A DNA Cryptographic Technique Based on Dynamic DNA Sequence Table

Emtious Md. Sazzad Hossain*1, Kazi Md. Rokibul Alam*2, Md. Rafiul Biswas*3, and Yasuhiko Morimoto+4
*Department of Computer Science and Engineering,
*Khulna University of Engineering & Technology, Khulna-9203, Bangladesh
+Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima 739-8521, Japan
Email: emtious@gmail.com1, rokib@cse.kuet.ac.bd2, rafiulbiswas@gmail.com3, morimoto@mis.hiroshima-u.ac.jp4

Abstract—**This paper proposes a new technique for DNA cryptography that uses dynamic DNA sequence table to enhance the level of security. While handling with secure data, the requirements like compression, speed-up computation and processing etc are crucial issues. Bio-molecular DNA features possess the capability to cope up with these requirements. Existing DNA cryptographic techniques usually consider fixed DNA sequence table i.e., DNA bases and thereby the security is suspected to be breached by the intruder. To overcome this limitation, the proposed technique considers dynamic sequence table that assigns random ASCII characters to DNA sequence table initially. Then a finite number of iterations are applied based on a mathematical series where in every iteration the positions of ASCII characters are changed dynamically in the sequence table. Later on, One-Time-Pad (OTP) is applied on the modified encoding binary value. Again OTP ciphertext is processed through genomic conversion. Finally, it is converted into compressed ciphertext using amino acid table consisting of protein sequence that increases the confusion of the ciphertext. At last, the time requirements for encoding-decoding and encryption-decryption are evaluated and comparisons with other DNA techniques are presented.**

Keywords-DNA cryptography; DNA Sequence Table; One Time Pad; Biological Mechanism; Iteration Process;

I. Introduction

Unlike traditional cryptography, DNA cryptography relies on DNA characteristics along with cryptographic techniques to ensure security. Valuable properties of this technique are: self-assembling criteria of DNA molecules, parallel computations, large data storage capacity etc [9]. For example, a single gram of DNA comprises of 10^{21} DNA bases which is equal to10^8 terabytes [10]. Traditional cryptosystems are only based on mathematical techniques that weaken the robustness of their encryption process [6]. For example, recent studies show that cryptosystems like RSA, DES, AES, MD5 etc are not secure enough [5]. It is expected that by proposing unbreakable cryptosystem, DNA cryptography would ensure the data security for the next generation.

Usually existing DNA cryptographic techniques rely on static DNA sequence table [1] *i.e.* use fixed DNA sequence for each ASCII character. The traditional encryption schemes *e.g.* Polymerase Chain Reaction (PCR) amplification technology of encrypted DNA does not have enough key space (key space-constrained problem) which severely hampers the security issues [17]. Also while small sized DNA fragmentation is used, the security becomes weaker. Besides, DNA cryptography requires high computational complexity, high-tech bio molecular laboratory which are costly, complex and needs lots of time [3]. Again, encryption increasingly becomes difficult if more complex keys and too many keys are used [18].

To overcome these limitations, this paper proposes a new technique for DNA cryptography that maintains a dynamic sequence table for ASCII characters *i.e.* the DNA sequences are changed dynamically w.r.t. iterations. Here at first, ASCII characters are assigned randomly to DNA sequence table. Then the positions of DNA sequences are changed according to a mathematical series *e.g.* odd-even series. For example, initially "AAAA" is assigned for ASCII character 'a'. But namely in the next iteration, "AAAA" is assigned for ASCII character 'e'. Thus numbers of iterations increase the confusion of the ciphertext. Again, binary sequence consists of 2-bit binary value for each character of DNA sequence is processed through some biological mechanisms like mRNA, tRNA and reverse simulation. Finally amino acid table consists of protein sequences grouped according to 26 ASCII capital letters is used to get the final compressed ciphertext. Thus the two fold security *i.e.* both mathematical and biological computation disable intruders to find out the correct sequence of plaintext.

The rest of this paper is organized as follows: Section II discusses the related works. Section III describes the proposed technique. Section IV illustrates the experimental analysis and finally, Section V concludes the paper.

II. Related Works

DNA cryptography is still an emerging field of research. The technique proposed in [1] considers a biological simulation technique that is based on unique DNA encoding table for character set. It randomly generates random encoding table after every interaction session of sender and receiver. Also it creates different DNA base for each of 96 characters and thereby the same plaintext can produce different ciphertext in every session. The technique proposed in [2] uses the key features of DNA and amino acid coding to overcome the limitations of the classical One-Time-Pad (OTP) ciphertext. It also checks the randomness of value through NIST statistical test. The technique proposed in [7] deploys OTP along with substitutions and transpositions and applies double columnar transposition method on OTP ciphertext. The technique proposed in [12] is based on molecular theory, OTP and 2-dimensional image is encrypted and decrypted. The technique proposed in [16] is a new secret data writing technique based on DNA sequences and generates a DNA OTP key of

350 bits which is 70 times longer than the plaintext and perform encryption and decryption on the plaintext using symmetric key cryptography.

The technique proposed in [10] suggests a biotic DNA based secret key cryptographic mechanism based on the genetic information of biological systems. The technique proposed in [8] is a new tile-colony algorithm that can utilize the DNA hybridization process as an effective source for the random key construction. It uses a physical process like as thermal noise for generating random number.

The technique proposed in [11] is based on public key which is used to encrypt messages in DNA sequences and encrypted message sequence is forwarded to immobilization process and then for PCR amplification where message is encoded using two prime numbers. The technique proposed in [4] is designed by using the technologies of DNA synthesis, PCR amplification and DNA digital coding as well as the theory of traditional cryptography. The traditional encryption method and DNA digital coding are used to preprocess the plaintext which can effectively prevent attacks from a possible word as PCR primers. By applying the special function of primers to PCR amplification, the primers and coding mode are used as the key of the scheme. The technique proposed in [14] is based on asymmetric encryption and signature method with DNA technology. DNA public key cryptosystem is based on DNA microarray chip, fabricated with probes (used as key) for encryption and decryption. At first plaintext is converted into its ASCII code and then it is converted into binary code, binary code is arranged in the form of matrix. Key and ciphertext is biological molecule in DNA (PKC) which is transmitted physically and it is difficult to replicate.

The DNA cryptographic technique proposed in [13] is based on symmetric key where key sequences are obtained from the genetic database and remains same at both ends (sender and receiver). Here plaintext is first converted into binary format and then to DNA format using substitution. The technique proposed in [15] works on DNA hybridization, DNA digital coding and uses OTP as an encryption key, and capable to reduce time complexity. It uses parallel technique to decrypt message and also minimizes time for decryption. However, all these schemes are based on fixed DNA sequence table.

III. Proposed Technique

This paper proposes a new DNA cryptographic technique based on dynamic DNA sequence table along with OTP. Multiple steps of DNA conversions and DNA sequences increase the secrecy of ciphertext. Here, the size of private key is as long as the size of input and generated randomly. Thereby, it is very hard to breach the system *i.e.* the determination of the plaintext from its ciphertext is almost impossible.

The DNA sequence table is comprised of 24*4 size matrix similar to [1]. The table structure is rendered into dynamic through a specific number of iterations using odd-even series. The dynamic table is initiated with random character for each DNA sequence (A, T, G and C). Two-bit binary value of DNA sequence is obtained by replacing A = 00, T = 01, G = 10 and C = 11 and ASCII characters are used as input. Initially, both the sender and the receiver have the same DNA sequence table. The table is iterated through a specific number of times following the odd-even series.

TABLE I. Dynamic DNA Sequence Table of ASCII Characters Through Iterations

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

TABLE II. Amino acid Table

Symbol	**Protein sequence**
A	GCT, GCC, GCA, GCG
B	TAA,TAG
C	TGT, TGC
D	GAT, GAC
E	GAA, GAG
F	TTT, TTC
G	GGT, GGC, GGA, GGG
H	CAT, CAC
I	ATT, ATC, ATA
J	TGA
K	AAA, AAG
L	CTT, CTC, CTA, CTG
M	ATG
N	AAT, AAC
O	TTA,TTG
P	CCT, CCC, CCA, CCG
Q	CAA, CAG
R	CGT, CGC, CGA, CGG
S	TCT, TCC, TCA, TCG
T	ACT, ACC, ACA, ACG
U	AGA, AGG
V	GTT, GTC, GTA, GTG
W	TGG
X	AGT, AGC
Y	TAT
Z	TAC

While transmission, the sender notifies the iteration number to the receiver to maintain the table as dynamic along with the private key. It is dynamic because through iteration, randomly it represents characters in DNA sequence table. Each cycle of iteration changes DNA sequence for any ASCII value. That's why DNA bases are not fixed for characters until iterations end and makes the table dynamic. Table 1 is the scenario of dynamic DNA sequence table at some arbitrary point during iterations. In initial table, DNA base positions can be changed according to users' wish while maintaining the format. Biological DNA replication to RNA *i.e.* mRNA, tRNA and conversion of RNA to amino acid table to get ciphertext are used to folding up the message into multiple times.

Conversion of RNA to amino acid uses protein sequence from Table 2 where the protein sequence is replaced by a capitalized letter *i.e.* 'A' for "GCT" similar to [2]. Amino acid has multiple cordons. Therefore, a number of DNA cordons may present in the same group of protein sequence. According to the table, 26 letters are mapped to protein sequences. As the mapping process is random, it also increases the secrecy.

A. Construction of Dynamic DNA Table of ASCII characters (Through Iterations)

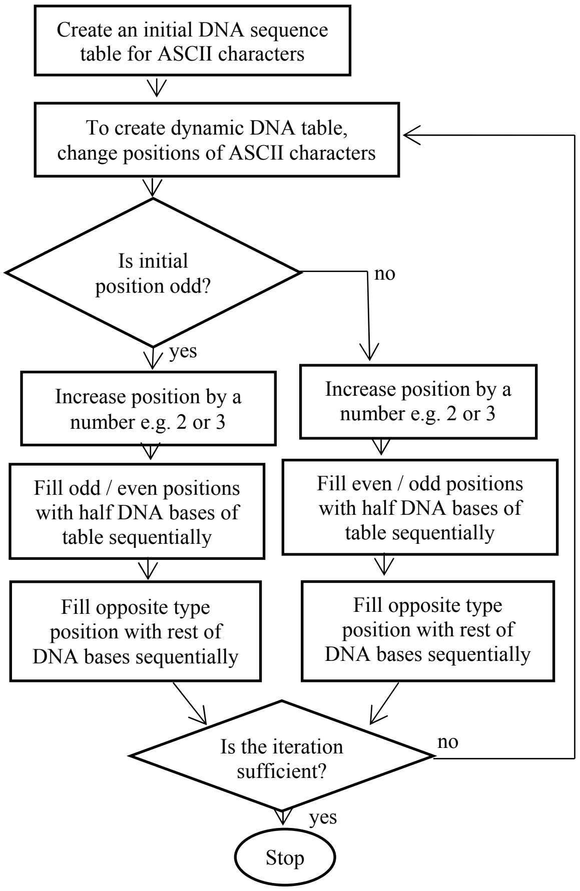
Create an initial DNA sequence table for ASCII characters
To create dynamic DNA table, change positions of ASCII characters
Is initial position odd?
no
yes
Increase position by a number e.g. 2 or 3
Increase position by a number e.g. 2 or 3
Fill odd / even positions with half DNA bases of table sequentially
Fill even / odd positions with half DNA bases of table sequentially
Fill opposite type position with rest of DNA bases sequentially
Fill opposite type position with rest of DNA bases sequentially
Is the iteration sufficient?
no
yes
Stop

Fig. 1. Iteration Process

Initial DNA sequence table is changed according to a mathematical series. During iterations, firstly initial position is checked and increased by a specific number e.g. 2/3. Then half of DNA bases of the table are filled at odd/ even positions of table sequentially according to initial position of each cycle and the rest of DNA bases are filled by opposite type position sequentially. Iteration continues until a given finite number. Iteration process creates dynamicity in the DNA sequence table through changing the DNA base position.

B. Encryption Process

Step 1: At first, an initial DNA sequence table of ASCII characters is generated and known by both the sender and the receiver. Now it is converted to dynamic DNA sequence table. By changing the DNA base positions through iterations, finally each ASCII character of the table are re-assigned that increases the security of the table. The table is destroyed after a period of time. The iteration number is sent to the receiver.

Step 2: Input plaintext is organized as the DNA sequence from the table. This is called the encoding format of input plaintext. Thus the plaintext is converted to DNA format.

Step 3: DNA sequences are converted to 2-bit binary value using the following mapping: A-00, T-01, C-10 and G-11.

Step 4: OTP key is generated which is fully random. The key size is same as the binary value of step 3.

Step 5: Binary XOR operation is applied between 2-bit binary value of input and OTP key.

Step 6: 00-A, 01-T, 10-C and 11-G mapping is applied to the XOR sequence. Thus the DNA representation of binary XOR value is obtained.

Step 7: Biological mRNA conversion process is applied on DNA sequence. The Thymine (T) is replaced with Uracil (U).

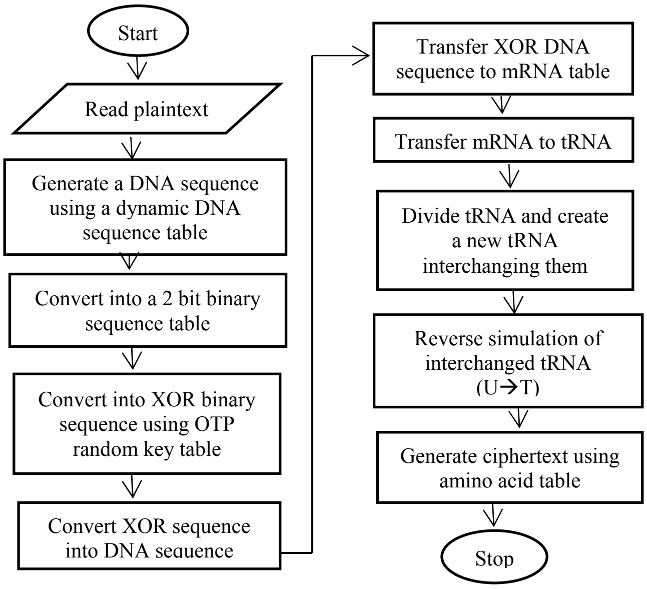
Start
Read plaintext
Generate a DNA sequence using a dynamic DNA sequence table
Convert into a 2 bit binary sequence table
Convert into XOR binary sequence using OTP random key table
Convert XOR sequence into DNA sequence
Transfer XOR DNA sequence to mRNA table
Transfer mRNA to tRNA
Divide tRNA and create a new tRNA interchanging them
Reverse simulation of interchanged tRNA (U→T)
Generate ciphertext using amino acid table
Stop

Fig. 2. The encryption process.

Step 8: mRNA is transferred to tRNA by replacing A→U, U→A, G→C and C→G. This is a real life biological conversion process.

Step 9: tRNA is divided into two parts (1st and 2nd). 1st part and 2nd part interchange their position.

Step 10: 'A' or 'AC' is added with the tRNA sequence to replace amino acid table (if sequence is not divided by 3).

Step 11: Uracil (U) is replaced by Thiamine (T). This is called the reverse simulation process and the simulated tRNA sequence is obtained.

Step 12: Then the protein sequence is obtained from amino acid table. The table is taken from [2] as a sample of protein sequence. For each letter of alphabet, a number between 0 and 25 is assigned where A = 0, B = 1,…, Z = 25. By replacing the simulated tRNA sequence by letters using this table, we get the final ciphertext.

C. Decryption Process

Step 1: Ciphertext and required iterations is received through a media. DNA encoding table and amino acid table are regenerated with the help of received information.

Step 2: Ciphertext and amino acid table are matched up. For each letter of the table the protein sequence is stored at the time of encryption. From the matching result of protein sequence and letter of ciphertext, simulated tRNA sequence of ciphertext is obtained.

Step 3: Interchanged tRNA sequence is formed by replacing Thiamine (T) with Uracil (U) from reverse simulated form.

Step 4: 'AC' or 'A' is removed from interchanged tRNA if added.

Step 5: By dividing the interchanged tRNA and changing their positions, the original tRNA is obtained.

Step 6: Then mRNA is acquired from the tRNA by replacing A→U, U→A, G→C and C→G.

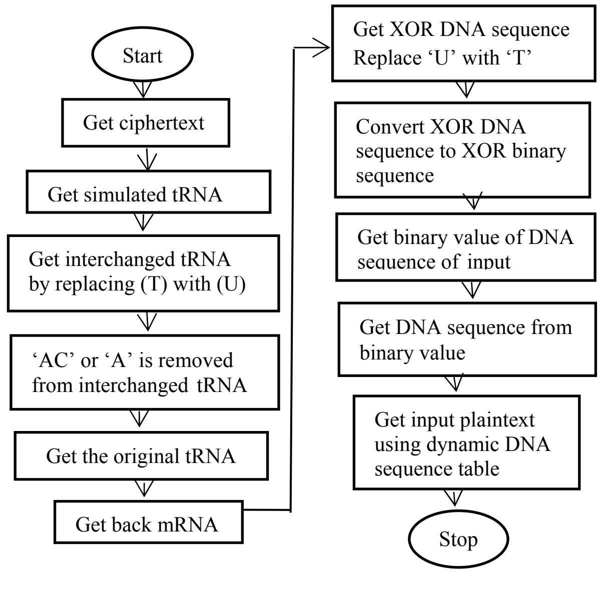
Start
Get ciphertext
Get simulated tRNA
Get interchanged tRNA by replacing (T) with (U)
'AC' or 'A' is removed from interchanged tRNA
Get the original tRNA
Get back mRNA
Get XOR DNA sequence Replace 'U' with 'T'
Convert XOR DNA sequence to XOR binary sequence
Get binary value of DNA sequence of input
Get DNA sequence from binary value
Get input plaintext using dynamic DNA sequence table
Stop

Fig. 3. Decryption process.

Step 7: mRNA is transferred to XOR DNA sequence (by replacing 'U' with 'T').

Step 8: XOR DNA sequence is converted to XOR binary sequence.

Step 9: From XOR binary sequence, random sequence is generated.

Step 10: Obtaining binary value of DNA sequence of input plaintext.

Step 11: Getting the DNA sequence of input value.

Step 12: Getting back the final input plaintext using the dynamic DNA sequence table.

IV. Experimental Analysis

A. Experimental Setup

The prototype of the proposed technique is developed under the environment on Intel(R) Core™ i5-2430M 2.50 GHz 64 bit processor with 8 GBytes of RAM running on Windows 10 operating system. It is developed in Java employing Eclipse Mars 1.0 along with jdk1.8.1 as kit where IDE default storage is used for storing data. Byte size matters calculator is used to measure the size of text.

B. Output of Encryption Steps

Step 1: Read the plaintext: "My university name is KUET".

Step 2: Plaintext is converted into DNA sequence using dynamic table of DNA sequence:

"CGAATGTAGGTCGGTGGGATCAGTGCTTGGCCGG
CGGCGCCAGTAACGTGTAGGTCGGATTCTGCGCGGG
CCGGTCCAGTGCGCGGTCCACCTGCCGATAAGGGCCC
C".

Step 3: DNA sequence is converted to 2-bit binary sequence (A = 00, T = 01, C = 10, G = 11):

"10110000011101001111011011110111111100011000110
1111001011111101011111011111011101000110100001011 0
1110100111101101111000101100111101110111111101011 1
1011010001101111011101111011010001010011110101 1000
100001111111010101 0".

Step 4: A random OTP key as long as the binary sequence of input is generated (but this is not shown herein):

Step 5: An XOR operation between binary sequence and random key is made and a binary sequence is generated:
"0001101110001111101100000100101001100111111100101
0111101000111010110011100110101111100111000010 1100
101111011011100100110010001011111110011100010010 01
111110100001010100100100010010010000101110001111 1
0001000111110011".

Step 6: XOR binary sequence is converted to DNA sequence (00 = A, 01 = T, 10 = C, 11 = G):

"ATCGCAGGCGAATACCTCTGGGCTTTGCCAGCCG
AGCTCCGGCTGAACGACGGTCGCTAGACACGGGCTG
ATACGCCATTTACTATACTAACGCAGGGATATGGAG".

Step 7: XOR DNA sequence is transferred to mRNA (By replacing 'T' with 'U'):

"AUCGCAGGCGAAUACCUCUGGGCUUUGCCAGCC
GAGCUCCGGCUGAACGACGGUCGCUAGACACGGGC
UGAUACUGGCCAUUUACUAUACUAACGCAGGGAUA
UGGAG".

Step 8: mRNA sequence is transferred to tRNA sequence (By replacing A→U, U→A, G→C, C→G):

"UAGCGUCCGCUUAUGGAGACCCGAAACGGUCGG
CUCGAGGCCGACUUGCUGCCAGCGAUCUGUGCCCGA

CUAUGACCGGUAAAUGAUAUGAUUGCGUCCCUAUACCUC”.

Step 9: Divide tRNA sequence and make new combination of tRNA sequence by interchanging first and second part:

“GCGAUCUGUGCCCGACUAUGACCGGUAAAUGAUAUGAUUGCGUCCCUAUACCUCUAGCGUCCGCUUAUGGAGACCCGAAACGGUCGGCUCGAGGCCGACUUGCUGCCA”.

Step 10: ‘AC’ or ‘A’(if required) is added with the new tRNA sequence as required for transfer the sequence according to the amino acid table in future step:

“GCGAUCUGUGCCCGACUAUGACCGGUAAAUGAUAUGAUUGCGUCCCUAUACCUCUAGCGUCCGCUUAUCCCGAAACGGUCGGCUCGAGGCCGACUUGCUGCCA”.

Step 11: Reverse simulation of interchanged tRNA sequence (U→T):
“GCGATCTGTGCCCGACTATGACCGGTAAATGATATGATTGCGTCCCTATACCTCTAGCGTCCGCTTATGGAGACCCGAAACGGTCGGCTCGAGGCCGACTTGCTGCCA”.

Step 12: Convert reversed tRNA with amino acid table into ciphertext:
“AICARLJPVNDMIASLZLBRPLMETRNGRLEADOLP”.

C. *Output of Dencryption Steps*

Step 1: To get the plaintext back, consider the ciphertext: “AICARLJPVNDMIASLZLBRPLMETRNGRLEADOLP”.

Step 2: Get the reversely simulated interchanged tRNA using the amino acid table:

“GCGATCTGTGCCCGACTATGACCGGTAAATGATATGATTGCGTCCCTATACCTCTAGCGTCCGCTTATGGAGACCCGAAACGGTCGGCTCGAGGCCGACTTGCTGCCA”

.*Step 3:* Get the interchanged tRNA sequence from the reversed simulated tRNA sequence:

“GCGAUCUGUGCCCGACUAUGACCGGUAAAUGAUAUGAUUGCGUCCCUAUACCUCUAGCGUCCGCUUAUGGAGACCCGAAACGGUCGGCUCGAGGCCGACUUGCUGCCA”.

Step 4: Remove “AC” or “A” (if added) from the interchanged tRNA sequence:

“GCGAUCUGUGCCCGACUAUGACCGGUAAAUGAUAUGAUUGCGUCCCUAUACCUCUAGCGUCCGCUUAUGCCGAAACGGUCGGCUCGAGGCCGACUUGCUGCCA”.

Step 5: Again divide the interchanged tRNA and get the original tRAN by changing their position:

“UAGCGUCCGCUUAUGGAGACCCGAAACGGUCGGCUCGAGGCCGACUUGCUGCCAGCGAUCUGUGCCCGACUAUGACCGGUAAAUGAUAUGAUUGCGUCCCUAUACCUC”.

Step 6: Get back the mRNA sequence from the tRNA sequence by (Replace A→U, U→A, G→C, C→G):

“AUCGCAGGCGAAUACCUCUGGGCUUUGCCAGCCGAGCUCCGGCUGAACGACGGUCGCUAGACACGGGCUGAUACUGGCCAUUUACUAUACUAACGCAGGGAUAUGGAG”.

Step 7: Transfer mRNA sequence to XOR DNA sequence (Replace ‘U’ with ‘T’):

“ATCGCAGGCGAATACCTCTGGGCTTTGCCAGCCGAGCTCCGGCTGAACGACGGTCGCTAGACACGGGCTGATACTGGCCATTTACTATACTAACGCAGGGATATGGAG”.

Step 8: Covert XOR DNA sequence to XOR binary sequence:
“0001101110001111101100000100101001100111111100101011110100011101011001110011010111110011100001011001011110110111001001100100010111111100111000100100111111010000101010010010001001001000010111000111111000100011111 0011”.

Step 9: Obtain random OTP key value (however this value is not shown herein):

Step 10: From random key, get the binary value of DNA sequence of input:

“10110000011101001111011011110111111100011000110111100101111110101111101111101110100011010000101101110100111101101111000101100111101110111111101011110110100011011110111011110110100010100111101011000100001111111010101 0”.

Step 11: Get the DNA sequence of input value using the dynamic DNA sequence table:

“TTGGCGTATTGGTACGGATTATCCAACGTTGGTCCCATTATTGGGGGACCTTCTGATTGGCACGCAGTTCCGGATTTGTA”.

Step 12: Get back the plaintext from the DNA sequence: “My university name is KUET”.

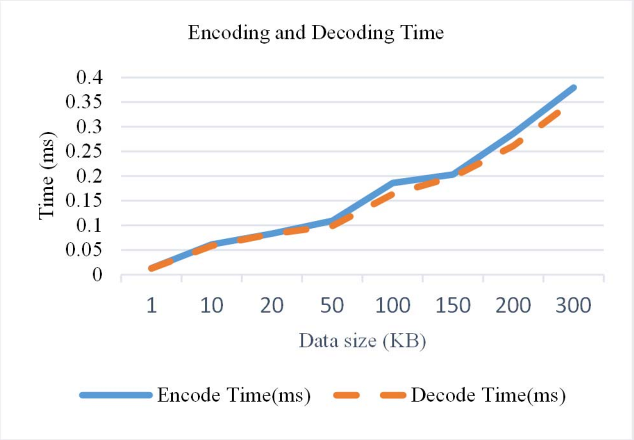

Fig. 4. Time requirement for encoding and decoding.

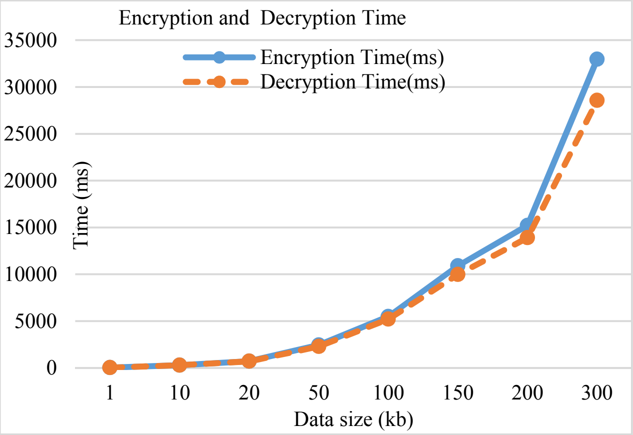

Fig. 5. Time requirement for encryption and decryption.

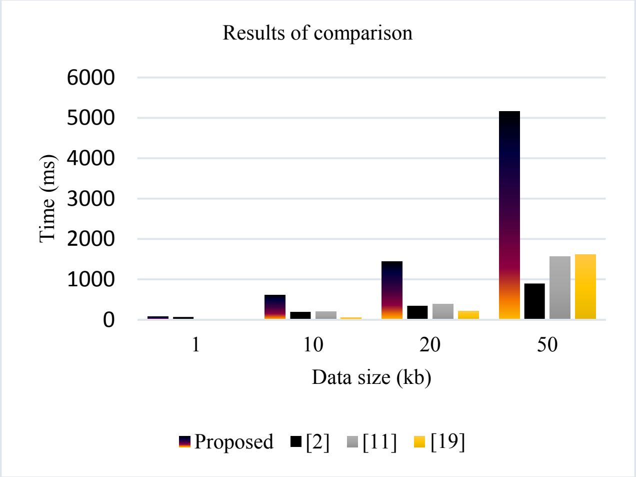

Fig. 6. Comparison with other schemes (for whole process).

D. Experimental Results and Comparisons

Employing the proposed technique, the time requirement for encoding-decoding is presented in Fig. 4 and for encryption-decryption is presented in Fig. 5. Herein, plaintexts with different lengths are chosen to show the result. Moreover, the proposed technique is compared with some other techniques proposed in [2], [11] and [19], and the result of comparison is shown in Fig. 6. Here to compute the time requirements for encoding-decoding and encryption-decryption of all the techniques, the same sized plaintext is considered as input.

E. Discussions

Fig. 5 shows that for the proposed technique, the time requirement of encryption is larger than that of decryption while the size of plaintext increases. Similarly for same sized plaintext, the time for encryption varies for different techniques. Also Fig. 6 shows that the proposed technique requires more time than other techniques for encoding-decoding and encryption-decryption. Although the time requirement for the proposed technique is higher than other techniques, it is expected that the implementation of dynamic DNA sequence table, random key with some biological concepts of the proposed technique ensures better security than other techniques.

V. Conclusions

The proposed DNA cryptographic technique based on dynamic DNA sequence table along with OTP enriches the level of data security. Here to breach the security of ciphertext, the attacker needs to perform all possible combination of checking before breaching the security of data which is expected to be quite impossible. The reason is, the decryption of data of the proposed technique relies on the use of dynamic DNA sequence table, random character for DNA sequence along with OTP in which the key is generated randomly. For this reason the proposed technique ensures better security than other related techniques, and it is powerful against certain attacks, especially against brute force attacks. The main concern of this work is to ensure security, not to reduce the time for the process. The use of dynamic DNA sequence table makes OTP more secure than ever.

References

[1] N. Hussain, U. Rahman,C. Balamuruganb, and R. Mariappan, “A Novel DNA Computing based Encryption and Decryption Algorithm,” Int. Conf. on Information and Communication Technologies (ICICT 2014), Procedia Computer Science 46 pp. 463 – 475, 2015.

[2] F. E. Ibrahim, M. I. Moussa, and H. M. Abdalkader, “A Symmetric Encryption Algorithm based on DNA Computing,” Int. Journal of Computer Applications, Vol. 97, No.16, July 2014.

[3] T. Anwar, S. Paul, and S. K. Singh, “Message Transmission Based on DNA Cryptography: Review”, Int. Journal of Bio-Science and Bio-Technology Vol.6, No.5, pp.215-222, 2014.

[4] G. Cui, L. Qin, Y. Wang, and X. Zhang. “An encryption scheme using DNA technology”. In Proc. of the 3rd Int. Conf. on Bio-Inspired Computing: Theories and Applications, USA, IEEE, pp. 37-42, 2008.

[5] K. S. Kabir, T. Chakraborty, and A.B.M. Alim Al Islam, “SuperCrypt: A Technique for Quantum Cryptography through Simultaneously Improving Both Security Level and Data Rate”, Proc. of 2016 Int. Conf. on Networking System and Security, pp. 25-33, 2016.

[6] Y. Huang, C. Chang and C. Wu, “A DNA-based data hiding technique with low modification rates”, Multimedia Tools and applications. Vol. 70, No. 3, pp. 1439-1451, 2014.

[7] S. Dhull, V. Saroha, “Enhancing Security of One Time Pad Cipher by Double Columnar Transposition Method,” Int. Journal of Advanced Research in Computer Science and Software Engineering,Vol. 3, No. 3, 2013.

[8] M. Hirabayashi, H. Kojima and K. Oiwa, “Design of True Random One-Time Pads in DNA XOR Cryptosystem”, F. Peper et al. (Eds.): IWNC 2009, PICT 2, pp. 174-183, Springer 2010.

[9] O. Tornea, “Contributions to DNA cryptography: applications to text and image secure transmission”, Diss. Université Nice Sophia Antipolis; Technical University of Cluj-Napoca (Roumanie), 2013.

[10] E. S. Babu, C. N. Raju, and M. HM K. Prasad, “Inspired Pseudo Biotic DNA Based Cryptographic Mechanism Against Adaptive Cryptographic Attacks”, Int. Journal of Network Security, Vol.18, No.2, pp.291-303, 2016.

[11] T. Kazuo, O. Akimitsu, and S. Isao, “Public-key system using DNA as a one-way function for key distribution,” BioSystems, Elsevier Science, Vol. 81, No. 1, pp. 25–29, 2005.

[12] C. Jie, “A DNA-based bio molecular cryptography design,” Proc. of IEEE International Symposium, Vol. 3, pp. III-822, 2003.

[13] S. T. Amin, M. Saeb and E. Salah, “A DNA-based implementation of YAEA encryption algorithm”, In Computational Intelligence, pp. 120-125, 2006.

[14] X. Lai, M. Lu, L. Qin, J. Han, and X. Fang, “Asymmetric encryption and signature method with DNA technology”, Science China Information Sciences, Vol. 53, No. 3, pp. 506-514, 2010.

[15] P. Sabari and K. S. Setua, “DNA cryptography”, In 7th IEEE Int. Conf. on Electrical and Computer Engineering (ICECE), , pp. 551-554, 2012.

[16] D. Kumar, and S. Singh, “Secret data writing using DNA sequences,” In IEEE Int. Conf. on Emerging Trends in Networks and Computer Communications (ETNCC), pp. 402-405, 2011.

[17] Y. Zhang, L. H. B. Fu, “Research on DNA Cryptography,” (Chapter 16), Dr. Jaydip Sen, Applied Cryptography and Network Security (March, 2012), InTech.

[18] R. Mor, and P. Kanth, “A Review Paper Of DNA Based Cryptographic,” National Conf. on Innovative Trends in Computer Science Engineering (ITCSE-2015), IJRRA ISSN: 2349-7688.

[19] A. Atito, A. Khalifa and S. Z. Reda, ”DNA-Based Data Encryption and Hiding Using Playfair and Insertion Techniques”, Journal of Communications and Computer Engineering, Volume 2, Issue 3, pages 44: 49, 2012.