
-  Represents the two distinct possible XOR input for diffusion

There are sixteen possible distinct cases for each cell, and the XOR between two green stars replaces the current cell that is the blue star, always take the lower (i,j) position before higher in XOR function.

The final result is the transposition matrix using the D-fusion matrix.



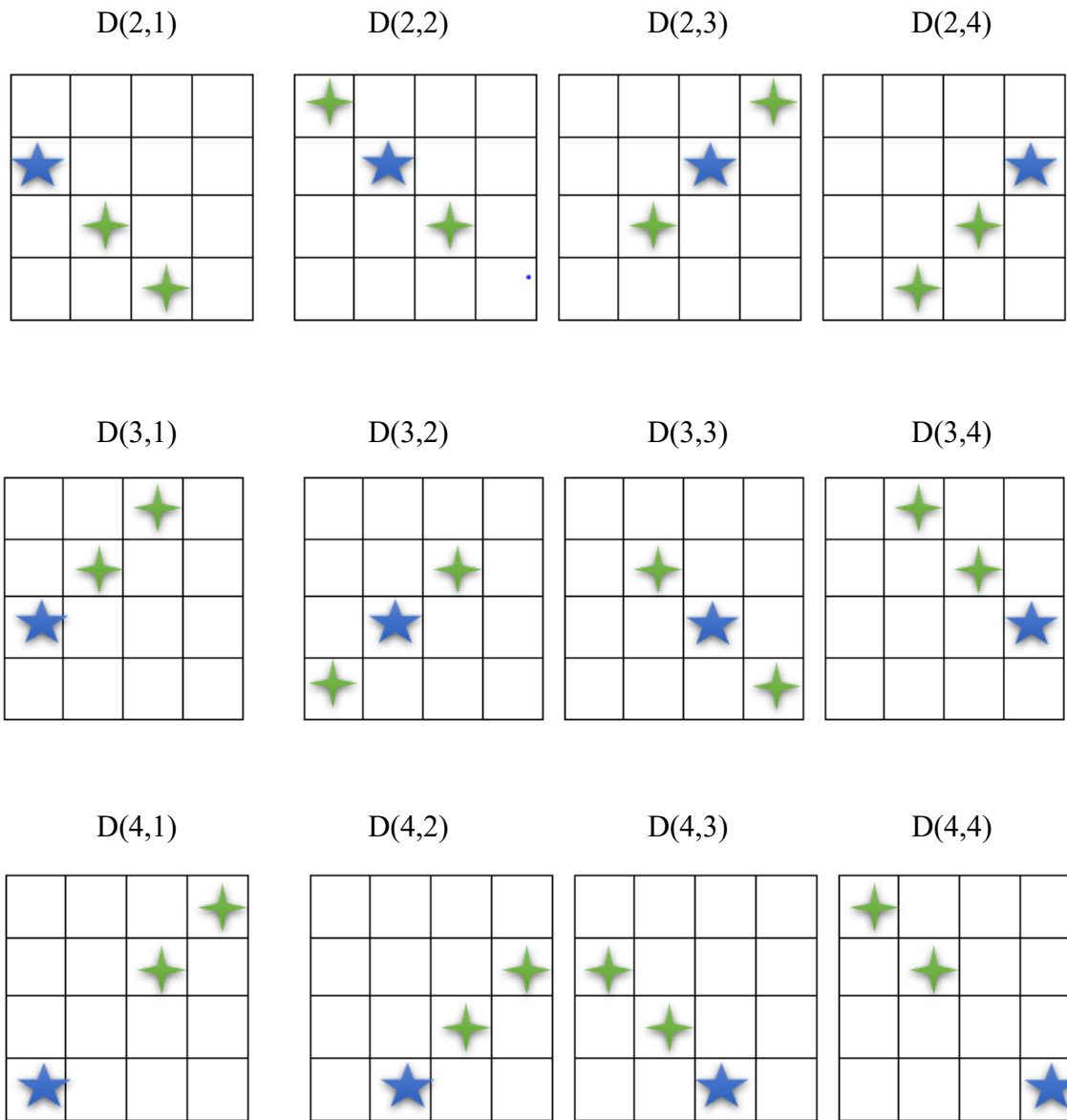


Figure 3. The 16 possible DNA XOR positions

CHAPTER 5: PROPOSED ALGORITHM ; Advanced DNA symmetric encryption (ADSE)

In this section, each module of the proposed work explained in details; the system comprises five steps explained in section 5.1, 5.2, 5.3, 5.4, and 5.5. The ADSE is implemented based on a key of length 128 bits in CBC mode as the 128 bits are converted into DNA sequences, where every nucleotide hides 2 bits, so we required a sequence of 64 nucleotides for the key process. For every encryption process, there are 10 rounds which are sequentially taking location. The variety of rounds in that given by the formula 1.

$$D_{N_r} = \frac{(2 \times D_{N_k})}{32} + 6 \quad (5.1)$$

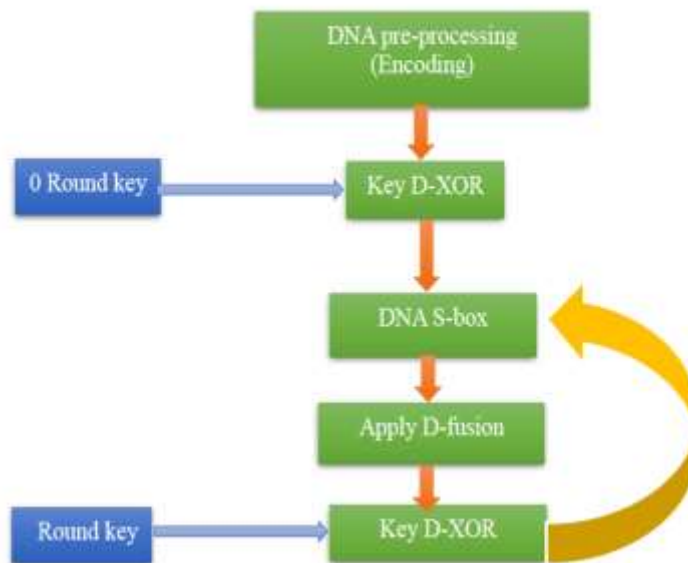


Figure 4. flow diagram of the proposed algorithm

Proposed Algorithm

The proposed work is having five main steps are given below and explained in details:

5.1 Document pre-processing

Initially, the documents are digitally present in binary bits now; and the role is to represent it in the form of DNA bases that are A, G, C and T. each base uniquely replaces a byte. Here the dictionary rule is not used for mapping instead, using an HDNA table, here the bits are converting into Hexadecimal form, and DNA mapping is done using the HDNA table. Here for single byte is replaced with two DNA nucleotides or each DNA hides 4 bits that are 4bits/nucleotides.

STEP 1: Preprocessing

Input: A [plain data]

Algorithm Body:

1. Read the plain text A.
2. Convert plain text into binary bit sequence B.

$$B \rightarrow \text{BINARY} [A]$$

3. Convert binary bit sequence into hexadecimal format sequence.

$$H \rightarrow \text{HEX} [B]$$

4. Convert hexadecimal into DNA sequence using proposed HDNA table.

Output: $D \rightarrow \text{DNA sequences} [H]$

End Algorithm.

This process based on reading the whole file bits converting it into DNA, here the time complexity used is $O(n)$, and space complexity is $O(1)$ but using the best algorithm to implement this.

5.2 Key generation

This manner is impartial from the user, like different cryptography set of rules the user wants no more extended input the KEY, the key generated from the unusually massive NCBI dataset of nucleotides and this is publicly available. The key is not always directly using in this it going via some processing before its use so, although it is publicly available, it is cracking probability may be very excessive.

The key is a part of the DNA sequence file randomly downloaded from the NCBI database. A random 64 nucleotides are selected from the loaded file to represent 128 bits of the key.

Dataset: collect the dataset from NCBI, the nucleotides.

The randomly downloaded data set (FASTA) that is used to generate nucleotides; on average, a FASTA file contains $3 * 10^6$ characters. Let the number of characters in a file s then the random number generator is used that generates a random number

R such that $R \in \{1, 2, \dots, s - 63\}$ set. From s starting from R select the 64 nucleotides in a sequence that used for a key for the encryption process.

Now the key is generated for every round like in symmetric block cipher cryptography; there are 10 rounds for 128bits that is 64 nucleotides. This process uses Shannon confusion diffusion belongings, which applied to the usage of DNA s-container and Diffusion conditions. DNA XOR is used to give diffusion and rapid shuffling.

This required to convert the key into 4x4 matrix, in this process first divide the nucleotides into chunks of 4 nucleotides and then represent this to fill single cell of the matrix, now the contain 64 nucleotides as 4 for group, 4 in each row and 4 in each column that is $4 \times 4 \times 4 = 64$ nucleotides, example given in figure 5. Now the key is processed for each round and the steps are given below:

AGTA	GTAA	GGGG	TACA
AGTT	CTAG	CTAA	GCTA
GTAA	CATC	CCTC	CCCA
GATT	CCAT	TTAC	TTAA

Figure 5. Example of key representation in a matrix

STEP 2: key generation

Input: KEY [64 nucleotides]

Algorithm Body:

1. Key in DNA nucleotide into the 4x4 matrix.
2. nucleotides Substitution (DNA S-Box) from table 3
3. Shuffling nucleotides using the proposed diffusion Algorithm.
4. DNA_XOR with the previous cycle key from table 2
5. Repeat step 2-4 for to generate 10 Round Key **Output:** KEY[0]-KEY[10]

End Algorithm.

5.3 DNA encryption:

The process of generating cipher data transmitted to the receiver through an insecure channel. Here the pre-processed data and key are provided as input to required to produce ciphertext. This process includes many steps to retain security and highly encrypted data. This encrypted as a block cryptography algorithm. This process has three functions repeated recursively for 10 rounds, and this implemented in Cipher block chaining mode (CBC) that removes the generation of producing the same cipher block. The encryption processes are *AddDNAKey()Transformation*, *DNAsubstitute() Transformation* and *Diffusion()transformation*.

The function used in the encryption process explained below:

- 5.3.1 **AddDNAKey()Transformation:** this function is similar to Addroundkey() of AES round where the 128 bits of plain text are chosen sequentially and stored in a 4x4 matrix. Now the method is to add the key for the round generated before in key generation process step 2. In the adding round key, there is nucleotide wise XOR between the KEY for the round and the chosen 128 bits of the document. This process is used to protect data from a brute force attack so that the ciphertext transmitted over an insecure channel.
- 5.3.2 **DNAsubstitute() Transformation:** this process is implemented using a DNA s-box that is proposed in by Sabry et al. [1]. A substitution method based on mapping using a table where the 16x16 table of DNA substitution inspired by AES s-box.
- 5.3.3 **Diffusion() transformation:** The diffusion is based on conditions of the 16 conditions to be replaced by the XOR of two other cells. Here the matrix is divided into coordinates L for left and R for right. The Left is divided into two parts the upper left UL and lower left LL similarly; the right divided into two parts the upper right and lower right. Now the four conditions are such that explained below and the idea to choose extreme two diagonal chunks:

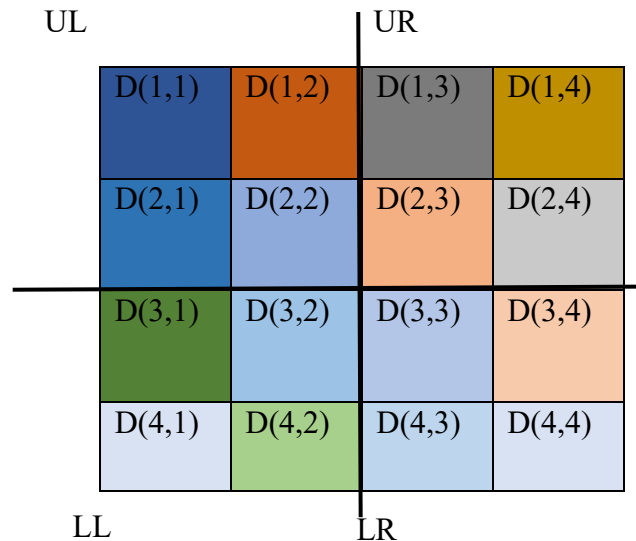


Figure 6 Division matrix into 4 coordinates UL, UR, LL and LR

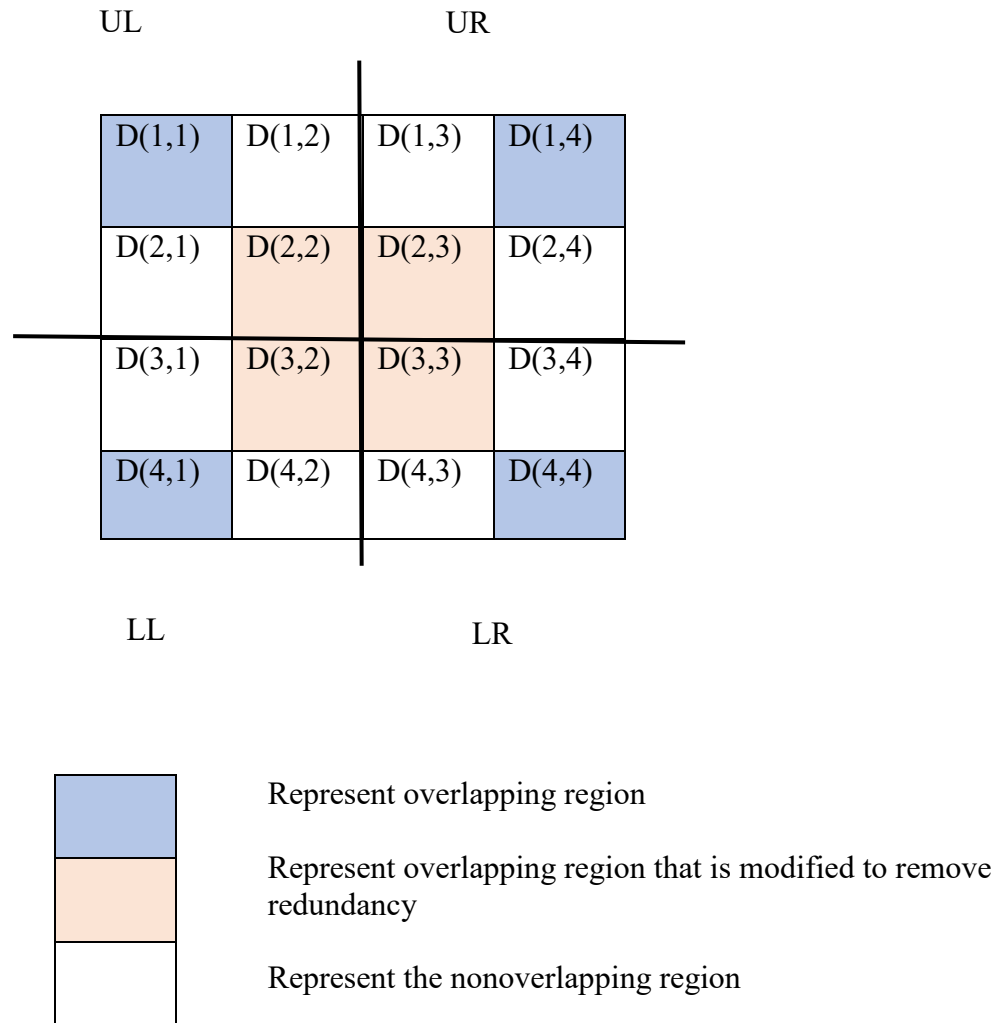


Figure 7. representation of redundant XOR result region

5.3.3.1 Case 1: Upper left coordinates: The UL contain elements D(1,1), D(1,2), D(2,1) and D(3,2) chose the extreme right diagonal elements, but there are collision in the chosen elements of D(1,1) and D(2,2) this is given by XOR operation between D(3,3) and D(4,4) given in figure 8. a) and 8. b). For removing the redundancy for D(2,2) chose the diagonal element immediate upper left and the lower right position shown in figure 8. d). for D(1,2) and D(2,1) the distinct XOR elements are possible without any redundancy. For

D(1,2), possible positions are D(2,3) and D(2,4) that are nonredundant and given in figure 8. e). For D(2,1), possible positions are D(3,1) and D(4,1) that are nonredundant and given in figure 8. f).

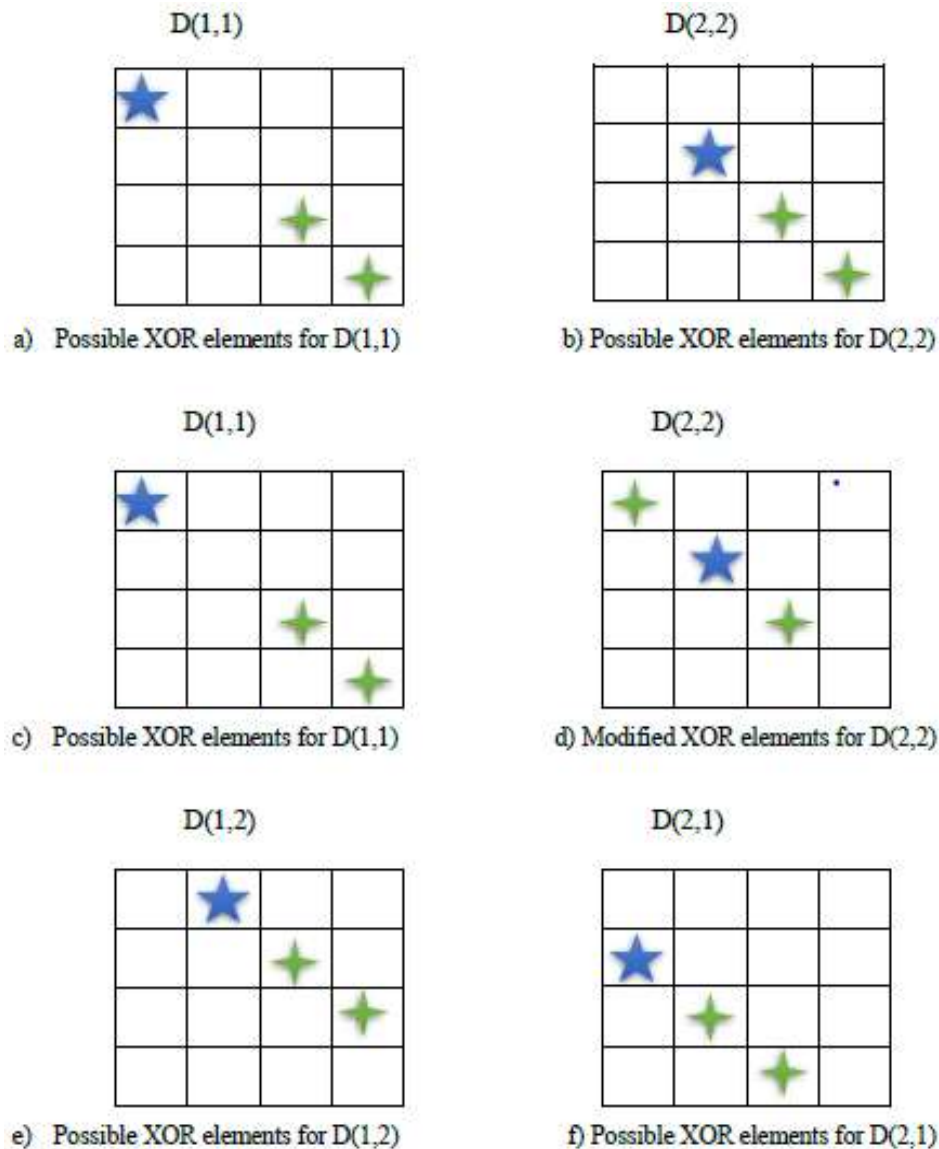


Figure 8. case1 for upper left UL coordinates.

5.3.3.2 Case 2: Upper right coordinates: The UL contains elements D(1,3), D(1,4), D(2,3) and D(2,4) chose the extreme right diagonal elements, but there are collision in the selected features of D(1,4) and D(2,3) this is given by XOR

operation between $D(3,2)$ and $D(4,1)$ given in figure 9. a) and 9. b). For removing the redundancy for $D(2,3)$ chose the diagonal element immediate upper left and lower right position is given in figure 9. d). for $D(1,3)$ and $D(2,4)$, the distinct XOR elements are possible without any redundancy. For $D(1,3)$, possible positions are $D(2,2)$ and $D(3,1)$ that are nonredundant and given in figure 9. e). For $D(2,4)$, possible places are $D(3,3)$ and $D(4,2)$ that are nonredundant and given in figure 9. f).

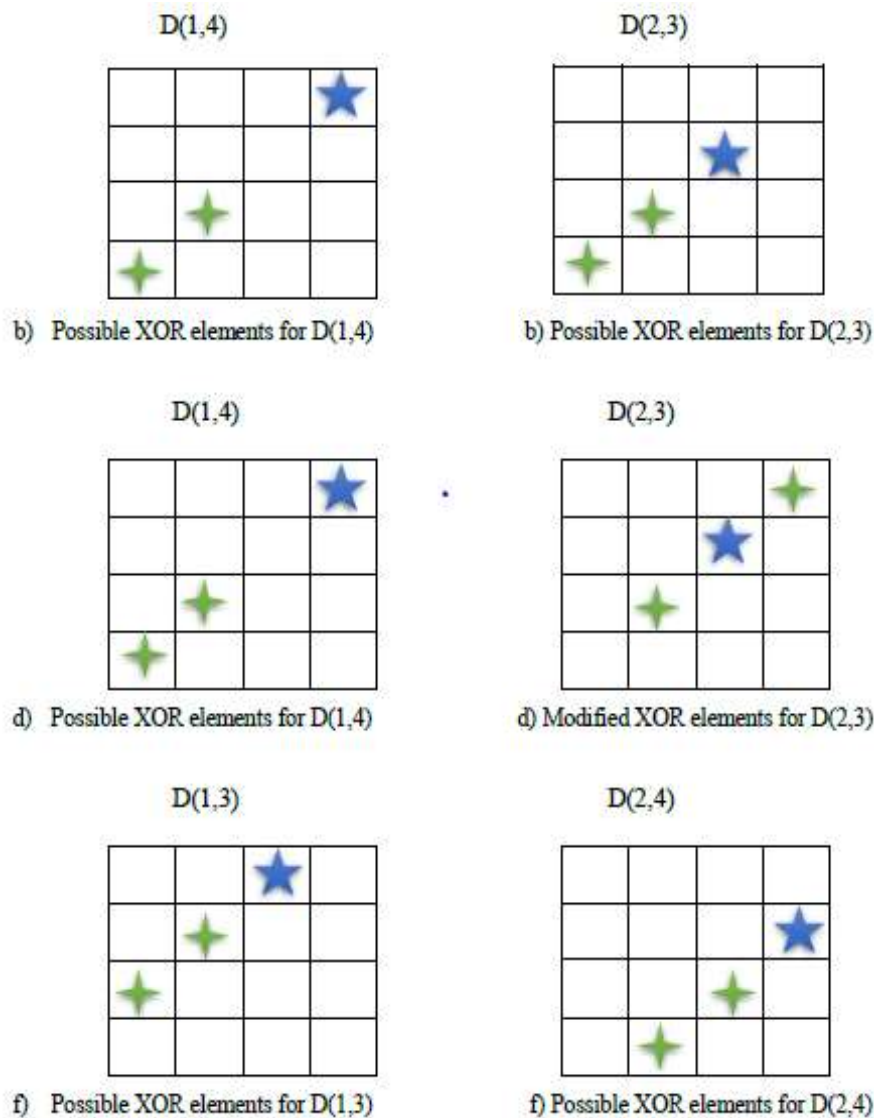


Figure 9 case2 for upper right UR coordinates.

5.3.3.3 *Case 3: Lower left coordinates:* The UL contains elements $D(3,1)$, $D(3,2)$, $D(4,1)$ and $D(4,2)$ chose the extreme right diagonal elements, but there are collision in the selected parts of $D(4,1)$ and $D(3,2)$ this is given by XOR operation between $D(1,4)$ and $D(2,3)$ given in figure 10. a) and 10. b). For removing the redundancy for $D(3,2)$ chose the diagonal element immediate upper left and lower right position is given in figure 10. d). for $D(3,1)$ and $D(4,2)$, the distinct XOR elements are possible without any redundancy. For $D(3,1)$, possible positions are $D(2,2)$ and $D(1,3)$ that are nonredundant and given in figure 10. e). For $D(4,2)$, possible positions are $D(3,3)$ and $D(2,4)$ that are nonredundant and given in figure 10. f).

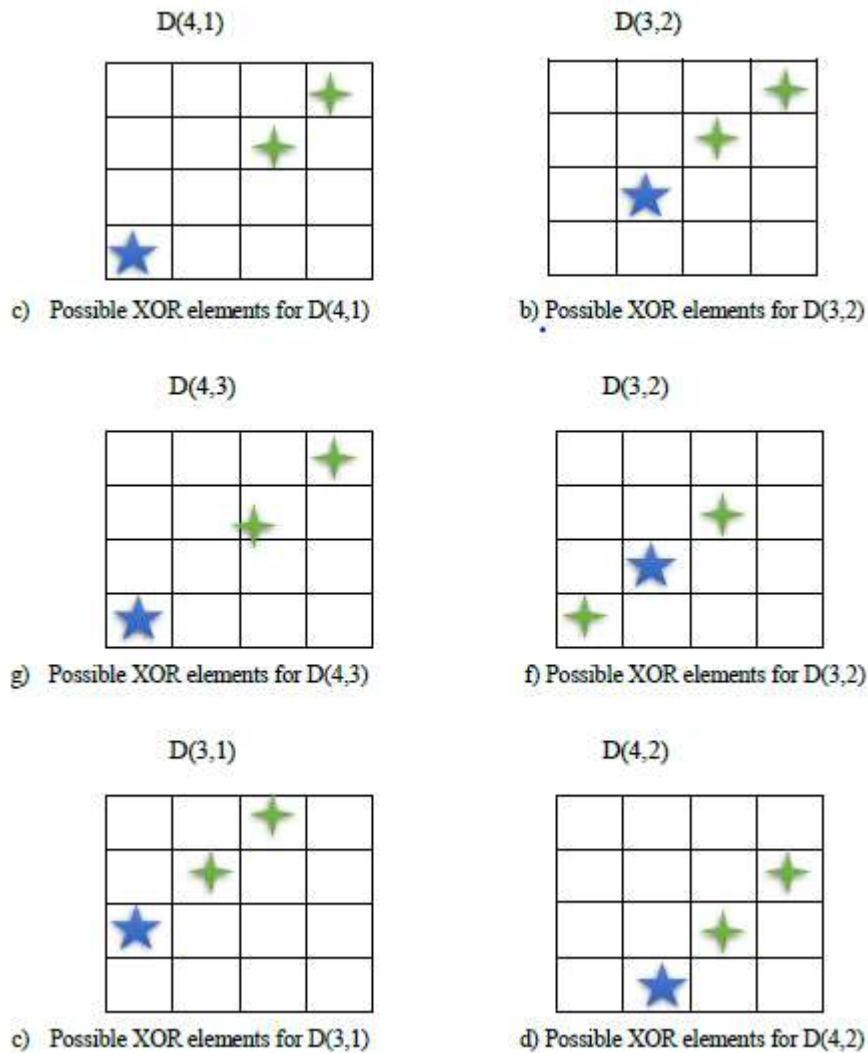


Figure 10 case 3 for lower left LL coordinates.

5.3.3.4 **Case 4: Lower right coordinates** : The UL contains elements $D(3,3)$, $D(3,4)$, $D(4,3)$ and $D(4,4)$ chose the extreme right diagonal elements, but there are collision in the selected elements of $D(3,3)$ and $D(4,4)$ this is given by XOR operation between $D(2,2)$ and $D(1,1)$ given in figure a) and b). For removing the redundancy for $D(3,3)$ chose the diagonal element immediate upper left and lower right position gives in figured). For $D(3,4)$ and $D(4,3)$ the distinct XOR elements are possible without any redundancy. For $D(3,4)$, possible positions are $D(2,3)$ and $D(1,2)$ that are nonredundant and given is figure e). For $D(4,3)$, possible positions are $D(3,2)$ and $D(1,1)$ that are nonredundant and given is figure f).

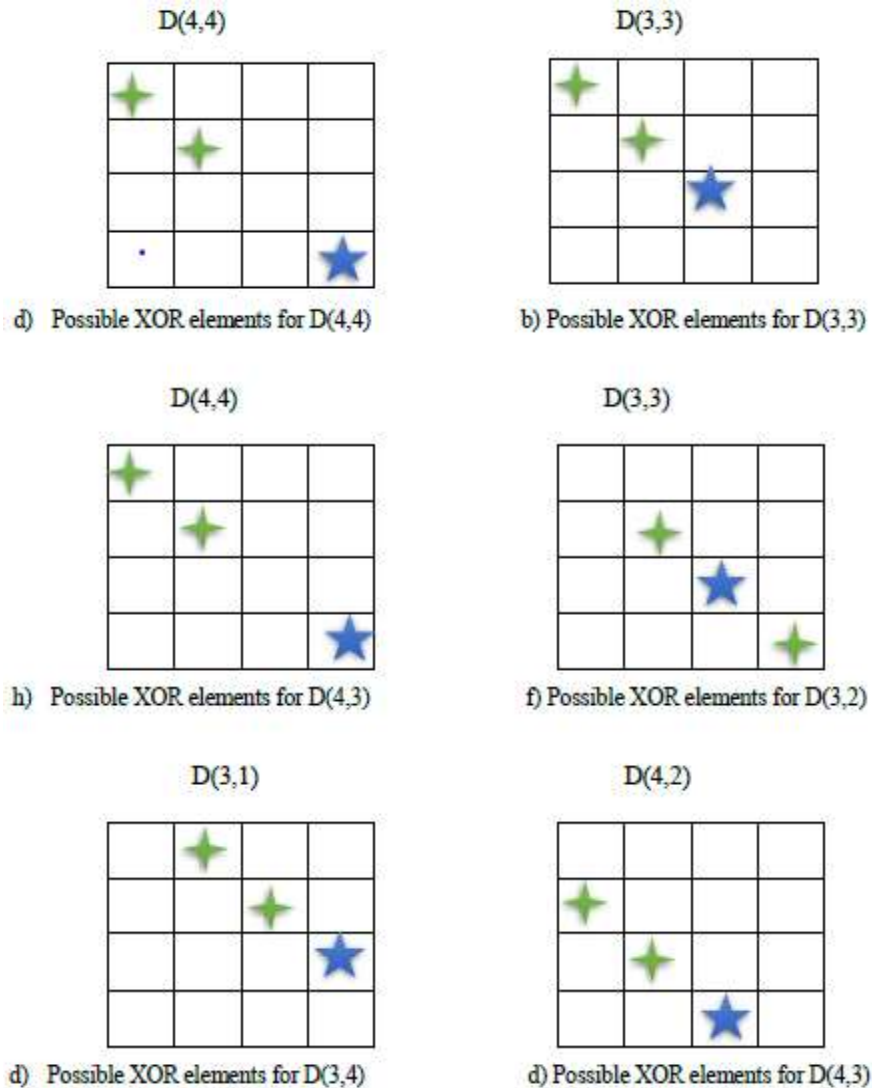


Figure 11 case4 for lower right LR coordinates

STEP 3: Encryption

Input: D [DNA sequence], K1[round Key].

Algorithm Body:

1. XOR with Key of the current state
2. nucleotides Substitution (DNA S-Box)
3. Shuffling nucleotides using the proposed D-fusion Algorithm.
4. Repeat step 1-3 for to generate 9 Rounds and for 10th round ignore step 3 this produces 64 cipher nucleotides.

Output: A [DNA generated key]

End Algorithm.

5.4 DNA Steganography Data post-processing (optional step)

In this step we again hiding cipher data into the FASTA file downloaded, this provides DNA steganography for hiding the sequences into another file. Now, this can be implemented using many approaches. Here, the implementation used is to record the position of quadruple that is PoQ. The PoQ proposed in order to record the position of a quadruple in FASTA file sequence. The FASTA file is a large file that usually contains all possible sequences that made from the nucleotides. For 4 bases that A, G, C, T 4⁴ possible cases are 256. In the FASTA file, these 256 cases can be present. Now, the idea is that divided the ciphertext into chunks of 4 nucleotides then search each chunk in the FASTA file individually and return its position given in the algorithm below. The space complexity is O(n) where n is the number of Nucleotides chunks. Here the fast file is an Array, and a chunk is a subarray of FASTA file.

$$D_{block} \subset F_{seq} \quad (5.2)$$

Problem:

1. this process has a high time complexity that is an NP-complete problem.
2. The possibility that the file does not contain all sequences.
3. Suitable for the small size of the document, as the position list may increase.

STEP 3: Steganography

ALGORITHM hiding ciphertext in the original FASTA file.

[The algorithm uses a search technique in order to locate and return the position of quadruple DNA nucleotides sequence in a FASTA file.]

Cipher nucleotide size = k nucleotide

Block size=4 nucleotide

Size of P is $k/4$

Input: D [DNA sequence], F [Random FASTA file of DNA sequences].

Algorithm Body:

1. NCBI provides a sample dataset of DNA strings, load the DNA string from the NCBI database of nucleotides.
2. Divide the cipher nucleotides into groups of 4 nucleotides then starting to find positions of each block.
3. The sequential search performed from the starting position.
4. P, a pointer to that store's position of the found quadruple DNA nucleotides sequence If the correct pattern found, its location then recorded in a pointer P.
5. Repeat the same procedure, starting from step 3, for all other characters;
6. **Output:** P, a pointer to the location of the found quadruple DNA nucleotides sequence and stored in a buffer.

End Algorithm.

5.5 DNA decryption

This process is the retrieving original Document from the encrypted format by providing Key. The process uses the same key for decryption also; hence, it is symmetric-key cryptography. The process is inverse of the encryption stages that decrypted on who has KEY for the cipher document. The receiver does not require the biological knowledge to decrypt the document all is required is the DNA cipher sequences and the KEY. The decryption process uses similar functions in reverse order, functions are reverse *R_AddDNAKey()Transformation*, *R_DNAsubstitute() Transformation* and *R_Diffusion()transformation*.

5.5.1 ***R_Diffusion()* transformation:** The reverse diffusion is again based on the 16 conditions to be replaced by the XOR of two other cells that are given above in figure Here the matrix of 4x4 is taken and divided into coordinates L for left and R for right. The Left is divided into two parts the upper left UL and lower left LL similarly; the right divided into two parts the upper right and lower right. Now the four conditions are such that explained previously is used to decrypt and the idea to choose extreme two diagonal chunks. The result of the diffusion
A function passed to the inverse substitution function.

5.5.2 ***R_DNA substitute()*Transformation:**The output from the *R_Diffusion()* transformation is used as input for this function. This process is implemented using reverse DNA s-box that is proposed in by Sabry et al. [1]. A reverse substitution method based on mapping using a table where the 16x16 table of reverse DNA substitution inspired by inverse AES s-box given in table 5.

5.5.3 ***R_AddDNAKey()*Transformation:** This function is similar to *AddDNAroundkey()* of encryption process round where the 64 size nucleotide is generated using *the Key_generation* process for 10 rounds and stored in a 4x4 matrix in a block of quadruples. Now the method is to add the key for the round generated before in key generation process step 2. In the adding round key, there is nucleotide wise XOR between the KEY for the round and chosen DNA cipher document. This process is used to provide the data to the next function, so the incremental movement of ciphertext to decipher text in an algorithm. This method uses the DNA XOR inverse property, which shown below such that the cipher of DNA block with key produce the original form.

$$DNA_{block_i} \oplus KEY = DNA'_{block_i}$$

$$DNA'_{block_i} \oplus KEY = DNA_{block_i}$$

STEP 5: Decryption

Input: P [position Key](optional), KEY [round Key], M [Random FASTA file of DNA sequences]/D[cipher nucleotide].

Algorithm Body:

1. If applied postprocessing then obtained cipher DNA sequences from position key P restore quadruples of nucleotides in an in ciphertext C.
2. Inverse Substitution (DNA S-Box)
3. Shuffling nucleotides using the proposed inverse D-fusion Algorithm.
4. XOR with Key of the current state
5. Repeat step 2-4 for to generate 10 Round and will get 128 cipher nucleotides.

Output: A [plain text]

End Algorithm.

CHAPTER 6: SECURITY ANALYSIS

In this section, Give C a chance to be such a foe, who has full power over the correspondence channel with the end goal that he/she can adjust, replay, catches the messages transmitted between A and B for security analysis. Here a comparative analysis is required in terms of security parameters. In the ADSE work based on Crucial symmetric block cipher, the input is 128 bits plain block and a 64 long sequence of nucleotides key extracted from randomly generated NCBI datasets and the output is encrypted cipher data the is present in the form of long DNA sequence that contains only A, G, C, T in its data set.

6.1 **The cracking probability:** in the ADSE work, the guessing probability is coming from different stages of the encoding steps. The probability of guessing at each level given and the levels are defined in figure 12 and describe below:

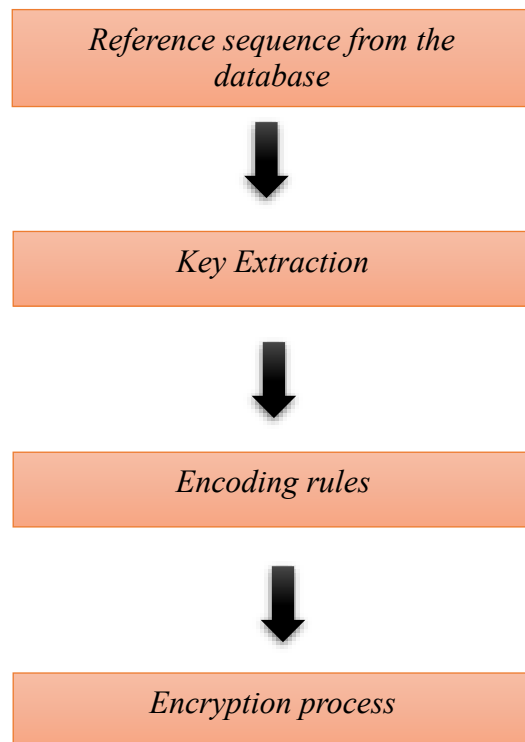


Figure 12. The stages of adding guessing probability

6.1.1 **Reference sequence from the database:** The reference sequences of nucleotides downloaded from the publicly available database. From the downloaded file, the length of 64 DNA nucleotides selected from a random position. The probability that an intruder successful obtained the correct data set given in equation 2.1.

6.1.2 **Key Extraction:** The probability of finding the starting position of reference sequence after extraction of the data set is high. On an average number of DNA nucleotides in a FASTA set is $3 * 10^6$, the let the size of a file is s then the probability to find starting position is:

$$p(starting) = \frac{1}{s - 63}$$

6.1.3 **Encoding rule:** After clearing two stages now the algorithm uses HDNA table to encode that where represent every two nucleotides as a combination of 8 bits that is 1 byte, there is 4^2 possible combination of DNA nucleotides. The cracking probability of the encoding given according to HDNA table rule:

$$p(e) = \frac{1}{16!}$$

6.1.4 **Apply encryption:** based on block cipher that uses 10 rounds for encryption. Also, the key size is 128 bits long key that is 64 nucleotides used and improves one level high security with high cracking probability and also improves efficiency, the cracking probability of the given key is:

$$p(k) = \frac{1}{2^{128}}$$

Therefore, the probability of guessing successful S is given by:

$$p(S) = p(DNA_{ref}) * p(starting) * p(e) * p(k)$$

$$p(S) = \frac{1}{1.63 * 10^8 * (s - 63) * 16! * 2^{128}}$$

On calculating the approximate results, the approximate value of 2^{128} is $3.40 * 10^{38}$ in decimal, and the $16!$ is approximately $2 * 10^{13}$ neglecting the other value, the approximate cracking probability for a key is:

$$p(S) = \frac{1}{10^{65}}$$

6.2 Security against the following possible attacks for symmetric encryption:

- 6.2.1 **Known plain content attacks:** As this attack explained in section 2.7.1, here the Shannon confusion and diffusion property are maintained by using 16×16 substitution DNA matrix and the diffusion using proposed D-fusion matrix.
- 6.2.2 **Chosen cipher attacks** as this attack explained previously in section 2.7.2, in the ADSE method random key generator system used that generates key randomly every time from NCBI data set of nucleotides. The distinct key for each document encryption resists this attack.
- 6.2.3 **Brute force:** as this attack explained previously in section 2.7.2, the key size is enormous, having very high cracking probability, not able to decrypt the Steganography without the key.
- 6.2.4 **Differential Cryptanalysis Attack:** this attack is similar to chosen plain text attack where the intruder has the set of the form where the plain text can be extracted, again every time a new key generated so the possibility to find the plain text is very low.

CHAPTER 7: RESULTS AND PERFORMANCE ANALYSIS

In this chapter, the theoretical comparison of AES cryptography algorithm and the ADSE work presented that is the time required for encryption and decryption process using XOR, AND, OR gates and shift operation. The comparison table with previous related work of the recent year, the experimental results and performance are given in table 6, 7, and 8.

7.1 Time and complexity analysis

Give C a chance to be such a foe, who have full power over the correspondence channel with the end goal that he/she can adjust, replay, catches the messages transmitted between A and B. In AES the mixing column steps where the multiplication of a 4x4 block with a fixed constant matrix of size 4x4. Here the multiplication is not the standard matrix multiplication; multiplication based-on GF (2^8) arithmetic. GF stands for Galois field Arithmetic where 2^8 represents eight binary bits or a byte that from 0 to 255. The representation is in the hexadecimal format a byte represents using 2 Hexadecimal number, and written in polynomial form.

Example: representation on polynomial form:

Input: 10011100

Output: $x^7 + x^4 + x^3 + x^2$

7.1.1 AES cryptography

Computation Evaluation comparison of the Proposed ADSE with AES cryptography and the performance mathematical computation of block cipher, given in [9-10], that the most general logical operations used in block symmetric key cipher are bitwise OR, bitwise AND, byte shifting and shuffling. From the logical gates, the two AND gate and an OR used to give an XOR gate, from this let us consider $XOR = 2AND + OR + 2NOT$. Also, the shift operation of 8 bits k position given by bitwise OR operation.

Here N_b is the block length/32.

The AES rounds are based upon the size of key, where each round is composed of 4 operations 1. Byte substitution, 2. Row shifting, 3. Mixing column, 4. Key adding round. In [10-11] given encryption time

TAES – ENCRYPT

$$= (46N_bN_r - 30N_b) T_a + [31N_bN_r + 12(N_r - 1) - 20N_b] T_o \\ + [64N_bN_r + 96(N_r - 1) - 61N_b] T_s$$

The Decryption process AES-Rijndael is the inverse of the encryption system that makes use of inverse code. For mixing column, it uses another polynomial this is $0Bx^3 + 0Dx^2 + 09x + 0E$, this alter the complexity of AES encryption time. The obstacle is inside the AES-Rijndael this growth the complexity. Consequently, the decryption method is costlier than the encryption method. The time required for the Decryption procedure in AES-Rijndael given underneath that is the addition of $96N_b$ bitwise ANDs, $72N_b$ bitwise ORs and $32N_b$ bitwise shifts of bytes of 1 block of round and apply for each besides the initial round the do not encompass Mixcolumn operation given through:

$$TAES - DECRYPT = TAES - ENCRYPT + [96N_bT_a + 72N_bT_o - 32N_bT_s] \times (N_r - 1)$$

7.1.2 Proposed Encryption

In the ADSE data is present in DNA format, the operation corresponding to the nucleotides. The rounds based on the key size, and in every round, there are three operations performed 1. Adding round key, 2. DNA s-box substitution, 3. Apply fusion conditions.

7.1.2.1 **AddDNAKey() Transformation:** In this round, the key for the corresponding round is XOR with the corresponding state matrix. The $D_{N_b} = \text{blocksize}/16$, where the block size is 64, and the D_{N_b} is 4, for 4 rows. For each cell 1 XOR function is used, that is 4 XORs and implemented using $4D_{N_b}$ XORs that is $8D_{N_b}$ ANDs and $4D_{N_b}$ ORs.

7.1.2.2 ***DNAsubstitute() Transformation:***The S-box in DNA is a function from $GF(2^8)$ which represent as $S(x) = x^2^8 - 2$. The non-linearity of produce by DNA S-box individually on each cell, the mapping. A 16 x 16 DNA S-box table used, the input is 4 sequential nucleotides to and mapping is used to give 4 nucleotides output. This process improves the complexity required for evaluation but required memory to store matrix, but overall improves efficiency and implements using $3D_{N_b}$ ANDs and $2D_{N_b}$ ORs.

7.1.2.3 ***Diffusion() transformation:*** The system is based upon the 16 conditions to give a brilliant shuffling, for every mobile required a shift and XOR operation. For every cell pick two distinct positions cells such that the result of XOR is specific for every position. This required $4D_{N_b}$ XORs to get the replaceable nucleotides and $4D_{N_b}$ shift operation to shift. That is implemented using $8D_{N_b}$ bitwise ANDs and $4D_{N_b}$ bitwise ORs and $4D_{N_b}$ shift operation to shift.

$$T_{DNA}ENCRYPT = (16D_{N_b}D_{N_R} - 8D_{N_b})T_a + (8D_{N_b}D_{N_R} - 4D_{N_b})T_o + (4D_{N_b}D_{N_R} - 2D_{N_b})T_s$$

Decryption

In the ADSE work data is present in DNA format, the operation corresponding to the nucleotides. In decryption algorithm, rounds based on the key size, and in every round, there are three reverse operations are performed to recover the original document 1. Reverse Adding round key, 2. DNA inverse s-box substitution, 3. Apply diffusion conditions.

7.1.2.4 ***R_AddDNAKey() Transformation:***In this round, the key for the corresponding round is XOR with the corresponding state matrix. The $D_{N_b} = \text{blocksize}/16$, where the block size is 64, and the D_{N_b} is 4, for four rows. For

each cell 1 XOR function is used, that is 4 XORs and implemented using $4D_{N_b}$ XORs that is $8D_{N_b}$ ANDs and $4D_{N_b}$ ORs.

7.1.2.5 *R_DNAsubstitute()* Transformation: The S-box in DNA is a function from $GF(2^8)$ which represent as $S(x) = x^2 - 2$. The non-linearity of produce by DNA S-box individually on each cell, used mapping. A 16 x 16 DNA S-box table used, the input is four sequential nucleotides to, and mapping is used to give four nucleotides output. This process improves the complexity required for evaluation but required memory to store matrix, but overall improves efficiency and implements using $3D_{N_b}$ ANDs and $2D_{N_b}$ ORs.

7.1.2.6 *R_Diffusion()* transformation: The system is based upon the 16 conditions to give a brilliant shuffling, for every mobile required a shift and XOR operation. For every cell pick two distinct positions cells such that the result of XOR is specific for every position. This required $4D_{N_b}$ XORs to get the replaceable nucleotides and $4D_{N_b}$ shift operation to shift. That is implemented using $8D_{N_b}$ bitwise ANDs and $4D_{N_b}$ bitwise ORs and $4D_{N_b}$ shift operation to shift.

$$T_{DNA}DECRYPT = (16D_{N_b}D_{N_R} - 8D_{N_b})T_a + (8D_{N_b}D_{N_R} - 4D_{N_b})T_o + (4D_{N_b}D_{N_R} - 2D_{N_b})T_s$$

The encryption and Decryption time complexity of the proposed work compared with AES, where AES is required more mathematical computation than the proposed work, so the encryption and decryption time greater than the proposed work. As the ADSE applied the biological complexity so this justifies its lesser mathematical computation compared to the original AES cryptography algorithm. The use of D-fusion matrix highly reduced Shift operation and the comparison results and analysis are shown in table 6 and 7.

Where:

N_b Number of blocks

N_r Number of rounds

D_{N_b} Number of DNA blocks

D_{N_r} Number of DNA rounds

ENCRYPTION						
Scheme	Number of AND operation		Number of OR operation		Number of Shift operation	
AES-Rijndael computational analysis [9]	$46N_bN_r - 30N_b$	1810	$31N_bN_r + 12(N_r - 1) - 20N_b$	1388	$64N_bN_r + 96(N_r - 1) - 61N_b$	2060
ADSE	$16D_{N_b}D_{N_r} - 8D_{N_b}$	632	$8D_{N_b}D_{N_r} - 4D_{N_b}$	304	$4D_{N_b}D_{N_r} - 2D_{N_b}$	152

Table 6 AES-Rijndael computational analysis and ADSE encryption comparison

DECRYPTION						
Scheme	Number of AND operation		Number of OR operation		Number of Shift operation	
AES-Rijndael computational analysis [9]	$46N_bN_r - 30N_b - 96N_b$	1426	$31N_bN_r + 12(N_r - 1) - 20N_b - 72N_b$	1100	$64N_bN_r + 96(N_r - 1) - 61N_b + 32N_b$	2188
ADSE	$16D_{N_b}D_{N_r} - 8D_{N_b}$	632	$8D_{N_b}D_{N_r} - 4D_{N_b}$	304	$4D_{N_b}D_{N_r} - 2D_{N_b}$	152

Table 7 AES-Rijndael computational analysis and ADSE work decryption comparison

7.2 Experimental results

7.2.1 The execution time, average execution time and changing probability of the ADSE scheme

The outcomes are the execution time of the distinctive class of audio documents, image and text additionally textual content files with .pdf, .txt and .doc extension. The document size given a wide variety of bits in a file, the variety of blocks is after mapping the bits to DNA nucleotides than the collection split into a group of 64 nucleotides. So, the size of one block is 64 nucleotides if the last block is not whole then add padding series. In table

8, the calculation of the execution time of documents, the average execution time per block and the changing probability shown, and decryption shown below.

7.2.1.1 **The execution time:** is given in seconds, is the time required using a document to transform from original to cipher document.

7.2.1.2 **Average Execution time:** The average number of execution time calculated for each sort of record in milliseconds given below in equation 7.1. After calculation, the Average execution time of each file lies between 0.13 to 0.18 milliseconds. Where the document file execution time faster and pdf takes more time than others. The implementation of the project is using BASH scripting and python3.

$$Avg_{ET} = \frac{ET}{Nb} \quad (6.1)$$

ET: execution time

Nb: is the number of blocks

7.2.1.3 **The changing probability:** is a change in the number of nucleotides; it is the DNA nucleotides from the time it converted into DNA nucleotides by applying HNDNA mapping to its conversion into DNA nucleotides. The probability is used to determine the number of DNA nucleotides is changed, the individual calculation of the bases and group by N(A), N(G), N(C) and N(T) as a number of bases in beginning of encoding process and by N'(A), N'(G), N'(C) and N'(T) the number of bases after the encryption process. The probability calculation formula given below shown by the subtraction of N(base) with N'(base) and the result is divide with the total number of bases that N(A, G, C, T). The changing probability is best when the result is closer to 0.5 and worst when closer to 0 and 1. If $P_{changing}$ closer to 0 then the substitution is negligible if $P_{changing}$ closer to 1 then the substitution is inverse of the nucleotides after inversion that is nearly equal to negligible substitution. The $P_{changing}$ is near to

0.5, that means the nucleotides is a high standard substitution. The tested and encrypted documents for changing probability are near to 0.5 is shown in table 8.

$$P_{changing} = \frac{|N(A) - N'(A)| + |N(G) - N'(G)| + |N(C) - N'(C)| + |N(T) - N'(T)|}{N(A) + N(G) + N(C) + N(T)}$$

File type	document size	Number of Blocks (Nb)	Execution time(seconds)	Average (Et/Nb) ms	Changing probability
.pdf	3287740	205484	36.11	0.17573	0.4885162
.txt	48518	3033	0.46	0.15166	0.3962524
.mp3	715546	44722	8.49	0.18983	0.4453574
.doc	298470	18655	2.65	0.14205	0.4943436
.jpg	26397	1650	0.27	0.16363	0.4209001

Table 8 The execution time, average execution time and changing probability.

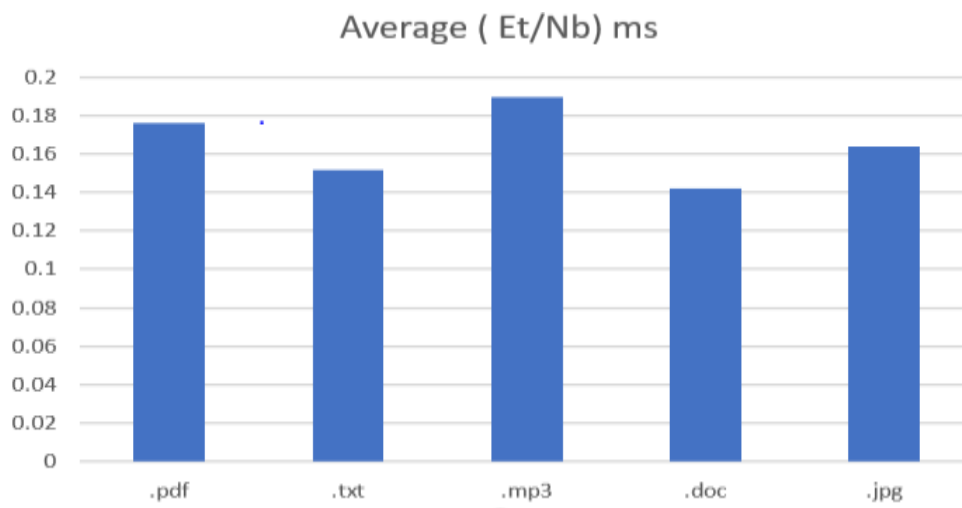


Figure 13. Average encryption time of document files

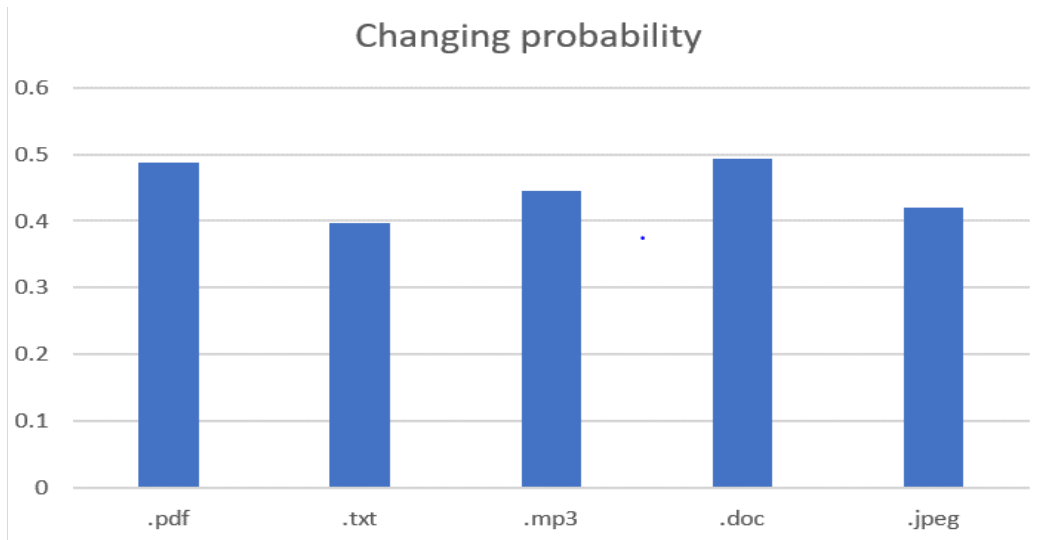


Figure 14 Changing the probability of document files.

7.2.2 The comparison of the execution of time of the paper [2] and the ADSE

The average execution time calculated by dividing the execution time by block size and the execution of the proposed work is better than the silent mutation using AES. In paper silent mutation the block size is 128 nucleotides, so the Average ET is further divided by 2. as in the proposed work the block size is 64 nucleotides.

Type of data	Average ET (Et/Nb) ms (silent mutation)	Average ET (Et/Nb) ms (proposed)
text	2.37	0.15166
image	3.01	0.16363
audio	3.48	0.18983

Table 9 comparison table of average encryption time of silent mutation and ADSE.

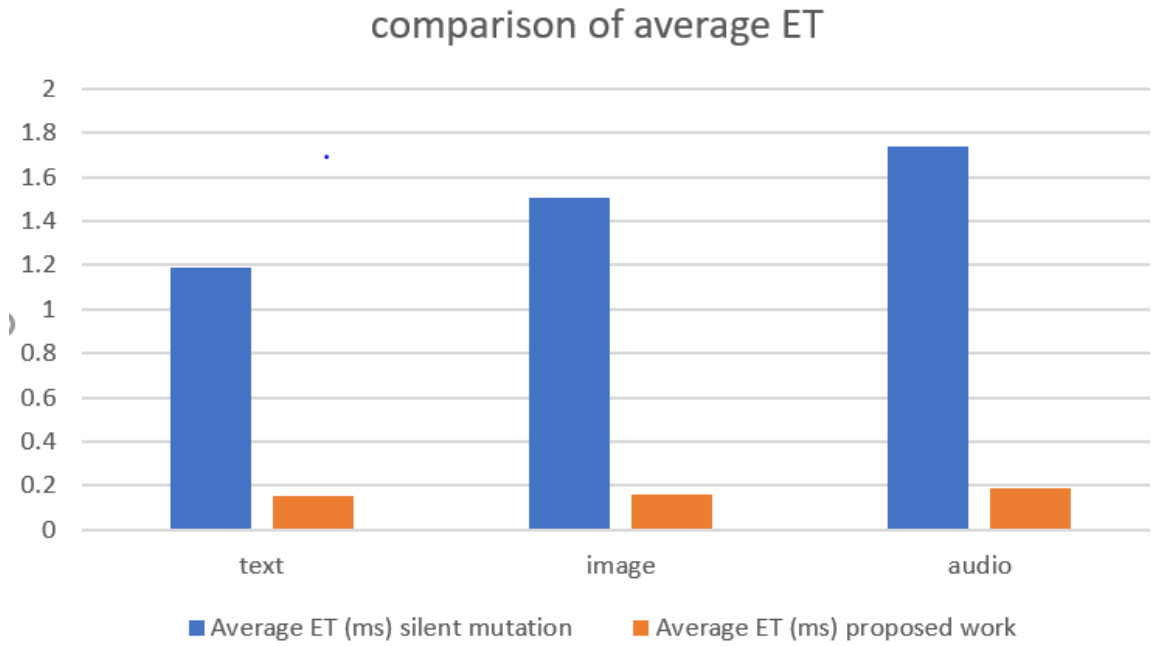


Figure 15 Graph of the comparison table

CHAPTER 7: CONCLUSION AND FUTURE WORK

DNA cryptography is a newly emerging approach, which requires more exertion in this area as DNA cryptography is not standardised and notably flexible. It uses belongings of DNA that hides lots of information in 4 nucleotides A, G, C and T bases. The ADSE presents the double layer of safety by making use of symmetric key cryptography and constructs use of DNA steganography hiding cipher information in FASTA document. The proposed work required minor amendment, and it holds the biological encryption. On the contrast, the proposed work has higher encryption time and higher guessing possibility. The work at ease shapes regarded plain text attack, selected cipher textual content attack and brute force attack. It has better potential as every nucleotide conceals 4 bits on which the ratio is 1:4 that is 4 bits are hiding per nucleotide.

The future work extends the algorithm to increment the key size as it is using only 64 nucleotides which can be extended to 128, 198 nucleotides and more. The machine learning and neural network concepts can be utilised to generate key, to improve the methodology in case of steganography as the existing process is more complex to decrypt.

REFERENCES

1. Sabry et al. (2015). Design of DNA-based Advanced Encryption Standard (AES). 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS).
2. Bahig et al. (2018). DNA-Based AES with Silent Mutations. *Arabian Journal for Science and Engineering*.
3. Zhang et al. (2014). On the security of symmetric ciphers based on DNA coding. *Information Sciences*, 289, 254–261.
4. Basu et al. (2019). Bio-Inspired Cryptosystem with DNA Cryptography and Neural Networks. *Journal of Systems Architecture*.
5. Zhang et al. (2014), “On the security of symmetric ciphers based on DNA coding. *Information Sciences*,” 289, 254–261.
6. Hamed et al. (2015)” Hybrid technique for steganography-based on DNA with n-bits binary coding rule” 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR).
7. Marwan et al. (2016). DNA-based cryptographic methods for data hiding in DNA media. *Biosystems*, 150, 110-118.
8. Malathi et al. (2017). Highly Improved DNA Based Steganography. *Procedia Computer Science*, 115, 651-659.
9. Doomun et al. (2009). Analytical Comparison of Cryptographic Techniques for Resource-constrained Wireless Security. *IJ Network Security*, 9(1), 82-94.
10. Doomun et al. (2008). AES-CBC software execution optimization. 2008 International Symposium on Information Technology.
11. Marwan et al. (2016). DNA-based cryptographic methods for data hiding in DNA media. *Biosystems*, 150, 110-118.
12. Clelland et al. (1999). Hiding messages in DNA microdots. *Nature*, 399(6736), 533.
13. Wang et al. (2006). A DNA procedure for solving the shortest path problem. *Applied mathematics and computation*, 183(1), 79-84.
14. Goldman et al. (2013). Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature*, 494(7435), 77.

15. Sohal et al. (2018). BDNA-A DNA inspired symmetric key cryptographic technique to secure cloud computing. *Journal of King Saud University-Computer and Information Sciences*.
16. Cui et al. (2008, September). An encryption scheme using DNA technology. In *2008 3rd International Conference on Bio-Inspired Computing: Theories and Applications* (pp. 37-42). IEEE.