Encryption and Decryption Algorithm Based DNA Sequence

Ass. Lec. INTEDHAR SHAKIR NASIR
College of Medicine\ University of Kerbala

خوارزمية التشفير وفك التشفير على أساس تسلسل الحمض النووي

م.م. انتظار شاكر ناصر

كلية الطب/ جامعة كربلاء

intedhar.shakir@uokerbala.edu.iq

الخلاصة

استخدام علم الأحياء في التشفير هو مسار واعد في مجال التشفير. في يومنا هذا، يتم اقتراح العديد من طرق الحمض النووي لمشكلات التشفير والتشفير وعلم إخفاء المعلومات، وهي فعالة في هذه المجالات. هناك بعض القيود في حسابات تشفير الحمض النووي الحالية، والتي تشمل استخدام تشفير الرياضيات الدائرة المنسقة في جزء من المطالبة مراحلها. بالتناوب، قد لا يكون التحليل المخبري الحي مناسبًا في مجال الطبيعة المتقدمة للتسجيل. في هذه الورقة، نقدم طريقة تشفير جديدة ومبنية على أساس الحمض النووي للتشفير، مستوحاة من بنية الحمض النووي وعلاقتها بالأحماض الأمينية في جدول الشفرة الوراثية القياسية. تشرح الورقة تقنية جديدة لتحويل البيانات من النموذج الثنائي إلى نموذج الحمض النووي (أو الحمض النووي الريبي) ثم إلى شكل الأحماض الأمينية والعكس.

الكلمات المفتاحية: فك التشفير، الحامض النووي، تسلسل الحامض النووي ، الأحماض الأمينية.

Abstract

Using biology in cryptography is a promising track in cryptographic field. Now a day, several DNA methods are suggested for cryptography, cryptanalysis and steganography problems, they are effective in these fields. There are some limitations in current DNA cryptography calculations, which include utilizing coordinated circuit math cryptography at a portion from claiming their stages. Alternately, living laboratory analysis may not be proper in the advanced registering nature’s domain. In this paper, we introduce a new, excellent DNA-based encryption method of encoding, inspired from the DNA structure and its relation to the amino acids in the standard genetic code table. The paper explains a new technique to convert data from binary form to DNA (or RNA) form then to amino acids' form and the reverse.

Keywords: Encryption; Decryption, DNA Cryptography, DNA Sequence, Amino acids.

1. Introduction

Nucleic acid (DNA) is an extended linear polymer found in the center of a cell. A DNA sequence is a sequence containing of four alphabets: A, C, G and T. Each of them is related to a nucleotide [10]. DNA is composed of numerous nucleotides in the form of a double spring, and it is connected with the transmission of heritable information.

Cryptography is a field of science that uses mathematics to encrypt and decrypt data for secure communication. It allows the user to transfer data in an insecure network so that it cannot be recited by anyone except the planned recipient.

DNA cryptography is a new native cryptographic field that results from research on DNA computing, in which DNA is used as a data transporter and current biological knowledge is used as application means. The vast parallelism and unusual information solidity inherent in DNA particles are used for cryptographic drives, such as encryption, authentication, and signature [1, 2]. The novel DNA cryptography [3, 7] is distant from established methods both in theory and understanding. The constraints of its high-tech laboratory requirements and computational limitations, combined with the labor-intensive extrapolation means, all prevent DNA computing from being efficiently used in today's security world.

DNA cryptography uses different processes to encode data. Different DNA cryptography methodologies are used for secure message transmission, such as polymerase chain reaction (PCR), biomolecular technique, and one-time pad. The PCR technique is a DNA digital coding technique, in which messages are converted from hexadecimal code into binary code and further converted into a DNA sequence, which is used in a DNA template. The biomolecular technique uses parallel processing capabilities of biomolecular computation. The one-time pad technique is used to encrypt and decrypt images [1]. In last years, much research work has been done on DNA based encryption schemes [9, 4, 11]. Most of them use biological properties of DNA sequences.

Secure communication is critical to facilitate trust exchange of data and information between a sender and receiver [8]. Today, the internet has become very important for all banking and electronic trade transactions such that the communication is made in a greatly secure method. To achieve these security necessities, many techniques and systems have been established in the mathematical cryptography for encrypting and decrypting plaintext. However, these methods are limited compared with DNA cryptography techniques. DNA cryptography is a developing field in DNA computing research.

In our work, we applied the transformation of character form of data to the DNA form and then to amino acid form. The importance of such transformation lies mainly in representing data in a biological form that can make data be able to go through biological experiments and processes, especially related to Amino Acids and DNA. It is also a way of viewing data moving through biological processes.

The next sections are organized as follows: Section 2 explains a biological background information that helps understanding the biological ideas involved in our algorithm. Section 3 describes the related work, in Section 4, we explained the proposed algorithm,

In Section 5, the analysis of experimental results are presented. Section 6 gives the conclusion of this work.

2. Biological Background

DNA is a nucleic acid that holds the genetic commands used in the growth and working of all known living creatures and some viruses. The DNA double helix is steadied by hydrogen bonds between the bases involved in the two strands. The four bases found in DNA are adenine (abbreviated A), cytosine (C), guanine (G), and thymine (T) [6].

A gene is defined as a sequence of DNA that holds genetic data. The genetic code contains three-letter "words" called codons that form from a sequence of three nucleotides (e.g., ACT, CAG, and TTT). Given that there are four bases in three-letter combinations, there are 64 probable codons (43 combinations). These codons encode the 20 standard amino acids, giving most amino acids more than one possible codon. Moreover, three "stop" or "nonsense" codons signify the end of the coding region, namely, the TAA, TGA, and TAG codons [6].

3. Related Work

No common theory exists for applying DNA particles in cryptography [7, 8]. Some key technologies in DNA research, such as PCR, DNA synthesis, and DNA digital coding, have only been developed and well accepted in recent years [9]. In 1999, Clelland et al. [9] completed a method to steganography by thumping secret messages encoded as DNA elements between a multitudes of random DNA sequences. In 2000, a one-time pad instrument was proposed by Gehani [8] based on DNA, who built two encryption approaches of one-time pads of DNA sequences. One approach converts a stable-length DNA plain code sequence cell into a DNA cryptograph sequence according to the defined mapping graph (substitution). The second approach, called the exclusive-OR method, uses biological molecular methods to transmit through exclusive-OR operation of DNA plain code and cipher key sequence.

4. Proposed Algorithm

The proposed algorithm involves two parts:

Part 1: Encoding algorithm using DNA computing.

Part 2: Decryption algorithm using DNA computing.

After converting data to binary form, can transferred it to DNA or RNA form according to Table 1. The difference between DNA and RNA form is that letter "T" in DNA is the same as letter "U" in RNA. After converting data to the RNA form, it is transferred to the Amino acids form based on standard Amino Acid Table.

4.1. Encoding Algorithm Using DNA Computing

Table 1 shows the sample output of the proposed DNA encoding algorithm. After every pre-defined session interval, the DNA encoding table is generated. The DNA sequences and transfer of alphabets to these sequences differ across various sessions.

Table 1. DNA and RNA Representation of bits

Bit 1	**Bit 2**	**RNA**	**DNA**
0	0	**A**	**A**
0	1	**T**	**T**
1	0	**C**	**C**
1	1	**G**	**G**

4.1.1 DNA Computing-based Encoding Algorithm

In this paper, we proposed an encryption and decryption algorithm. The algorithm uses Table 1 to convert one half of the plaintext into a DNA sequence, which is available with the sender. Table 2 will obtained from the receiver, the other half of the plaintext is converted into a DNA sequence.

After receiver send Table 2, the output of the DNA encoding algorithm is shown in Table 2. The algorithm steps includes encoding and decoding of the given input text message. First plaintext is given as input. Each character of the plaintext is mapped with nucleotides which will be the first level of security for secure data communication.

Table 2. DNA Encoding Table [5]

	C	**G**	**T**	**A**
A	ACAT- a	AGAA – y	ATAA– W	AAAG – {
	ACTG- b	AGTT – z	ATTT – X	AATA – [
	ACCC- c	AGCC – A	ATCG – Y	AACG - }
	ACGA – d	AGGG– B	ATGC – Z	AAGG -]
T	TCAT – e	TGAT – C	TTAA – 0	TAAA - |
	TCTG – f	TGTG – D	TTTT – 1	TATT - \
	TCCG – g	TGCC – E	TTCC – 2	TACG - +
	TCGT – h	TGGA – F	TTGG - 3	TAGC - =
G	CCAG – i	CGAT – G	CTAT – 4	CAAA - _
	CCTA – j	CGTG – H	CTTG – 5	CATT - -
	CCCG – k	CGCG – I	CTCC – 6	CACC -)
	CCGG – l	CGGT – J	CTGA – 7	CAGG – (
C	CGAA – m	GGAG– K	GTAT – 8	GAAT - *
	CGTT – n	GGTA – L	GTTG – 9	GATG - &
	CGCG – o	GGCG– M	GTCG – <	GACC - ^
	CGCG – p	GGGG– N	GTGT - >	GAGA - %

A	ACTC – q	AGTA – O	ATTA - ,	AATT - $
	ACCG – r	AGCG – P	ATCC - .	AACC - #
T	TCTC – s	TGTC – Q	TTTA - ?	TATA - @
	TCCC – t	TGCG – R	TTCG - /	TACC - !
C	CCTT – u	CGTC – S	CTTC - :	CATA - ~
	CCCC – v	CGCC – T	CTCG - ;	CACG - `
G	GCTA – w	GGTT – U	GTTC – “	GATC - €
	GCCC – x	GGCC – V	GTCC – ‘	GACG - £

4.2. Encryption Processing

The encryption operation starts by encrypting plaintext into ciphertext. Before encryption processing starts, the encoding processes are carried out for the conversion of plaintext to DNA sequence.

1. As shown in Figure 1, a sender will generate a DNA encoding table for the encoding of plaintext into a DNA sequence using a DNA encoding algorithm.

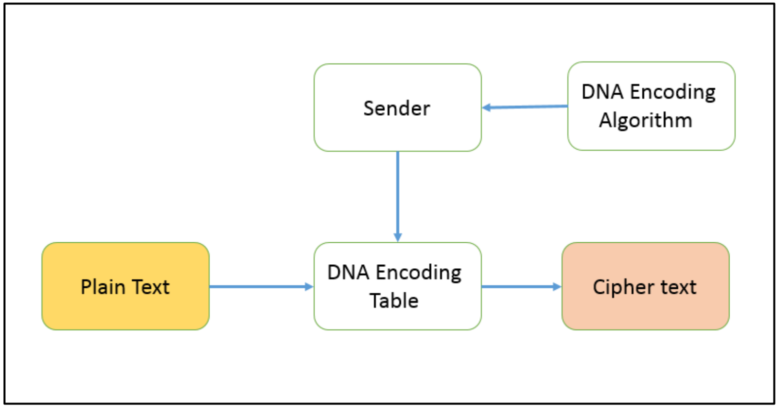


Figure 1. Generating DNA Encoding Table and Creating of Cipher Text

2. As shown in Figure 2, the receiver sends a hint to the sender that uses it to generate a code (Table 2) through the same encoding technique.

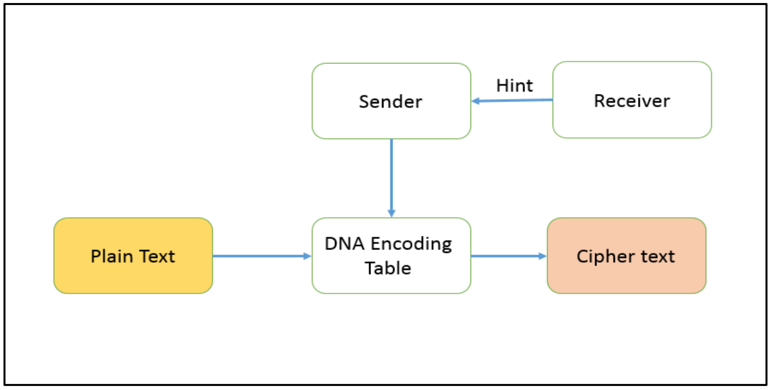


Figure 2. Generating DNA Encoding Table 2

3. The plaintext is separated into two parts. In the case when the plaintext is not even, the algorithm appends one random number in the last part to make both parts even.
4. The algorithm uses Table 1 to convert one half of the plaintext into a DNA sequence, which is available with the sender. Using DNA encoding Table 2, which is obtained from the receiver, the other half of the plaintext is converted into a DNA sequence.

Example: If the plaintext contains four characters, it will be divided into two equal parts assuming that the DNA sequences obtained using encoding Tables 1 and 2 are as follows.

5. The algorithm applies multiple-round function on both plaintext sides. The number of function rounds will be greater than or equal to 5.

Before explaining the steps of multiple-round function, we will define some important ideas:

- **An Intron Sequence** is the sequence that is used in the generation of encoding Tables 1 and 2. In each encoding operation, two intron sequences exist: one is generated by the sender and the other is generated by the receiver.
- **The Transformation Operation** with the DNA-encoded plaintext sequences for both left and right halves, which involves an XNOR of the DNA-encoded plaintext with the respective intron sequences.

The steps of multiple-round function are as follows:

1- Converting the plaintext to a binary representation by XNOR operations using the following mapping (Table 3):

Table 3. DNA and RNA Representation of bits.

Bit 1	Bit 2	RNA	DNA
0	0	**A**	**A**
0	1	**T**	**T**
1	0	**C**	**C**
1	1	**G**	**G**

2- Converting the transformed DNA sequence into an mRNA sequence by changing each Thymine (T) with Uracil (U) on both DNA side sequences.

3- Converting the mRNA sequence into a tRNA sequence by replacing every DNA alphabet with its complement DNA alphabet (A-U, U-A, G-C, and C-G).

4- Converting the tRNA sequence into a DNA sequence by replacing Uracil (U) with Thymine (T) in the tRNA sequences.

5- The DNA sequence is right-shifted from reverse transcription on both sides.

6- 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. A suitable amino acid table is then generated. The process for amino acid table generation is described below.

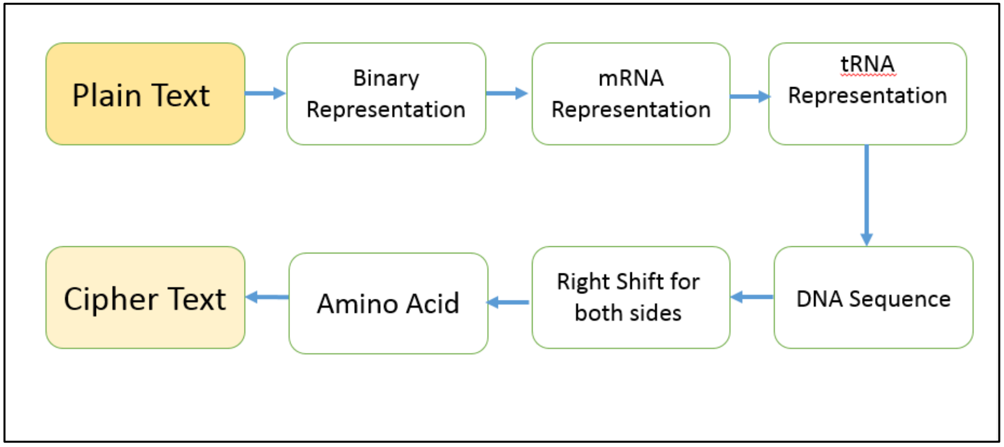


Figure 3. Steps for Converting Plain Text to Cipher Text

7- **Generating Amino Acid Table**: Generating an amino acid includes the following steps:

A: Randomly generating two sequences of DNA that involves four DNA alphabets. The produced sequences should have all the four chemical mixes of DNA, and the two sequences should not match.

B: Changing the two generating sequences into mRNA.

C: Generating a 4 × 4 matrix includes two mRNA sequences randomly selected from a row and column. The matrix elements are products of the corresponding row and column elements. Each matrix holds a two-letter DNA alphabet sequence, as shown in Table 2. Based on the two sequences above AGUC and CGAU, the table will be generated as Table 4 follows:

Table 4. Amino Acid Table (4 X 4)

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

D: The first matrix will be extended to 16 × 16 matrix using matrix elements of the above 4 × 4, as shown in Table 5.

Table 5. Amino Acid Table (16x16)

	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

E: The row and column headers are joined to establish a four-letter DNA sequence, as shown in Table 6.

F: The slandered amino acid table involves 256 amino acids. As shown in Table 6, which includes only 20 elements, these elements will be extended into 256 amino acids as follows. These 256 elements are divided into four groups, namely, 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)

The four groups of amino acids are allocated to the elements of the matrix either row-wise or column-wise, as shown in Table 6.

Table 6. Amino Acid Table (256)

	GA	**UA**	**AA**	**CA**	**CU**	**AU**	**UU**	**GU**	**GC**	**UC**	**AC**	**CC**	**CG**	**AG**	**UG**	**GG**
CG	CGGA – E1	CGUA – Q4	CGAA – G7	CGCA – HA	CGCU – A1	CGAU – R4	CGUU – N7	CGGU – DA	CGGC – L1	CGUC – K4	CGAC – M7	CGCC – FA	CGCG – S1	CGAG – U4	CGUG – W7	CGGG – YA
AG	AGGA – E2	AGUA – Q5	AGAA – G8	AGCA – HB	AGCU – A2	AGAU – R5	AGUU – N8	AGGU – DB	AGGC – L2	AGUC – K5	AGAC – M8	AGCC – FB	AGCG – S2	AGAG – U5	AGUG – W8	AGGG – YB
UG	UGGA – E3	UGUA – Q6	UGAA – G9	UGCA – HC	UGCU – A3	UGAU – R6	UGUU – N9	UGGU – DC	UGGC – L3	UGUC – K6	UGAC – M9	UGCC – FC	UGCG – S3	UGAG – U6	UGUG – W9	UGGG – YC
GG	GGGA – E4	GGUA – Q7	GGAA – GA	GGCA – HD	GGCU – A4	GGAU – R7	GGUU – NA	GGGU – DD	GGGC – L4	GGUC – K7	GGAC – MA	GGCC – FD	GGCG – S4	GGAG – U7	GGUG – WA	GGGG – YD
GC	GCGA – E5	GCUA – Q8	GCAA – GB	GCCA – I1	GCCU – A5	GCAU – R8	GCUU – NB	GCGU- C1	GCGC – L5	GCUC – K8	GCAC – MB	GCCC – P1	GCCG – S5	GCAG - U8	GCUG – WB	GCGG – V1
UC	UCGA – E6	UCUA – Q9	UCAA – GC	UCCA – I2	UCCU – A6	UCAU – R9	UCUU – NC	UCGU – C2	UCGC – L6	UCUC – K9	UCAC – MC	UCCC – P2	UCCG – S6	UCAG – U9	UCUG – WC	UCGG – V2
AC	ACGA – E7	ACUA – QA	ACAA – GD	ACCA – I3	ACCU – A7	ACAU – RA	ACUU – ND	ACGU – C3	ACGC – L7	ACUC – KA	ACAC – MD	ACCC- P3	ACCG – S7	ACAG – UA	ACUG – WD	ACGG – V3
CC	CCGA – E8	CCUA – QB	CCAA – H1	CCCA – I4	CCCU – A8	CCAU – RB	CCUU – D1	CCGU – C4	CCGC – L8	CCUC – KB	CCAC – F1	CCCC – P4	CCCG – S8	CCAG - UB	CCUG – Y1	CCGG – V4
CU	CUGA – E9	CUUA – QC	CUAA – H2	CUCA – I5	CUCU – A9	CUAU – RC	CUUU – D2	CUGU – C5	CUGC – L9	CUUC – KC	CUAC – F2	CUCC – P5	CUCG – S9	CUAG – UC	CUUG – Y2	CUGG – V5
AU	AUGA – EA	AUUA – QD	AUAA – H3	AUCA – I6	AUCU – AA	AUAU – RD	AUUU – D3	AUGU – C6	AUGC – LA	AUUC – KD	AUAC – F3	AUCC – P6	AUCG – SA	AUAG – UD	AUUG – Y3	AUGG – V6
UU	UUGA – EB	UUUA – G1	UUAA – H4	UUCA – I7	UUCU – AB	UUAU – N1	UUUU – D4	UUGU – C7	UUGC – LB	UUUC – M1	UUAC – F4	UUCC – P7	UUCG – SB	UUAG – W1	UUUG – Y4	UUGG – V7
GU	GUGA – EC	GUUA – G2	GUAA – H5	GUCA – I8	GUCU – AC	GUAU – N2	GUUU – D5	GUGU – C8	GUGC – LC	GUUC – M2	GUAC – F5	GUCC – P8	GUCG – SC	GUAG – W2	GUUG – Y5	GUGG – V8

GA	GAGA – ED	GAUA – G3	GAAA – H6	GACA – I9	GACU – AD	GAAU – N3	GAUU – D6	GAGU – C9	GAGC – LD	GAUC – M3	GAAC – F6	GACC – P9	GACG – SD	GAAG – W3	GAUG – Y6	GAGG – V9
UA	UAGA – Q1	UAUA – G4	UAAA – H7	UACA – IA	UACU – R1	UAAU – N4	UAUU – D7	UAGU – CA	UAGC – K1	UAUC – M4	UAAC – F7	UACC – PA	UACG – U1	UAAG – W4	UAUG - Y7	UAGG – VA
AA	AAGA – Q2	AAUA – G5	AAAA – H8	AACA – IB	AACU – R2	AAAU – N5	AAUU – D8	AAGU – CB	AAGC – K2	AAUC – M5	AAAC – F8	AACC – PB	AACG – U2	AAAG – W5	AAUG – Y8	AAGG – VB
CA	CAGA – Q3	CAUA – G6	CAAA – H9	CACA – IC	CACU – R3	CAAU-N6	CAUU – D9	CAGU – CC	CAGC – K3	CAUC – M6	CAAC – F9	CACC – PC	CACG – U3	CAAG – W6	CAUG – Y9	CAGG – VC

Other collating sequences have their own defined assignment of amino acid labels to the tRNA sequence generated in Table 6.

8: The result protein sequence is called the ciphertext of the given plaintext.

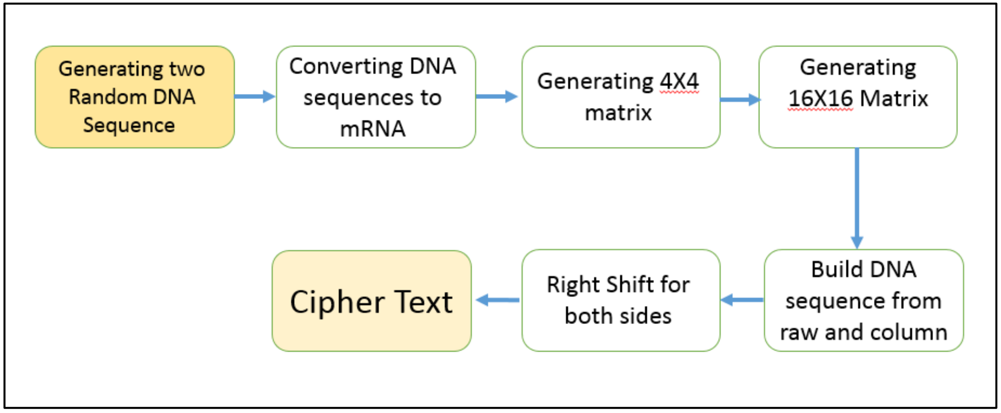

Generating two Random DNA Sequence
Converting DNA sequences to mRNA
Generating 4X4 matrix
Generating 16X16 Matrix
Build DNA sequence from raw and column
Right Shift for both sides
Cipher Text

Figure 4. Generating an Amino Acid

4.3. Decryption Process

The decryption algorithm includes the following steps for decrypting the ciphertext into plaintext:

1- After the cipher text and clue are received by the receiver from the sender, two tables are then generated for DNA encoding by the receiver using two clues. The first clue is the receiver's own clue, and the second clue is sent by the sender using the DNA encoding algorithm.

2- Dividing the ciphertext of the protein sequence into two halves equally.

3- Using the amino acid table, the protein sequences on both sides will converted into a tRNA sequence.

4- Converting the tRNA sequence into an mRNA sequence by exchanging every DNA alphabet with its complement DNA alphabet.

Example:

C-G, A-U, U-A, G-C

5- Converting the mRNA sequence into a DNA sequence by changing each U with T on both sides.

6- Applying the multiple-round functions on both sides to increase the difficulty of the decryption procedure. The steps in multiple-round function are as follows:

a. Sequences 1 and 2 are simultaneously taken for transformation with the DNA sequence for both left and right parts. The next step is transformation of the DNA sequence (i.e., the DNA sequence is XNOR-ed with the intron sequence).

b. The DNA sequence is right-shifted once on both the left and right sides.

c- The shift sequence is converted into an mRNA sequence by Uracil (U) with Thymine (T) on both the left and right sides. This process is a simulation of biological reverse transcription. For example:

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

e- The tRNA sequence is converted into a DNA sequence by replacing Uracil (U) with Thymine (T) on both the left and right sides. This process is a simulation of biological reverse transcription.

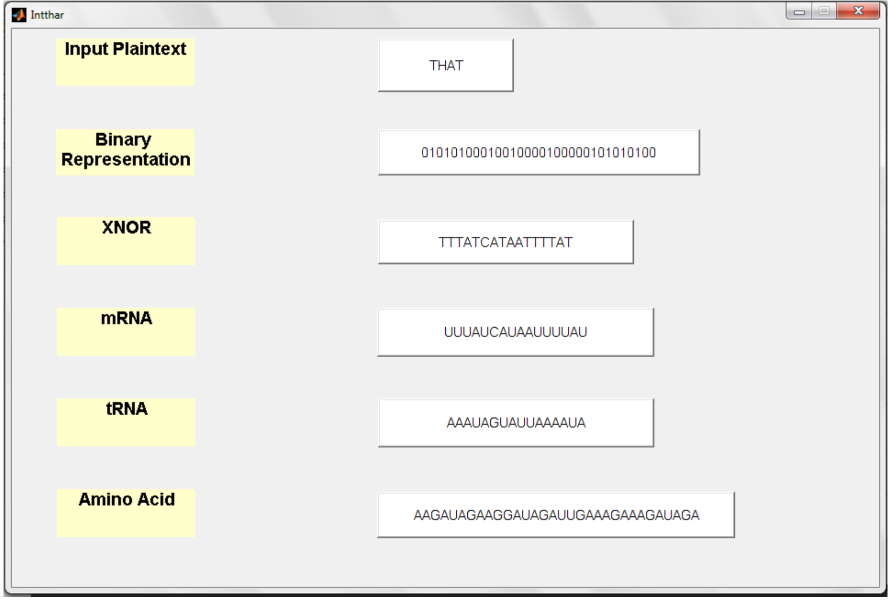
f. The tRNA sequence is converted into a DNA sequence by replacing Uracil (U) with Thymine (T) on both the left and right sides. This process is a simulation of biological reverse transcription.

In the final round, the DNA sequences on both the left and right sides are transformed with intron sequence 1 and intron sequence 2, respectively.

The transformed DNA sequence from the left and right sides are converted into plaintext using DNA encoding tables. For example,

5. Experimental and Performance Analysis

The experiments are simulated using MATLAB software (Version 8.1), the input is pliantext and the output is ciphertext, Figure 5 shows the output of encryption process.



Input Plaintext: THAT

Binary Representation: 01010100010010000100000101010100

XNOR: TTTATCATAATTTTAT

mRNA: UUUAUCAUAAUUUUAU

tRNA: AAAUAGUAUUAAAAUA

Amino Acid: AAGAUAGAAGGAUAGAUUGAAAGAAAGAUAGA

Figure 5. Encryption Processing

5. Security Analysis

A good encryption procedure must robust against all types of cryptanalytic attack. The study of the letter's frequency in a cipher text called frequency analysis. This study considered as a method to breaking ciphers. In this section we depends on calculating the correlation between the cipher text patterns [1]. The generated cipher texts has a high security if it has very minimal correlation.

For test the efficiency of proposed algorithm, we test two cipher texts that have been generated using the proposed algorithm for two types of cipher text, the first cipher texts for two plain texts that use the same encoding table, the second cipher texts when we use different encoding tables for each plain texts (a new encoding tables are generated for every interaction session between sender and receiver). The two cipher text are tested using Pearson’s correlation coefficient method (SPSS tool).

In comparison to the correlation coefficient results, using the same encoding table increased the relationship's influence between the plaintext and the ciphertext, which decreasing the opportunities of breaking the cipher as shown in Figure 6 and 7.

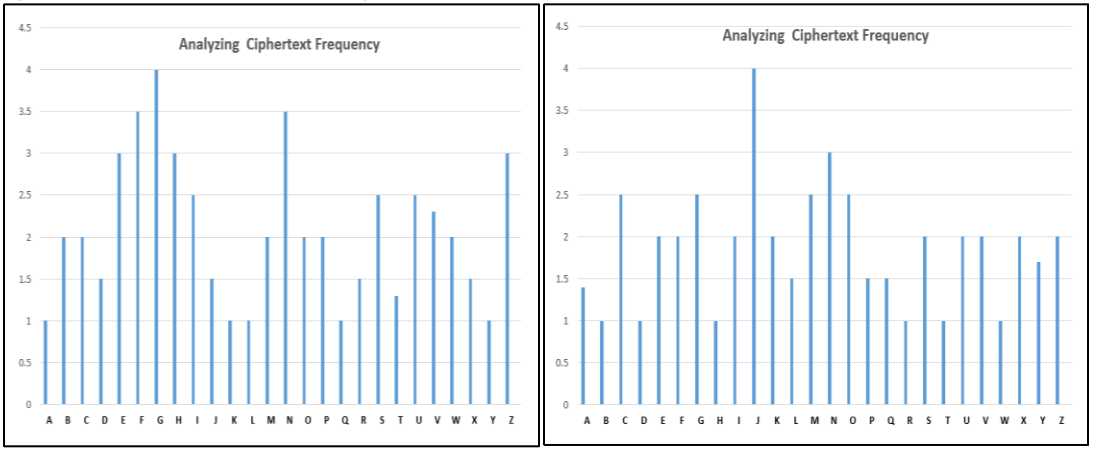


Figure 6. Correlation rates of cipher texts that are generated using the same encoding tables

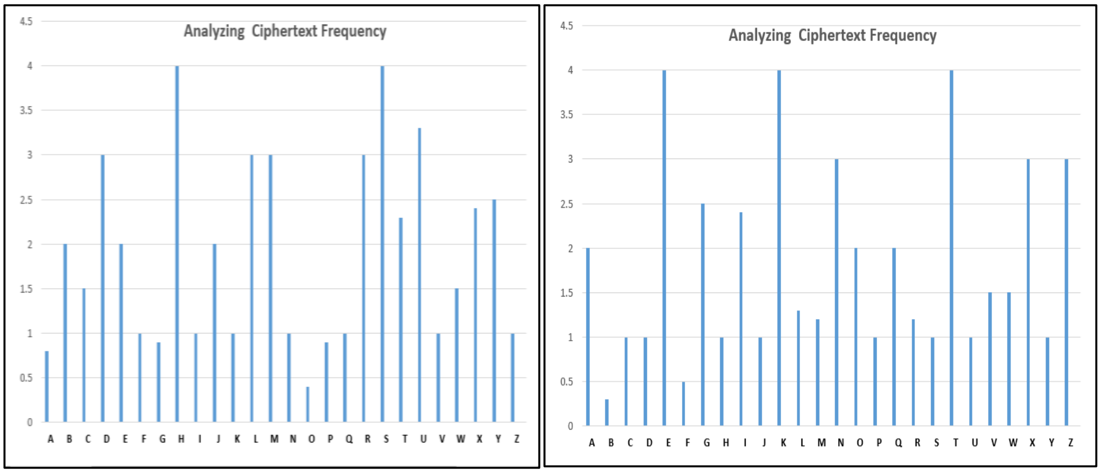


Figure 7. Correlation rates of cipher texts that are generated using the different encoding tables

6. Conclusion

DNA cryptography is a new natural cryptographic field that was developed from research on DNA computing. Numerous methods and systems have been established based on modular arithmetic cryptography for encryption and decryption. However, these methods are eliminated using DNA cryptography systems and methods. Some algorithms that are available in DNA cryptography have restrictions in that they use sectional arithmetic cryptography at some of their phases. In this study, a new biological simulation-based algorithm for DNA encryption and decryption was proposed. The proposed algorithm satisfies all the attributes that should be characteristic of a DNA computing-based encryption algorithm. Data analysis shows the performance of the algorithm.

Reference

[1] Kari, L. (1997) DNA Computing: Arrival of Biological Mathematics. The Mathematical Intelligencer, 19 , 9–22.

[2] Kartalopoulos, S. V. (2005) DNA-inspired Cryptographic Method in Optical Communications, Authentication and Data Mimicking. In Military Communications Conference. MILCOM, IEEE (774-779).

[3] Cui, G., Qin, L., Wang, Y., & Zhang, X. (2007). Information security technology based on DNA computing. In Anti-counterfeiting, Security, Identification, 2007 IEEE International Workshop on (288-291).

[4] Leier, A., Richter, C., Banzhaf, W., & Rauhe, H. (2000). Cryptography with DNA binary strands. Biosystems, 57(1), 13-22.

[5] UbaidurRahman, N. H., Balamurugan, C., & Mariappan, R. (2015). A Novel DNA Computing Based Encryption and Decryption Algorithm. Procedia Computer Science, 46, 463-475.

[6] Sabry, M., Hashem, M., & Nazmy, T. (2012). Three Reversible Data Encoding Algorithms based on DNA and Amino Acids Structure. International Journal of Computer Applications, 54(8), 24-30.

[7] Lu, M., Lai, X., Xiao, G., & Qin, L. (2007). Symmetric-key cryptosystem with DNA technology. Science in China Series F: Information Sciences, 50(3), 324-333.

[8] Gehani, A., LaBean, T., & Reif, J. (2004). DNA-based cryptography. In Aspects of Molecular Computing (pp. 167-188). Springer Berlin Heidelberg.

[9] Clelland, C. T., Risca, V., & Bancroft, C. (1999). Hiding messages in DNA microdots. Nature, 399(6736), 533-534.

[10] Hsu, H. Z., & Lee, R. C. T. (2006). DNA based encryption methods. The 23rd Work—shop on Combinatorial Mathematics and Computation Theory, National Chi Nan University Puli, Nantou Hsies, Taiwan, 545.

[11] Shimanovsky, B., Feng, J., & Potkonjak, M. (2003, January). Hiding data in DNA. In Information Hiding (pp. 373-386). Springer Berlin Heidelberg.