Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 463 – 475

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

A Novel DNA Computing based Encryption and Decryption Algorithm

Noorul Hussain UbaidurRahman[a,*],Chithralekha Balamurugan[b],Rajapandian Mariappan[c]

[a]*Dept. of Banking Technology, Pondicherry University, Puducherry – 605014, India*
[b]*Dept. of Computer Science, Pondicherry University, Puducherry – 605014, India*
[c]*Dept. of Mathematics & Computer Science, KMCPGS, Puducherry – 605008, India*

Abstract

Lot of techniques and systems has been developed based on modular arithmetic cryptography for encryption and decryption. However, these techniques are broken using DNA cryptography techniques and methods. DNA Cryptography is a new instinctive cryptographic field that has emerged from the research of DNA computing. Some algorithms that are available in DNA Cryptography have limitations in that they still use modular arithmetic cryptography at some of their steps or they are biological laboratory experiment based which is not suitable in the digital computing environment. To overcome this lacuna, we describe a novel, secure, unique and dynamic DNA based encryption and decryption algorithm and also provide an analysis of its performance.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Cryptography; DNA Computing; DNA Cryptography; Encryption; Decryption

1. Introduction

Secure communication is vital to facilitate confidential exchange of information between any sender and receiver. Nowadays, the internet has become the media for all banking and electronic commerce transactions and it is very

* Corresponding author. Tel.: +91-9944981398
E-mail address: unoorulhussain@gmail.com

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.045

essential that the communication is made in a highly secure manner. In order to cater to these security requirements, a lot of techniques and systems have been developed in the mathematical cryptography for encoding and decoding the plain text. However, these techniques are overcome using DNA cryptography techniques and methods. The DNA cryptography is an emerging field in the area of DNA computing research. DNA cryptography plays major a role in next generation security. Some algorithms that are available in DNA Cryptography have limitations in that they still use modular arithmetic cryptography at some of their steps or they are biological laboratory experiment based which is not suitable in the digital computing environment. To overcome this lacuna, we describe a novel, secure, unique and dynamic DNA based encryption and decryption system and also provide an analysis of its performance. This DNA cryptography uses Central Dogma of Molecular Biology (CDMB) concept for encryption and decryption. The rest of this paper is organized as follows. Section 2 describes the requirements to be fulfilled by DNA encryption algorithm. Section 3 describes the proposed algorithm. In Section 4, the experimental results are presented. Section 5 describes fulfilment of the requirements. Section 6 gives the conclusion of this work.

2. Requirement to be fulfilled by DNA Encryption Algorithm

Every DNA Encryption algorithm should fulfill a set of requirements. These requirements have been identified in this paper based on the observed limitations in the existing encryption algorithms as listed in Table 1. The fulfillment of set of requirements in existing works in the DNA encryption algorithms are as shown in table 2.

Table.1. Definition of Requirements to be fulfilled by DNA Encryption Algorithm

Sl.No	Requirements	Definition
1	DNA Encoding of Complete character set fulfilment	The DNA encoding table should provide for DNA encoding sequences for the complete character set, which contains 96 elements.
2	Dynamic Encoding Table Generation	Encoding Table is generated randomly after every session interval providing different DNA sequence for every element of the character set
3	Unique sequence for encoding of every character of plaintext to DNA sequence	The encoding of plaintext into DNA sequence is unique for every element of the character set in every generation of encoding table in every session.
4	Robustness of encoding.	The encoding table should be based on highly randomized secure encoding table and also the generations should contains random steps.
5	Biological Process Simulation	The DNA encryption and decryption algorithm should be based on biological process which is simulated to adapt to digital computing environment.
6	Dynamicity of encryption process	The same plaintext can produce different cipher text for every session due to unique DNA encoding table generation for every session.

Table.2. Fulfilment of DNA Encryption algorithm requirement in existing works

Authors	DNA Encoding of Complete character set fulfilment	Dynamic Encoding Table Generation	Unique sequence for encoding of every character of plaintext to DNA sequence for every session	Robustness of encoding	Biological Process Simulation	Dynamicity of encryption process
Guangzhao Cui et.al [3]	×	×	×	×	*	*
Qiang Zhang et.al[10]	×	×	×	×	*	×
Souhila Sadeg et. al[11]	×	×	×	×	✓	×
Sherif T.Amin et. al[12]	×	×	×	×	✓	×
O.Tornea & M.E.Borda[8]	×	×	×	×	*	×
Zhang, Qiang et. al[16]	×	×	×	×	×	×
Kang Ning et. al[4]	×	×	×	×	✓	×
Mona Sabry et. al[6]	×	×	×	×	*	*

Padma Bh et. al[9]	×	×	×	×	×	×
Xing Wang & Qiang Zhang[14]	×	×	×	×	×	×
Akanksha Agarwal et.al[1]	✓	×	×	×	×	×

×- Indication of minimum level of supporting. ✓- Indication of Acceptable Level of Supporting. * - Partial fulfilment

The above table 2 illustrates that, the fulfilment of requirements of existing works with respect to DNA Cryptography is not complete. The following discussion elaborates the same.

1. *DNA Encoding of Complete character set fulfilment* – The DNA encoding of complete character set fulfilment should provide alphabets (uppercase, lowercase), numbers and special character. It is used to encode all the character set of the plaintext into DNA sequence. The ones which are available in the existing works are not complete and also do not help in enforcing security at the encoding phase itself. Akanksha Agrawal et.al proposes a mapping table which fulfills all the alphabet characters (uppercase, lowercase), Numbers, and special characters. But the mapping table is developed manually which is very easy to decode. In order to overcome this problem, we need to provide an encoding table which provides encoding sequences for the complete character set, which contains all the alphabets (uppercase & lowercase), numbers and special characters of 96 elements. Also, the generation of the encoding table should be based on a well-defined procedure which can be repeated any number of times to generate the encoding table instead of being a manual process.

2. *Dynamic Encoding Table Generation* **-** In order to ensure a higher level of security, the encoding table should be generated new at periodic intervals or for every interaction session between the sender and receiver. Also providing different DNA sequences for every element of the character set is important. This objective is not within the scope of any of the existing encoding table generation algorithms.

3. Unique sequence for encoding of every character of plaintext to DNA sequence – The encoding of plaintext into DNA sequence should be unique for every element of the character set in every generation of encoding table in every session between sender and receiver. This requirement is not within the scope of any of the existing works.

4. *Robustness of encoding*- In order to ensure attack resistance, the DNA encoding of the plaintext should provide a robust encoding scheme which is very difficult to decipher. In the existing works, the encoding is not robust due to manual formulation.

5. *Biological Process Simulation*- the DNA encryption and decryption algorithm should be based on the biological processes which are simulated to adapt to digital computing environment. In the existing works, some cryptographic algorithms are purely biology laboratory experiment based which is not suitable to application in the digital computing environment. In some other algorithms, a partial of the algorithm is based on a biological process simulation whereas a part of it is based on modern cryptography. Since modern cryptographic algorithms have been broken using DNA cryptography, a complete algorithm which is completely based on simulation of difficult biological processes is required.

6. *Dynamicity of encryption process*- The dynamicity of encryption process is required to ensure that the same plaintext can produce different cipher texts. In the existing algorithms, this is achieved by a combination of modern and DNA cryptography and not individually through DNA cryptography.

Since, all the above requirements have not been completely fulfilled in the existing works, a novel DNA based encryption and decryption algorithm is proposed and described below.

3. Proposed Algorithm

The proposed algorithm consists of two parts,

1. The DNA Computing based encoding algorithm.
2. The DNA Computing based encryption and decryption algorithm.

3.1. The DNA Computing based Encoding Algorithm

The DNA Encoding Algorithm has a well-defined process and explained in detail in[17]. For sake of brevity, the focus of this paper is retained on the encryption and decryption algorithm and not dealt in detail. A sample output of DNA Encoding algorithm will be as shown in table 3. However, as described in the requirements, the DNA encoding table is generated after every pre-defined session intervals and hence the DNA sequences and the assignment of alphabets to them would be different across different sessions.

Table.3. DNA Encoding Table

	C	**A**	**T**	**G**
A	ACAT- a	AAAA – y	ATAA– W	AGAG – {
	ACTG- b	AATT – z	ATTT – X	AGTA – [
	ACCC- c	AACC – A	ATCG – Y	AGCG - }
	ACGA – d	AAGG– B	ATGC – Z	AGGG -]
T	TCAT – e	TAAT – C	TTAA – 0	TGAA - |
	TCTG – f	TATG – D	TTTT – 1	TGTT - \
	TCCG – g	TACC – E	TTCC – 2	TGCG - +
	TCGT – h	TAGA – F	TTGG - 3	TGGC - =
C	CCAG – i	CAAT – G	CTAT – 4	CGAA - _
	CCTA – j	CATG – H	CTTG – 5	CGTT - -
	CCCG – k	CACG – I	CTCC – 6	CGCC -)
	CCGG – l	CAGT – J	CTGA – 7	CGGG – (
G	GCAA – m	GAAG– K	GTAT – 8	GGAT - *
	GCTT – n	GATA – L	GTTG – 9	GGTG - &
	GCCG – o	GACG– M	GTCG – <	GGCC - ^
	GCGC – p	GAGG– N	GTGT - >	GGGA - %
A	ACTC – q	AATA – O	ATTA - ,	AGTT - $
	ACCG – r	AACG – P	ATCC - .	AGCC - #
T	TCTC – s	TATC – Q	TTTA - ?	TGTA - @
	TCCC – t	TACG – R	TTCG - /	TGCC - !
C	CCTT – u	CATC – S	CTTC - :	CGTA - ~
	CCCC – v	CACC – T	CTCG - ;	CGCG - `
G	GCTA – w	GATT – U	GTTC – “	GGTC - €
	GCCC – x	GACC – V	GTCC – ‘	GGCG - £

3.2. The DNA Computing based encryption and decryption algorithm.

3.2.1. Encryption Process

The encryption algorithm comprises of the following steps for encrypting plaintext into Cipher Text: Before encryption starts, the encoding processes is carried out for plaintext to DNA sequence conversion.

Step 1: Sender generates a DNA encoding table1 for encoding of plain text into DNA sequence using DNA encoding algorithm and also receives a clue [17] from the receiver to generate encoding table 2 through the same encoding technique.

Step 2: The plaintext to be encoded is divided into two halves equally. If the plaintext is not even, one random element is appended in the last to make both even. One half of the plaintext is converted into DNA sequence using

DNA Encoding Table 1 available with the sender and the other half of the plaintext is converted into DNA sequence using DNA Encoding table 2 obtained from the receiver.

For example: Let us assume that the plaintext is "BANK"

The plaintext is divided into two halves equally and let us assumes that the DNA sequences obtained using encoding table 1 and 2 are as follows.

BA	NK
AAGG ACAT	GCTT GAAG

Step 3: The multiple round functions are applied on both left side and right side plain text and the steps involved in multiple round function is as follows. The minimum number of rounds should be greater than or equal to 10.

a) There is an intron sequences used in the generation of encoding tables 1& 2 generated by the sender and receiver respectively. These two intron sequences are taken for performing transformation operation with the DNA encoded plaintext sequence respectively for both left and right halves simultaneously. The transformation operation is actually an XNOR of the DNA encoded plaintext with the respective intron sequences. To Perform this XNOR operations DNA sequences is first converted to binary using the following mapping A-00, T-01, C-10, G-11, the XNOR is carried out on this Binary sequence. For e.g.:

CACA CCAG Transformation DNA CTGT TAAT

b) The transformed DNA Sequence is converted into mRNA sequence by replacing Thymine (T) with Uracil (U) on both right and left side DNA sequences. This process is a simulation of the biological transcription process.For e.g.:

CACA CCAG mRNA Sequence CUGU UAAC

c) The mRNA sequence is converted into tRNA sequence by replacing every DNA alphabet with its complement DNA alphabet. For e.g. A-U, U-A, G-C, C-G conversions are carried out. This process is a simulation of biological translation process. For e.g.:

GUGU GGUC tRNA sequence GACA AUUA

d) The tRNA sequence is converted into DNA sequence by replacing Uracil (U) with Thymine (T) in the tRNA sequences. This process is a simulation of biological reverse transcription. For e.g.:

GTGT GGTC Reverse Transcription GACA ATTA

e) The DNA sequence from reverse transcription is right shifted once on both sides. For e.g.:

CGTG TGGT Shift Sequence AGAC AATT

Step 4: In this step, the tRNA sequence obtained after the multiple rounds executed in step 4 is taken and converted to amino acid. For this conversion, every tRNA sequence requires a corresponding amino acid sequence. For this conversion, a suitable amino acid table is generated. The process for amino acid table generation is described below.

Amino Acid Table Generation:

Step1: Generate two DNA sequences randomly consisting of 4 DNA alphabets. The generated sequences should have all the four chemical compounds of DNA and the two sequences should not be identical. For e.g.:

Seq1. ATCG Seq2. GTAC

Step2: Converting the two sequences into mRNA. For e.g.:

Seq1. ATCG - Column Seq2. GTAC - Row

mRNA Conversion: AUCG GUAC

Step3: Assigning the two mRNA sequences randomly row wise and column wise in 4X4 matrix. The matrix elements are product of the corresponding row and column elements. Now, every matrix element should contain a 2 letter DNA alphabet sequence as shown table 4. For e.g.:

Table.4. Amino Acid Table

A	U	C	G
GA	GU	GC	GG
UA	UU	UC	UG
AA	AU	AC	AG
CA	CU	CC	CG

Step4: The 4X4 matrix is extended into 16X16 matrix by using matrix elements of the above 4X4. The extension is carried out as follows. The 16 matrix elements are assigned to the row and column headers. For this, the matrix elements of the 4X4 matrix is taken either row wise or column wise and at any position to start randomly for assignment to both row and column headers of the 16X16 matrix

Step 5: Now the row and column headers are combined to constitute for the 4 letter DNA sequence. This is illustrated in table 6 below.

Step 6: The standard universal amino acid table contains 20 amino acids. Since, the above table 5 would consists of 256 elements, the 20 amino acids are extended into 256 amino acids as follows. These 256 elements are and divided into four groups such as A, U, C, and G as follows:

A group – (A1, A2, A3, A4, A5, A6, A7, A8, A9, AA, AB, AC,AD, R1, R2, R3, R4, R5, R6, R7, R8, R9, RA, RB, RC,RD, N1,N2, N3, N4, N5, N6, N7, N8, N9, NA, NB, NC, ND, D1,D2,D3,D4,D5,D6,D7, D8, D9, DA, DB, DC, DD, C1, C2, C3, C4, C5, C6, C7, C8, C9, CA,CB,CC)

U group – (E1, E2, E3, E4, E5, E6, E7, E8, E9, EA, EB, EC,ED, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, QA, QB, QC, QD, G1, G2, G3, G4, G5, G6, G7, G8, G9, GA, GB, GC, GD, H1, H2, H3, H4, H5, H6, H7, H8, H9, HA, HB, HC, HD, I1, I2, I3, I4, I5, I6, I7, I8, I9, IA, IB, IC)

C group – (L1, L2, L3, L4, L5, L6, L7, L8, L9, LA, LB, LC,LD,K1, K2, K3, K4, K5, K6, K7, K8, K9, KA, KB, KC, KD, M1, M2, M3, M4, M5, M6, M7, M8, M9, MA, MB, MC,MD,F1, F2, F3, F4, F5, F6, F7, F8, F9, FA, FB, FC, FD, P1, P2, P3, P4, P5, P6, P7, P8, P9, PA, PB, PC)

G group – (S1, S2, S3, S4, S5, S6, S7, S8, S9, SA, SB, SC, SD, T1, T2, T3, T4, T5, T6, T7, T8, T9, TA, TB, TC, TD, W1, W2, W3, W4, W5, W6, W7, W8, W9, WA, WB, WC,WD, Y1,Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, YA, YB, YC,YD,V1,V2, V3, V4, V5, V6, V7, V8, V9, VA, VB, VC)

Step7: Now, these amino acids are assigned to the 16X16 matrix element of 4 letter DNA alphabet sequence available in table 7 using the collating sequence specified below

*Step8:*The 24 possible collating sequences using A, U, C, and G are as follows: 1) A,U,C,G 2) A,C,U,G 3) A,G,U,C 4) A,U,G,C 5) A,G,C,U 6) A,C,G,U 7) U,A,C,G 8) U,C,A,G 9) U,G,A,C 10) U,A,G,C 11) U,G,C,A 12) U,C,G,A 13) C,U,A,G 14) C,C,A,G 15) C,G,A,C 16) C,U,G,A 17) C,G,A,U 18) C,A,G,U 19)G,U,C,A 20) G,C,U,A 21) G,A,U,C 22) G,U,A,C 23) G,A,C,U 24) G,C,A,U

Step9: If the collating sequence 1 (A, U, C, G) is used, the amino acids of group1 followed by group2 followed by group3 & group 4 are assigned to the matrix elements either row wise or column wise as shown in Table 6. Other collating sequences have their own defined assignment of amino acid labels to the tRNA sequence generated in Table 5.

Step 5: The resultant protein sequence is called as the cipher text of the given plaintext.

Table.5. Amino Acid Table

	GA	UA	AA	CA	CU	AU	UU	GU	GC	UC	AC	CC	CG	AG	UG	GG
CG	CGGA	CGUA	CGAA	CGCA	CGCU	CGAU	CGUU	CGGU	CGGC	CGUC	CGAC	CGCC	CGCG	CGAG	CGUG	CGGG
AG	AGGA	AGUA	AGAA	AGCA	AGCU	AGAU	AGUU	AGGU	AGGC	AGUC	AGAC	AGCC	AGCG	AGAG	AGUG	AGGG
UG	UGGA	UGUA	UGAA	UGCA	UGCU	UGAU	UGUU	UGGU	UGGC	UGUC	UGAC	UGCC	UGCG	UGAG	UGUG	UGGG
GG	GGGA	GGUA	GGAA	GGCA	GGCU	GGAU	GGUU	GGGU	GGGC	GGUC	GGAC	GGCC	GGCG	GGAG	GGUG	GGGG
GC	GCGA	GCUA	GCAA	GCCA	GCCU	GCAU	GCUU	GCGU	GCGC	GCUC	GCAC	GCCC	GCCG	GCAG	GCUG	GCGG
UC	UCGA	UCUA	UCAA	UCCA	UCCU	UCAU	UCUU	UCGU	UCGC	UCUC	UCAC	UCCC	UCCG	UCAG	UCUG	UCGG

AC	ACGA	ACUA	ACAA	ACCA	ACCU	ACAU	ACUU	ACGU	ACGC	ACUC	ACAC	ACCC	ACCG	ACAG	ACUG	ACGG
CC	CCGA	CCUA	CCAA	CCCA	CCCU	CCAU	CCUU	CCGU	CCGC	CCUC	CCAC	CCCC	CCCG	CCAG	CCUG	CCGG
CU	CUGA	CUUA	CUAA	CUCA	CUCU	CUAU	CUUU	CUGU	CUGC	CUUC	CUAC	CUCC	CUCG	CUAG	CUUG	CUGG
AU	AUGA	AUUA	AUAA	AUCA	AUCU	AUAU	AUUU	AUGU	AUGC	AUUC	AUAC	AUCC	AUCG	AUAG	AUUG	AUGG
UU	UUGA	UUUA	UUAA	UUCA	UUCU	UUAU	UUUU	UUGU	UUGC	UUUC	UUAC	UUCC	UUCG	UUAG	UUUG	UUGG
GU	GUGA	GUUA	GUAA	GUCA	GUCU	GUAU	GUUU	GUGU	GUGC	GUUC	GUAC	GUCC	GUCG	GUAG	GUUG	GUGG
GA	GAGA	GAUA	GAAA	GACA	GACU	GAAU	GAUU	GAGU	GAGC	GAUC	GAAC	GACC	GACG	GAAG	GAUG	GAGG
UA	UAGA	UAUA	UAAA	UACA	UACU	UAAU	UAUU	UAGU	UAGC	UAUC	UAAC	UACC	UACG	UAAG	UAUG	UAGG
AA	AAGA	AAUA	AAAA	AACA	AACU	AAAU	AAUU	AAGU	AAGC	AAUC	AAAC	AACC	AACG	AAAG	AAUG	AAGG
CA	CAGA	CAUA	CAAA	CACA	CACU	CAAU	CAUU	CAGU	CAGC	CAUC	CAAC	CACC	CACG	CAAG	CAUG	CAGG

Table.6. Amino acid Table

	GA	**UA**	**AA**	**CA**	**CU**	**AU**	**UU**	**GU**	**GC**	**UC**	**AC**	**CC**	**CG**	**AG**	**UG**	**GG**
CG	CGGA	CGUA	CGAA	CGCA	CGCU	CGAU	CGUU	CGGU	CGGC	CGUC	CGAC	CGCC	CGCG	CGAG	CGUG	CGGG
	– E1	– Q4	– G7	– HA	– A1	– R4	– N7	– DA	– L1	– K4	– M7	– FA	– S1	– U4	– W7	– YA
AG	AGGA	AGUA	AGAA	AGCA	AGCU	AGAU	AGUU	AGGU	AGGC	AGUC	AGAC	AGCC	AGCG	AGAG	AGUG	AGGG
	– E2	– Q5	– G8	– HB	– A2	– R5	– N8	– DB	– L2	– K5	– M8	– FB	– S2	– U5	– W8	– YB
UG	UGGA	UGUA	UGAA	UGCA	UGCU	UGAU	UGUU	UGGU	UGGC	UGUC	UGAC	UGCC	UGCG	UGAG	UGUG	UGGG
	– E3	– Q6	– G9	– HC	– A3	– R6	– N9	– DC	– L3	– K6	– M9	– FC	– S3	– U6	– W9	– YC
GG	GGGA	GGUA	GGAA	GGCA	GGCU	GGAU	GGUU	GGGU	GGGC	GGUC	GGAC	GGCC	GGCG	GGAG	GGUG	GGGG
	– E4	– Q7	– GA	– HD	– A4	– R7	– NA	– DD	– L4	– K7	– MA	– FD	– S4	– U7	– WA	– YD
GC	GCGA	GCUA	GCAA	GCCA	GCCU	GCAU	GCUU	GCGU-	GCGC	GCUC	GCAC	GCCC	GCCG	GCAG	GCUG	GCGG
	– E5	– Q8	– GB	– I1	– A5	– R8	– NB	C1	– L5	– K8	– MB	– P1	– S5	- U8	– WB	– V1
UC	UCGA	UCUA	UCAA	UCCA	UCCU	UCAU	UCUU	UCGU	UCGC	UCUC	UCAC	UCCC	UCCG	UCAG	UCUG	UCGG
	– E6	– Q9	– GC	– I2	– A6	– R9	– NC	– C2	– L6	– K9	– MC	– P2	– S6	– U9	– WC	– V2
AC	ACGA	ACUA	ACAA	ACCA	ACCU	ACAU	ACUU	ACGU	ACGC	ACUC	ACAC	ACCC-	ACCG	ACAG	ACUG	ACGG
	– E7	– QA	– GD	– I3	– A7	– RA	– ND	– C3	– L7	– KA	– MD	P3	– S7	– UA	– WD	– V3
CC	CCGA	CCUA	CCAA	CCCA	CCCU	CCAU	CCUU	CCGU	CCGC	CCUC	CCAC	CCCC	CCCG	CCAG	CCUG	CCGG
	– E8	– QB	– H1	– I4	– A8	– RB	– D1	– C4	– L8	– KB	– F1	– P4	– S8	- UB	– Y1	– V4
CU	CUGA	CUUA	CUAA	CUCA	CUCU	CUAU	CUUU	CUGU	CUGC	CUUC	CUAC	CUCC	CUCG	CUAG	CUUG	CUGG
	– E9	– QC	– H2	– I5	– A9	– RC	– D2	– C5	– L9	– KC	– F2	– P5	– S9	– UC	– Y2	– V5
AU	AUGA	AUUA	AUAA	AUCA	AUCU	AUAU	AUUU–	AUGU	AUGC	AUUC	AUAC	AUCC	AUCG	AUAG	AUUG	AUGG
	– EA	– QD	– H3	– I6	– AA	– RD	D3	– C6	– LA	– KD	– F3	– P6	– SA	– UD	– Y3	– V6
UU	UUGA	UUUA	UUAA	UUCA	UUCU	UUAU	UUUU	UUGU	UUGC	UUUC	UUAC	UUCC	UUCG	UUAG	UUUG	UUGG
	– EB	– G1	– H4	– I7	– AB	– N1	– D4	– C7	– LB	– M1	– F4	– P7	– SB	– W1	– Y4	– V7
GU	GUGA	GUUA	GUAA	GUCA	GUCU	GUAU	GUUU	GUGU	GUGC	GUUC	GUAC	GUCC	GUCG	GUAG	GUUG	GUGG
	– EC	– G2	– H5	– I8	– AC	– N2	– D5	– C8	– LC	– M2	– F5	– P8	– SC	– W2	– Y5	– V8
GA	GAGA	GAUA	GAAA	GACA	GACU	GAAU	GAUU	GAGU	GAGC	GAUC	GAAC	GACC	GACG	GAAG	GAUG	GAGG
	– ED	– G3	– H6	– I9	– AD	– N3	– D6	– C9	– LD	– M3	– F6	– P9	– SD	– W3	– Y6	– V9
UA	UAGA	UAUA	UAAA	UACA	UACU	UAAU	UAUU	UAGU	UAGC	UAUC	UAAC	UACC	UACG	UAAG	UAUG	UAGG
	– Q1	– G4	– H7	– IA	– R1	– N4	– D7	– CA	– K1	– M4	– F7	– PA	– U1	– W4	- Y7	– VA
AA	AAGA	AAUA	AAAA	AACA	AACU	AAAU	AAUU	AAGU	AAGC	AAUC	AAAC	AACC	AACG	AAAG	AAUG	AAGG
	– Q2	– G5	– H8	– IB	– R2	– N5	– D8	– CB	– K2	– M5	– F8	– PB	– U2	– W5	– Y8	– VB
CA	CAGA	CAUA	CAAA	CACA	CACU	CAAU-	CAUU	CAGU	CAGC	CAUC	CAAC	CACC	CACG	CAAG	CAUG	CAGG
	– Q3	– G6	– H9	– IC	– R3	N6	– D9	– CC	– K3	– M6	– F9	– PC	– U3	– W6	– Y9	– VC

3.2.2. Decryption process

The decryption algorithm comprises the following steps for decrypting the above Cipher text into plaintext: The decryption process is a reversal of the encryption process.

Step 1: The Receiver receives the cipher text from the secure channel and clue from the alternate network and the receiver generates two DNA Encoding tables from his own clue and clue received from the sender using the DNA encoding algorithm[17].

Step 2: The cipher text of the protein sequence is divided into two halves equally. For e.g.:

Q5 L2 Cipher text D1 T7

Step 3: The Protein Sequences on both left side and right side are converted into tRNA sequence using amino acid table generated above. For e.g.:

UGUA UGGU tRNA Sequence CUUU GCAG

Step 4: The tRNA sequence is converted into mRNA sequence by replacing every DNA alphabet with its complement DNA alphabet. For e.g. A-U, U-A, G-C, C-G conversions are carried out. This process is a simulation of biological Reverse Translation. For e.g.:

ACAU ACCA mRNA Sequence GAAA CGUC

Step 5: The mRNA sequence is converted into DNA Sequence by replacing Uracil (U) with Thymine (T) on both left side and the right side. This process is a simulation of biological reverse Transcription. For e.g.:

ACAT ACCA Reverse Transcription GAAA CGTC

Step 6: The multiple round functions is applied on both left side and right side in order to improve the complexity of the decryption process and the steps involved in multiple round function are as follows:

a) The Intron sequence 1 and 2 are taken for transformation with the DNA sequence simultaneously for both left and right halves. In this step, the DNA sequence is XNOR-ed with Intron Sequence. This process is called transformation of DNA sequence. For e.g.:

CCTC CCCC CGCA AGTA

b) The DNA sequence is right shifted once on both the left & right side. For e.g.:

CTCC CCCC GCAA GTAC

c) The Shift Sequence is converted into mRNA sequence by Uracil (U) with Thymine (T) on both left side and the right side. This process is a simulation of biological Reverse Transcription. For e.g.:

CUCC CCCC GCAA GUAC

d) The mRNA sequence is converted into tRNA sequence by replacing every DNA alphabet with its complement DNA alphabet. For e.g. A-U, U-A, G-C, C-G conversions are carried out. This process is a simulation of biological Translation. For e.g.:

GAGG GGGG CGUU CAUG

e) The tRNA sequence is converted into DNA Sequence by replacing Uracil (U) with Thymine (T) on both left side and the right side. This process is a simulation of biological Reverse Transcription. For e.g.:

GAGG GGGG CGTT CATG

Step 7: In the Final Round, DNA sequence on both left side and the right side are transformed with Intron sequence 1 & Intron sequence 2 respectively. For e.g.:

AAGG ACAT Transformation DNA GCTT GAAG

Step 8: The transformed DNA sequence from left side and the right side are converted into plaintext using DNA Encoding tables. For e.g.:

Ba Plaintext Nk

Step 9: Finally, the plaintext from both left side and right side are merged together. For e.g.:

Bank.

4. Experimental Analysis

4.1. Time Taken for DNA Encryption and Decryption Algorithm.

The Figure 1 shows that time taken by encoding/decoding process and encryption/decryption algorithm.

Table.7. Time Taken for Encoding and Decoding

Input Size (words in count)	Encoding Time (ms)	Decoding Time (ms)
500	0.000006	0.000102
1000	0.000209	0.00027
2000	0.000868	0.000929
4000	0.003872	0.003361
6000	0.010531	0.011246
8000	0.023422	0.020673
10000	0.03894	0.029886

Table.8.Time Taken for Encryption and Decryption

Input Size (words in count)	Encryption Time (ms)	Decryption Time (ms)
500	0.00015	0.000173
1000	0.000313	0.000371
2000	0.000963	0.001037
4000	0.004175	0.004735
6000	0.012813	0.013377
8000	0.024362	0.021547
10000	0.040071	0.032520

4.2. The Frequency analysis of Cipher text

The Frequency analysis is the study of the frequency of letters or groups of letters in a cipher text. The method is used as an aid to breaking ciphers. This is because the correlation between the cipher text patterns is used as an effective tool in cryptanalysis leading to breaking of cipher text. In order to prove that the cipher texts generated out of the proposed algorithm has very minimal correlation – thereby decreasing the chance of its successful cryptanalysis, we have taken two cipher texts that have been generated using the proposed algorithm in the following two scenarios

1. Cipher texts for two plain texts generated using the same encoding tables
2. Cipher texts for two plain texts generated using different encoding tables (because encoding tables are generated new for every interaction session between sender and receiver)

The two cipher text relationship in the two scenarios is tested using Pearson's correlation coefficient method. The result of the correlation analysis is that the two cipher texts have weak relationships in both the above scenarios enabling reducing the possibilities of cryptanalysis and breaking the cipher. This is depicted in figures 4&5.

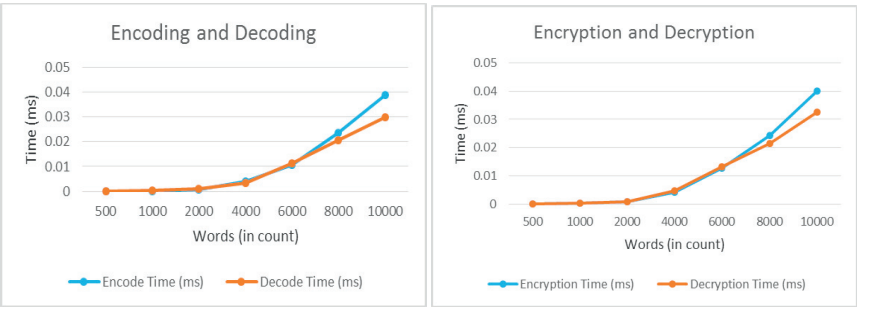

Fig.1. Time Taken for Encoding & Decoding and Encryption & Decryption

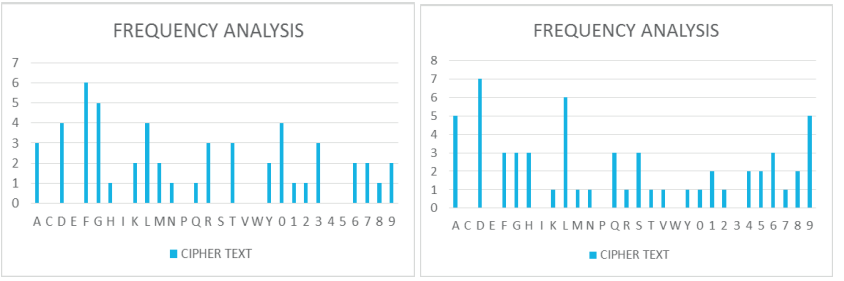

Figure 2.Cipher texts for two plain texts generated using the same encoding tables.

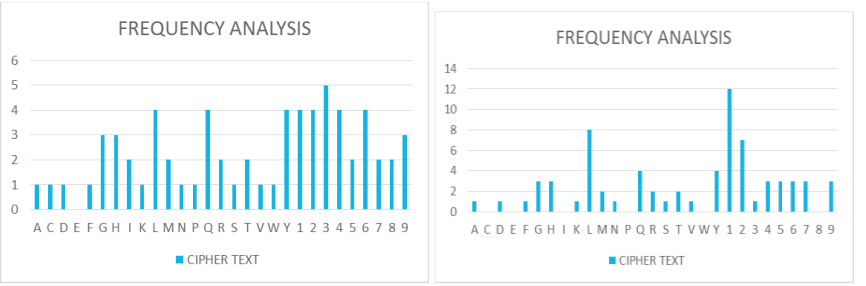

Figure 3.Cipher texts for two plain texts generated using the different encoding tables.

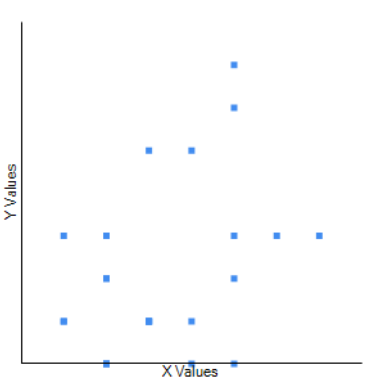

Fig.4. Correlation analysis of Scenario1

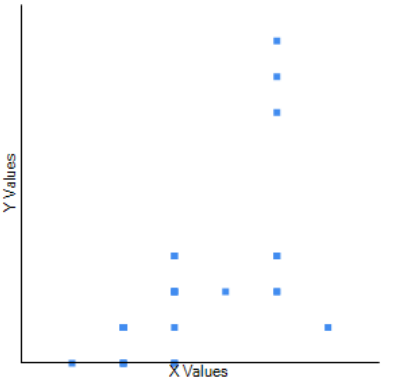

Fig.5. Correlation analysis of Scenario2

5. Fulfilment of Requirements

In this algorithm it is obvious that the stipulated requirements are completely fulfilled.

Table.9. Fulfilment of requirements of proposed algorithm

Sl.No	Requirements	Fulfilment
1	DNA Encoding of Complete character set fulfilment	The encoding table provides for encoding sequences for the complete character set as shown in table 5.
2	Dynamic Encoding Table Generation	The Encoding Table is generated randomly after every session interval between the sender and receiver [17].
3	Unique sequence for encoding of every character of plaintext to DNA sequence	The encoding of plaintext into DNA sequence is unique for every element of the character set in every generation of encoding table in every session.
4	Robustness of encoding.	The robust encoding scheme is provided due to the more randomness.
5	Biological Process Simulation	The DNA encryption and decryption algorithm is based on the transcription and translation processes of CDMB which is simulated to adapt to digital computing environment.
6	Dynamicity of encryption process	This requirement is fulfilled. The same plaintext can produce different cipher text for every session due to unique DNA encoding table generation for every session.

6. Conclusion

In this paper a novel and unique biological simulation based technique for DNA encryption and decryption, which fulfills all of the functional and non-functional attributes that should be characteristic of a DNA computing based encryption algorithm has been developed. The analysis of performance illustrates the strength of the algorithm.

References

1. Akanksha Agrawal, Akansha Bhopale, Jaya Sharma, Meer Shizan Ali, and Divya Gautam. Implementation of DNA algorithm for secure voice communication. *International Journal of Scientific & Engineering Research* 2012; **3**.
2. Guangzhao Cui, Cuiling Li, Haobin Li, Xiaoguang Li. DNA Computing and Its Application to Information Security Field. In: Proceedings of the 5th International Conference of Natural Computation: 2009 Aug 14-16; Tianjian, China; IEEE; 2009.
3. Guangzhao Cui, Limin Qin, Yanfeng Wang, Xuncai Zhang. An encryption scheme using DNA technology. In: Proceedings of the 3rd International Conference on Bio-Inspired Computing: Theories and Applications; 2008 Sep 28 – Oct 1; United States. IEEE; 2008. p 37-42.
4. Kang Ning. A Pseudo DNA Cryptography Method. http://arxiv.org/abs/0903.269; 2009.

5. Li Xin she, Zhang Lei, Hu Yu pu. A Novel Generation Key Scheme Based on DNA. In: Proceedings of the International Conference on Computational Intelligence and Security; 2008.p. 264-266.
6. Mona Sabry, Mohamed Hashem, Taymoor Nazmy. Three Reversible Data Encoding Algorithms based on DNA and Amino Acids Structure. International Journal of Computer Applications 2012; **54**: 0975 – 8887
7. NRDC, Govt. of India, http://www.nrdcindia.com/Patent%20Asistance%20 (in%20India) %20Form%202011.pdf
8. O Tornea, ME Borda. DNA Cryptographic Algorithms. In: IFMBE Proceedings of the International Conference on Advancements of Medicine and Health Care through Technology: 2009 Sep 23-26; Cluj-Napoca, Romania. Springer; 2009. p 223-226.
9. Padma Bt. *DNA computing theory with ECC'* http://www.scribd.com/doc/55154238/Report, 2010.
10. Qiang Zhang, Ling Guo, Xianglian Xue, Xiaopeng Wei. An image encryption algorithm based on DNA sequence addition operation. In: Proceedings of 4th International Conference on *Bio*-Inspired Computing: Theories and Applications. IEEE; 2009; 16-19.
11. Souhila Sadeg, Mohamed Gougache, Mansouri N, Drias H. An encryption algorithm inspired from DNA. In: Proceedings of the International Conference on Machine and Web Intelligence. IEEE; 2010; p.344-349.
12. Sherif T Amin, MagdySaeb, Salah El Gindi. A DNA-based Implementation of YAEA Encryption Algorithm. In: Proceedings of the International Conference on Computational Intelligence. IASTED; 2006; p.120-125.
13. X Guozhen, L Mingxin, Q Lei, L Xuejia. New field of cryptography: DNA Cryptography. *Chinese Science Bulletin*. Springer Verlag, Germany; 2006; 51:1413-1420.
14. Xing Wang, Qiang Zhang, *'DNA computing- based cryptography'*, *IEEE.2009,* pp. 67-69.
15. Xiutang Geng, Linqiang Pan, Jin xu. A DNA Sticker algorithm for bit substitution in a block cipher. *Journal of Parallel and Distributed Computing*; *2008*; 68:1201-1206.
16. Zhang, Qiang, Wang, Qian, Wei, Xiaopeng. A Novel Image Encryption Scheme based on DNA Coding and Multi-Chaotic Map. Advanced Science Letters; 2010; 3:447-451.
17. Noorul Hussain U, Chithralekha T, inventors; assignee. A Novel DNA Encoding Technique and System for DNA Cryptography. India Patent 5107, CHE, 2012. 2012 Dec 7.