# A DNA Cryptographic Technique Based on Dynamic DNA Encoding and Asymmetric Cryptosystem

Md. Rafiul Biswas[*1], Kazi Md. Rokibul Alam[*2], Ali Akber[*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: rafiulbiswas@gmail.com[1], rokib@cse.kuet.ac.bd[2], mail2aliakber@gmail.com[3], morimoto@mis.hiroshima-u.ac.jp[4]

***Abstract*—This paper proposes a new DNA cryptographic technique based on dynamic DNA encoding and asymmetric cryptosystem to increase the level of secrecy of data. The key idea is: to split the plaintext into fixed sized chunks, to encrypt each chunk using asymmetric cryptosystem and finally to merge the ciphertext of each chunk using dynamic DNA encoding. To generate chunks, characters of the plaintext are transformed into their equivalent ASCII values and split it into finite values. Now to encrypt each chunk, asymmetric cryptosystem is applied and the ciphertext is transformed into its equivalent binary value. Then this binary value is converted into DNA bases. Finally to merge each chunk, sufficient random strings are generated. Here to settle the required number of random strings, dynamic DNA encoding is exploited which is generated using Fibonacci series. Thus the use of finite chunks, asymmetric cryptosystem, random strings and dynamic DNA encoding increases the level of security of data. To evaluate the encryption-decryption time requirement, an empirical analysis is performed employing RSA, ElGamal and Paillier cryptosystems. The proposed technique is suitable for any use of cryptography.**

***Keywords—DNA cryptography; RSA; ElGamal; Paillier; DNA bases; Dynamic DNA encoding;***

## I. Introduction

DNA cryptography is a promising technique to ensure the secrecy of data. It is more advantageous than traditional cryptography because it provides two fold securities whereas the traditional cryptography provides only one fold security. To provide two fold securities, DNA cryptography combines computational difficulties as well as biological difficulties [1] where the first fold is ensured by an existing cryptosystem and the second fold is provided by the features of DNA characteristics. The binding capabilities of DNA bases (A-T, C-G) offer the opportunity of creating self-assembly structures that are an excellent means of executing computations and ensuring secrecy [2]. Thus DNA bases are used as the information carrier to make the transmitted data more secure over the public communication channel.

To develop DNA cryptographic technique, symmetric cryptosystem like DES, triple- DES, AES, one time pad (OTP) etc are usually used. However exploiting asymmetric cryptosystem e.g. RSA, ElGamal, Paillier; DNA cryptographic technique also has been proposed recently. While symmetric cryptosystem is used, the sender and the receiver use the same key for both encryption and decryption. Thereby the secure distribution of the key between the involved parties becomes complicated. Intuitively, the requirement of storage as well as computation is also increased herein. In contrast, asymmetric cryptosystem provides numerous advantages than symmetric cryptosystem i.e. increased security, convenience, public key is revealed from the beginning, no need to disclose the private key etc. Thereby while DNA cryptographic technique is developed using asymmetric cryptosystem along with related efficient technique of mathematics, it can ensure increased security and decreased computation.

The goal of this paper is to ensure more security over traditional asymmetric cryptosystems like RSA, ElGamal, Paillier etc using the concept of DNA. Here, it is not intended to utilize the real DNA to perform the cryptographic process; rather, a new DNA cryptographic technique based on dynamic DNA encoding has been proposed. The technique splits the plaintext into chunks, encrypts each chunk using an asymmetric cryptosystem and transforms the ciphertext into binary value to convert it into DNA bases. Then it generates sufficient random strings and Fibonacci series to develop dynamic DNA encoding which settles the required number of random strings to merge the chunks altogether. Thus the multi fold security makes the technique more efficient and can be applied in any use of data security.

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

## II. Related Works

DNA cryptography [9] is comparatively a new direction of research in the domain information security. Recently numerous DNA cryptographic techniques have been proposed to settle the issue. The technique proposed in [3] is based on DNA OTP encryption algorithm where the technique is exploited to encrypt a huge number of short messages. It also proposed another DNA steganographic technique to explore the two-dimensional image encryption technique of DNA on a chip technology which is capable to hide unrelated DNA strands. The disadvantages of this technique are, it is difficult to prepare a huge DNA OTP in which data can be easily separated and read out. The involved parties i.e. the sender and the receiver are bound to perform complex biological experiments, which can only be done in a well equipped lab and it is highly expensive to achieve. Therefore within many years, the technique is not feasible for the above reasons [12].

978-1-5386-3288-8/17/$31.00 ©2017 IEEE

The technique proposed in [4] develops two DNA cryptographic techniques: one is based on symmetric cryptosystem and another is based on asymmetric cryptosystem. Thus it presents a comparison between them. Here to implement the symmetric based one, OTP encryption technique is used; and to evaluate its performance Java language, BioJava and MatLab platforms are employed. To develop the asymmetric based one, password phrase is used that strengthens the secrecy of the technique. Here the drawn conclusion shows that the asymmetric based technique is more consistent and robust than the symmetric based technique.

The technique proposed in [5] is based on symmetric cryptosystem. But it uses modified RSA encryption function to encrypt the key prior to sending it to the receiver to be used for the purpose of decryption. The sender transmits the encrypted data along with the encrypted key to the receiver. However for encryption and decryption operations, the comparison presented herein shows that it requires much more time than DES, triple DES, AES, IDEA etc.

The technique proposed in [6] is based on symmetric cryptosystem where both encryption and decryption keys are generated by DNA probes. Here encryption and decryption is performed through DNA fabrication and DNA hybridization respectively, and the ciphertext is embedded in the DNA chip. It is based on the concept that most difficult DNA microarray technology is used to attain information security in a cryptosystem same as [13]. The technique proposed in [7] uses the same DNA chip technology where digital signature is exploited as asymmetric cryptosystem to authenticate the sender and the receiver.

The technique presented in [8] proposes a symmetric key block cipher inspired from DNA which consists of three phases. These are: the initial, the iteration and the final phases. It includes a step that simulates the idea of original biological molecules of transcription processes i.e. transfers from DNA to mRNA. Then it translates from mRNA to amino acids. While designing, it follows recommendations of experts in cryptography and focuses on ‘confusion’ and ‘diffusion’ which are essential properties of ciphertext.

The technique proposed in [10] is based on DNA and RSA cryptosystem, capable to provide an architectural framework for encryption and generation of digital signature for all characters, simple text data, and text files. Here the whole process consists of four different steps. These are: the key generation, the data pre/post processing, the DNA and the signature generation.

The technique proposed in [11] considers the concept of dynamic DNA sequence table that assigns random ASCII characters to DNA sequence table initially. Then it applies a finite number of iterations to change the positions of ASCII characters dynamically in the sequence table based on a mathematical series. However herein, the use of OTP sacrifices the efficiency of the technique because OTP plaintext and key must be equal in size and thereby the secure transmission of the key is little bit difficult.

The technique proposed in [14] uses DNA cryptographic technique to design secure cloud storage and data distribution platform among clients and server from client devices. Here DNA cryptographic technique is used because for public cloud it ensures light weight and efficient communication with simplicity and low cost implementation. Besides, it provides enhanced security for data and file transmission.

Using DNA bases the technique proposed in [15] presents a module of cryptosystem which is partially reconfigurable in run time. It consists of two modules where the first one generates the dynamic key and the second one performs inverse permutation block in DES and shift rows block in AES for maintaining data security. Also to enhance the level of security, it applies DNA bases on both ciphertext and key in protein form while transmitting over the communication channel.

Analyzing existing techniques it has been observed that numerous DNA cryptographic techniques extensively deploy DNA bases along with symmetric cryptosystems. Although few techniques are available that exploit asymmetric cryptosystem with DNA bases, rarely they deploy additional mechanisms. In contrast to ensure better security, the proposed technique at first splits up the plaintext into several fixed sized chunks, then encrypts each chunk using an asymmetric cryptosystem, and finally generates random strings of DNA bases and a smart number series to form dynamic DNA encoding which merges the ciphertext of each chunk. Thereby it is applicable in any application of cryptography i.e. to ensure the secrecy of ASCII character, text, data, image etc.

## III. The Proposed Technique

The proposed technique mainly consists of three stages. These are: the formation of dynamic DNA encoding, the encryption and the decryption stages. They are described below.

### A. *Dynamic DNA Encoding*

The objective of dynamic DNA encoding is to increase the level of secrecy of ciphertext. For this purpose, the ciphertext of each chunk is merged using it. The value of DNA encoding which is used to merge any two ciphertext of chunks changes dynamically due to the use of Fibonacci series and random strings. This formation procedure is as follows. Fig. 1 depicts its block diagram.

*Step* 1: At first sufficient number of random binary strings are generated.

*Step* 2: Random binary strings are converted into DNA base as 00 = A, 01 = C, 10 = G and 11 = T.

*Step* 3: Now the ciphertext of each chunk is merged using Fibonacci series where the series is: 1, 1, 2, 3, 5, 8, 13….. , From this series for example, values from the 4th positions are used to pick DNA bases. Thus for the value 3, first three DNA bases are used to merge the ciphertext of chunk 1 and the ciphertext of chunk 2.

*Step* 4: Here if Fibonacci series is used in ascending order, while the number of chunk increases the required number of DNA bases also increases.

Instead of 4th position while another position is selected from the Fibonacci series, DNA bases used for merging the ciphertext of chunks are also changed. Thus for different position of Fibonacci series, dynamic DNA encoding produces different DNA bases.

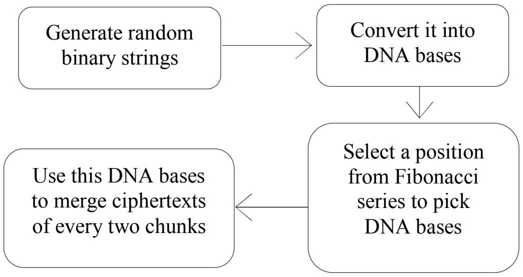

Generate random binary strings
Convert it into DNA bases
Select a position from Fibonacci series to pick DNA bases
Use this DNA bases to merge ciphertexts of every two chunks

Fig. 1. Block Diagram of Formation of Dynamic DNA Encoding.

## B. Encryption Stage

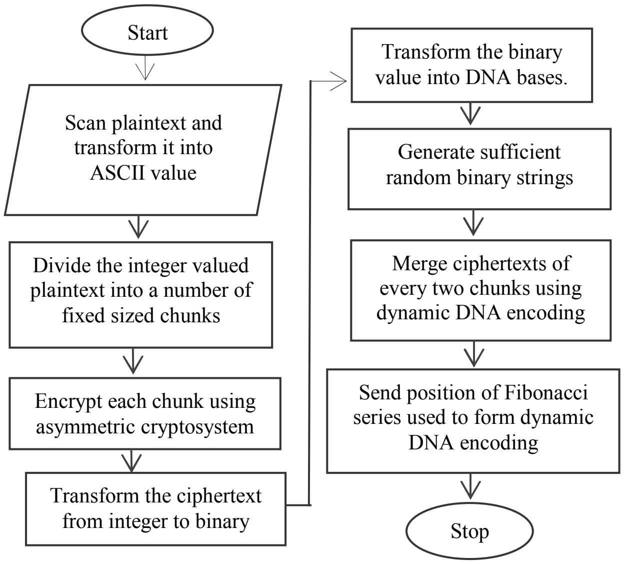

Start
Scan plaintext and transform it into ASCII value
Divide the integer valued plaintext into a number of fixed sized chunks
Encrypt each chunk using asymmetric cryptosystem
Transform the ciphertext from integer to binary
Transform the binary value into DNA bases.
Generate sufficient random binary strings
Merge ciphertexts of every two chunks using dynamic DNA encoding
Send position of Fibonacci series used to form dynamic DNA encoding
Stop

Fig. 2. Encryption Stage

Initially the receiver generates the keys of the exploited asymmetric cryptosystem i.e. it publicly announces the public encryption key to be used by any sender and securely keeps the private decryption key to itself. Now the encryption process conducted by the sender is as follows.

*Step* 1: At first scan the plaintext and transform each character of it into the corresponding ASCII value consisting of 3 digits to turn into integer value.

*Step* 2: Divide this integer valued plaintext into a number of fixed sized chunks.

*Step* 3: Now encrypt each chunk using the public key of the asymmetric cryptosystem to convert it into the ciphertext (it is also integer value).

*Step* 4: Transform the ciphertext from integer to binary.

*Step* 5: Now transform the binary value into DNA bases through substitution as 00 = A, 01 = C, 10 = G and 11 = T.

*Step* 6: Generate sufficient sized random binary strings to form dynamic DNA encoding.

*Step* 7: Use dynamic DNA encoding, to merge the ciphertext of every two chunks to generate the final ciphertext.

*Step* 8: Send the position of the Fibonacci series chosen to form the dynamic DNA encoding. While sending, an asymmetric cryptosystem can also be used to encrypt it.

## C. Decryption Stage

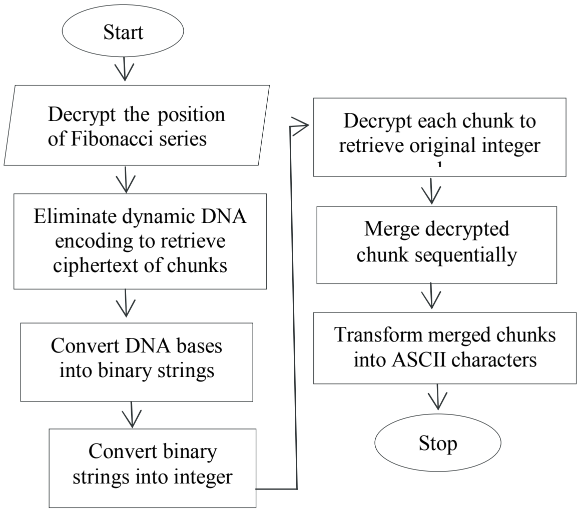

Start
Decrypt the position of Fibonacci series
Eliminate dynamic DNA encoding to retrieve ciphertext of chunks
Convert DNA bases into binary strings
Convert binary strings into integer
Decrypt each chunk to retrieve original integer
Merge decrypted chunk sequentially
Transform merged chunks into ASCII characters
Stop

Fig. 3. Decryption Stage

*Step* 1: Decrypt the position of Fibonacci series.

*Step* 2: Eliminate the portion of dynamic DNA encoding to retrieve the original ciphertext of chunks. While a chunk e.g. the first one is encrypted using *n* bit asymmetric cryptosystem, its ciphertext contains *n* bits binary i.e. (*n* / 2) DNA bases and it is stored as the ciphertext of chunk 1. Then the next DNA bases are eliminated according to the position of Fibonacci series. Again next (*n* / 2) DNA bases are stored as the ciphertext of chunk 2. This process continues up to the end of the last chunk.

*Step* 3: Convert the DNA bases into binary strings through reverse substitution as A = 00, C = 01, G = 10 and T= 11.

*Step* 4: Convert the binary strings into its corresponding integer value which is the ciphertext of chunks.

*Step* 5: Apply the private key of the asymmetric cryptosystem to decrypt each chunk to retrieve the ASCII value of plaintext.

*Step* 6: Merge the decrypted chunks sequentially.

*Step* 7: Transform merged chunks into their corresponding ASCII characters. Thus the original plaintext is retrieved.

## IV. Experimental Analysis

### A. *Experimental Setup*

The prototype of the proposed technique has been developed under the environment of Intel(R) CoreTM i5-2430M 2.50 GHz 64 bit processor with 8 GBytes of RAM running on Windows 8.1 operating system. It has been developed on Integrated Development Environment Code Blocks 10.05 and C and C++ is used as the primary language to implement the storage structure. Besides to improve the performance, GMP (The GNU Multiple Precision Arithmetic Library) and STL (Standard Template Library) are used. Also in every encryption and decryption operations of exploited asymmetric cryptosystems, 1024 bit key is used. Moreover for experiment, three asymmetric cryptosystems i.e. RSA, ElGamal and Paillier have been chosen.

### B. *Example of Dynamic DNA Encoding*

*Step* 1: Generate sufficient random binary strings as: 01001110100101110111101

*Step* 2: Convert it into DNA bases as: CACTGCATGATTC

*Step* 3: Generate Fibonacci series: 1, 1, 2, **3, 5, 8, 13……..** Use it from the 4th position.

*Step* 4: To merge the ciphertext of chunk 1 and chunk 2, use the value of 4th position of Fibonacci series i.e. three (3). Use it to pick first three DNA bases i.e. CAC. Similarly to merge chunk 2 and chunk 3, select the value of 5th position of Fibonacci series i.e. five (5). Use it to pick next five DNA bases i.e. TGCAT. Thus dynamic DNA encoding consecutively merges the ciphertext of chunks.

### C. *Output of Encryption Stage*

Considering the plaintext 'Go to room 101, KUET', this section presents the output for 1024 bit RSA. Also it is assumed that the receiver has already generated and announced the public encryption and the private decryption keys of the used asymmetric cryptosystem.

*Step* 1: Convert each character of the plaintext into its corresponding ASCII value (3 digits) which is shown in Table I. For example: "KUET" → 075**085**069**084**

K **U** E **T**

*Step* 2: Divide the integer text into a number of equal sized chunks which is shown in Table II. To make the chunk size equal, '0' is added in front of the integer text. Here, chunk size = 33 (for 1024 bit key one can choose chunk ~ 300). The integer text of this example is as follows.

**000000**07111103211611103211411111110903204904804 9044032075085069084

*Step 3:* Encrypt each chunk using 1024 bit RSA which is shown in Table III where only a portion of the ciphertext is shown.

*Step* 4: Convert the ciphertext from integer to binary which is shown in Table IV.

*Step* 5: Convert the binary strings into DNA bases as 00 = A, 01 = C, 10 = G and 11 = T which is shown in Table V. For example binary strings '01**11**01**00**' is replaced as 'C**T**C**A**'.

TABLE I. ASCII Value of Plaintext

| **Plaintext** | **Corresponding ASCII Value** |
|---|---|
| Go to room 101, KUET | 071**111**032**116**111**032**114**111**111**109**032**049**048**04**<br>**9**044**032**075**085**069**084** |

TABLE II. Plaintext Divided into Chunks

| | |
|---|---|
| **Plaintext of Chunk 1** | 000000071111032116111032114111111 |
| **Plaintext of Chunk 2** | 109032049048049044032075085069084 |

TABLE III. Ciphertext of Chunks (a portion )

| | |
|---|---|
| **Ciphertext of Chunk 1** | 798257557215921646025060201764056662071543<br>071010670673957392511404263169248041123785<br>464591185039889378114326231536149156159504 |
| **Ciphertext of Chunk 2** | 360184208643488414527031614724039693366771<br>304654892673399239845694660659954605240542<br>871945889813251215289217134235296125430515 |

TABLE IV. Binary Form of Ciphertext (A portion)

| | |
|---|---|
| **Binary strings of Chunk 1** | 001101111011110010110001100011100000000100<br>111110011000000011110101011111111110101110<br>001111100111000110101001101001110011111101<br>111111001010001110101101110101001111101110<br>1010 |
| **Binary strings of Chunk 2** | 100110111001001000001000110100110001011010<br>001011110110001101110010011110111100000010<br>111010101100111101000010010110100001001110<br>000111111101000111100010111011100110101010<br>1011 |

TABLE V. DNA Bases form of Binary Strings (A portion)

| | |
|---|---|
| **DNA bases of Chunk 1** | ATCTGTTAGTACGATGAAACATTGCGAAATT<br>CCCTTTTGGTGATTGCTACGGGCGGCTATTT<br>CT |
| **DNA bases of Chunk 2** | ATATCAGGGTTAAGTCTAGCTTTCTTCCACA<br>CATGAGCTACCCAGGGCCCGGCCGATGTAG<br>TC |

*Step* 6: Exploit dynamic DNA encoding to merge DNA bases of chunks to generate the final ciphertext which is shown in Fig. 4. The receiver has to decrypt it to retrieve the original plaintext. In the figure, the bold characters are appearing for the role of dynamic DNA encoding.

ATGAGATGGTGCAGAGTTGAGAGGCTTGGCAGGAACT<br>GGTTAAGATCAACCCACTCGATGCCTA**CAG**ATATCAGG<br>GTTAAGTCTAGCTTTCTTCCACACATGAGCTACCCAGGT<br>TGGATTACCTCTATCCCCCGGGGCATCGATGAGGTGCC<br>AGCTACCATACTGGCGC**AATGT**

Fig. 4. Final Ciphertext.

*Step* 7*:* Encrypt the position of Fibonacci series which is used in dynamic DNA encoding. As the position 4th is used here, this is (i.e. 4) encrypted to be sent to the receiver.

## *D. Output of Decryption Stage*

*Step* 1*:* Decrypt the position of Fibonacci series. As the 4th position was used, the decrypted output will be four (4).

*Step* 2*:* To divide the ciphertext into chunks remove the DNA bases for the portion of dynamic DNA encoding. As 1024 bit (RSA) is used, the chunk size is 1024 / 2 = 512. A portion of the ciphertext in the form of DNA bases is shown in Table VI.

TABLE VI. CIPHERTEXT OF CHUNKS IN THE FORM OF DNA BASES (A PORTION)

| | |
|---|---|
| **DNA bases of Chunk 1** | ATCTGTTAGTACGATGAAACATTGCGAAATTCCCTTTTGGTGATTGCTACGGGCGGCTATTTCT |
| **DNA bases of Chunk 2** | ATATCAGGGTTAAGTCTAGCTTTCTTCCACACATGAGCTACCCAGGGCCCGGCCGATGTAGTC |

*Step* 3*:* Convert DNA bases into binary strings as A = 00, C = 01, G = 10 and T= 11. Here the 512 bit DNA bases are converted into 1024 bit binary. A portion of the ciphertext in the form of binary strings is shown on Table VII.

TABLE VII. CIPHERTEXT OF CHUNKS IN THE FORM OF BINARY STRINGS (A PORTION)

| | |
|---|---|
| **Binary strings of Chunk 1** | 0011011110111100101100011000111000000001001111100110000000111101010111111111110101110001111100111000110101001101001110011111101111111001010001110101101110101001111101110 1010 |
| **Binary strings of Chunk 2** | 1001101110010010000010001101001100010110100010111101100011011100100111101111000000101110101011001111010000100101101000010011100001111111010001111000101110111001101010101011 |

*Step* 4: Convert binary strings of chunks into its corresponding integer value. A portion of the ciphertext in the form of integer is shown on Table VIII.

TABLE VIII. CIPHERTEXT OF CHUNKS IN THE FORM OF INTEGER (A PORTION)

| | |
|---|---|
| **Ciphertext of Chunk 1** | 798257557215921646025060201764056662071543071010670673957392511404263169248041123785464591185039889378114326231536149156159504 |
| **Ciphertext of Chunk 2** | 360184208643488414527031614724039693366771304654892673399239845694660659954605240542871945889813251215289217134235296125430515 |

*Step* 5: Now apply RSA decryption technique over each chunk to retrieve the integer representation of the plaintext which is shown in Table IX. As already mentioned that at this step, each chunk should be equal sized ('0' is added in front of integer to make chunks equal sized, same as the encryption stage).

TABLE IX. PLAINTEXT OF CHUNKS IN INTEGER FORM

| | |
|---|---|
| **Plaintext of Chunk 1** | 000000071111032116111032114111111 |
| **Plaintext of Chunk 2** | 109032049048049044032075085069084 |

*Step 6*: Merge the decrypted plaintext of chunks sequentially. Thereby, the retrieved integer plaintext is:

**000000**07111103211611103211411111110903204904804 9044032075085069084.

*Step* 7: Finally convert every three digits of integer to its corresponding ASCII character. While "000" is counted, nothing will be replaced. Thus the finally retrieved plaintext is shown in Table X.

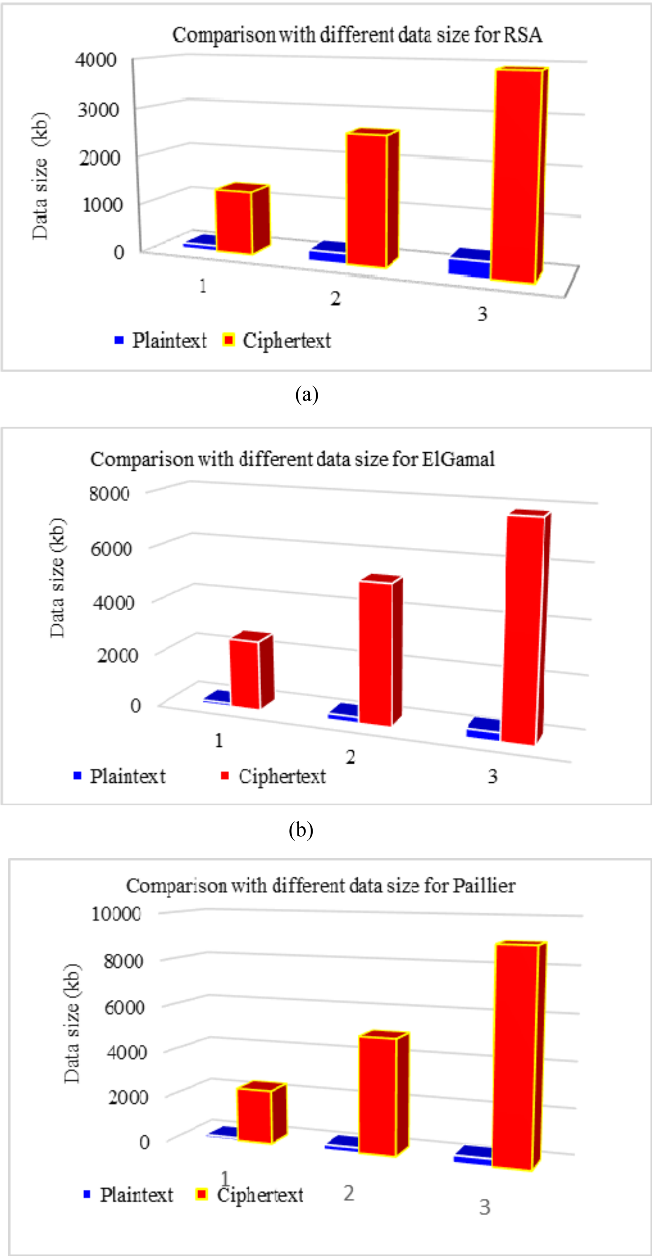
TABLE X. PLAINTEXT FINALLY RETRIEVED

| **Final Integer Text** | **Original Plaintext** |
|---|---|
| 071**111**032**116**111**032**114**111**111**109** 032**049**048**049**044**032**075**085**069**084** | Go to room 101, KUET |

## *E. Experimental Results, Comparisons and Discussions*

The proposed technique has been implemented with different data size while employing RSA, ElGamal and Paillier cryptosystems. Here the size of the plaintext (i.e. input data) is chosen as fixed i.e. 109.632, 219.264 and 328.896 kb for all the cryptosystems. Then the generated output i.e. the size of the ciphertext is shown in Fig. 5. Here Fig. 5 (a), Fig. 5 (b) and Fig. 5 (c) shows the result for RSA, ElGamal and Paillier cryptosystems respectively. From the figure for all the data size it has been observed that for RSA the size of the ciphertext is almost 12 times greater than the size of the plaintext. Whereas for ElGamal the size of the ciphertext is almost 24 times greater than the size of the plaintext. Besides for Paillier, (on average) the size of the ciphertext is almost 16 times greater than the size of the plaintext.

From the figures it has been observed that for all three different data sets and different cryptosystems, the size of the ciphertext becomes sufficiently larger than the size of the plaintext. Actually this is usual for cryptographic operations especially while an asymmetric cryptosystem is exploited. The underlying reason is, the exponentiation operation (s) involved in the encryption stage of the asymmetric cryptosystem makes the data size of the ciphertext so bigger. Besides, the proposed technique focuses on security rather than space. Here, DNA bases from random binary strings are used to merge the ciphertexts of chunks to scramble the plaintext. Thereby the proposed technique ensures more security than the traditional asymmetric cryptosystem.



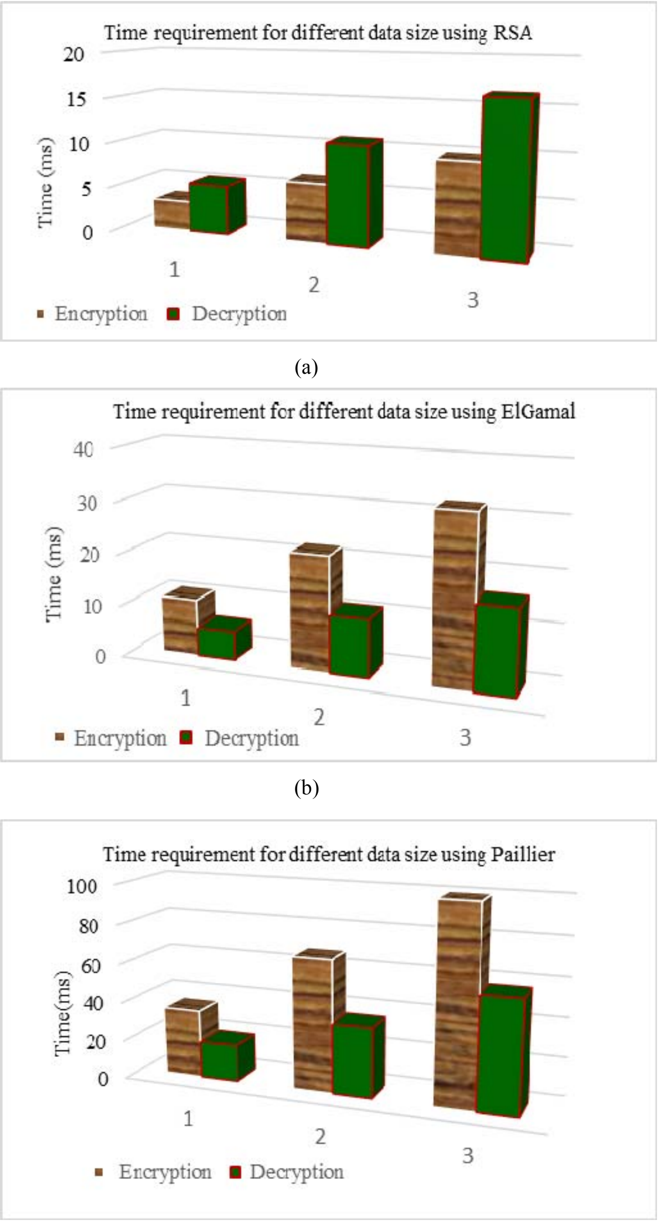
(a)

(b)

(c)

Fig. 5. Comparison between plaintext and ciphertext with different data size: (a) RSA; (b) ElGamal; (c) Paillier.

In order to depict a comparison in case of time requirement for encryption and decryption operations among RSA, ElGamal and Paillier cryptosystems, an experiment has been conducted employing the same plaintext (input data) i.e. 109.632, 219.264 and 328.896 kb that is also already used. Then the output is shown in Fig. 6. Here, Fig. 6 (a), Fig. 6 (b) and Fig. 6 (c) shows the time requirement of RSA, ElGamal and Paillier respectively. From Fig. 6 (a) it has been observed that the decryption time for RSA is greater than the encryption time. The reason is that the value of the decryption key selected herein is sufficiently larger than the value of the encryption key. Whereas from Fig. 6 (b) it has been observed that the time requirement of encryption for ElGamal is larger than the time requirement of decryption. The reason is ElGamal encryption stage performs two modular exponentiation operations. Similarly from Fig. 6 (c) it has been observed that the time requirement of encryption for Paillier is also larger than the time requirement of decryption.



(a)

(b)

(c)

Fig. 6. Comparison between encryption and decryption time with different data size: (a) RSA; (b) ElGamal; (c) Paillier.

Fig. 7 shows the comparison of encryption time requirement among RSA, ElGamal and Paillier. Fig. 8 shows the comparison of decryption time requirement among RSA, ElGamal and Paillier. In both cases, the same plaintext (input data) i.e. 109.632, 219.264 and 328.896 kb is used to conduct the experiment.

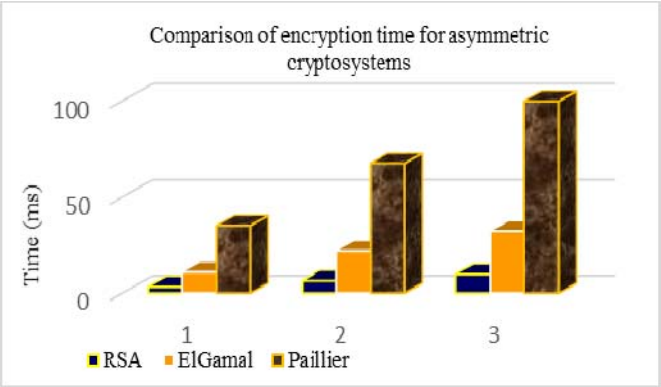


Fig. 7. Comparison among RSA, ElGamal and Paillier with different data size with respect to encryption time.

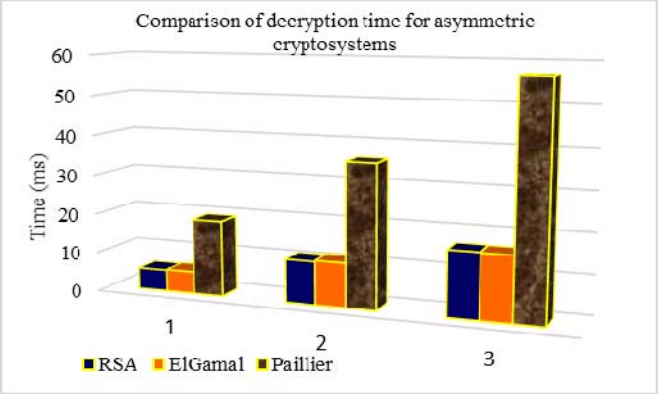


Fig. 8. Comparison among RSA, ElGamal and Paillier with different data size with respect to decryption time.

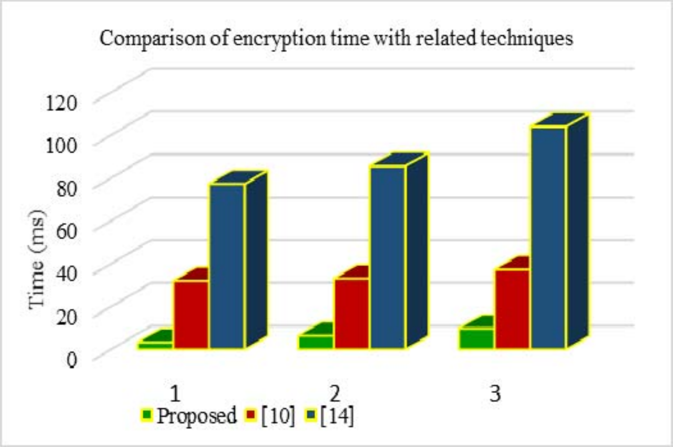


Fig. 9. Comparison of encryption time with other techniques.

In case of time requirement of encryption and decryption, the proposed technique has been compared with some other related techniques proposed in [10] and [14]. Here the size of the plaintext is also as same as the previous experiments. The comparison of time in case of encryption has been presented in Fig. 9 whereas in case of decryption it has been presented in Fig. 10. The figures prove that for plaintext of any size, the time requirement of encryption along with decryption of the proposed technique is always smaller than other techniques. Although not presented in the figures, in the same aspects the time requirement of the technique proposed in [11] is also extensively larger than the proposed technique. Thus while maintaining quite security the proposed technique performs faster than the compared techniques.

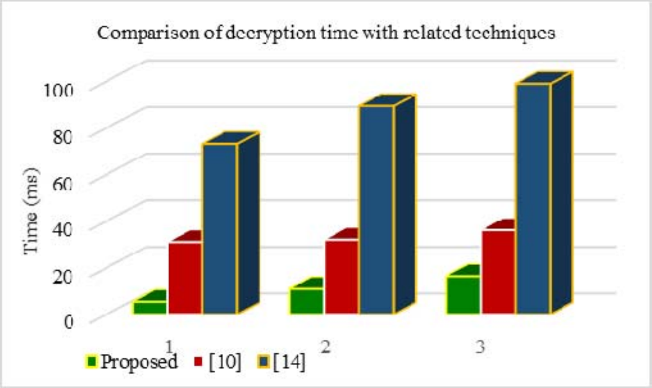


Fig. 10. Comparison of decryption time with other techniques.

## V. CONCLUSION

The proposed DNA cryptographic technique introduces and exploits dynamic DNA encoding, a new concept that enriches the level of secrecy of the ciphertext extensively. The dynamic DNA encoding consists of random strings and Fibonacci series, is used to merge the ciphertext of every two chunks of the plaintext. Here the DNA bases used to merge the ciphertext changes dynamically due to the use of Fibonacci series. Thus the proposed technique enhances the strength of traditional asymmetric cryptosystems. Hence it is almost impossible for any intruder or attacker to breach the secrecy of the data i.e. to mount any form of attack on the ciphertext. The proposed technique splits up the plaintext, employs an asymmetric cryptosystem to encrypt it and uses dynamic DNA encoding to merge the ciphertext of split data. Thereby the ultimate ciphertext generated by the proposed technique becomes highly robust and secure. A future plan of improvement is to employ the technique to ensure the secrecy of various forms of data (e.g. image, text, audio, video etc) in the cloud storage and data distribution platform.

## REFERENCES

[1]. B. Anam, K. Sakib, M. Hossain and K. Dahal, "Review on the Advancements of DNA Cryptography," arXiv preprint arXiv: 1010.0186 (2010).

[2]. O. Tornea, and M. E. Borda, "DNA cryptographic algorithms," Int. Conf. on Advancements of Medicine and Health Care through Technology. pp. 223-226, Springer, Berlin, Heidelberg, 2009.

[3]. A. Gehani, T. LaBean, and J. Reif, "DNA-based cryptography," Lecture notes in computer science 2950, pp. 167-188, 2003.

[4]. R. Terec, M.F. Vaida, L. Alboaie and L. Chiorean, "DNA security using symmetric and asymmetric cryptography," Int. Journal of New Computer Architectures and Their Applications (IJNCAA) 1.1, pp. 34-51, 2011.

[5]. S.Tripathi, P. Shiv, M. Jaiswal, and V. Singh, “Securing DNA Information through Public Key Cryptography,” MIS Review 19.1, pp. 45-59, 2013.

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

[7]. X. Lai, M. Lu, L. Qin, J. Han and X. Fang, “Asymmetric encryption and signature method with DNA technology,” Science China Information Sciences 53(3), pp. 506-514. 2010

[8]. S. Sadeg, M. Gougache, N. Mansouri and H. Drias, “An encryption algorithm inspired from DNA,” Int. Conf. on Machine and Web Intelligence (ICMWI), pp. 344-349, IEEE, 2010.

[9]. Adleman and M. Leonard, “Molecular computation of solutions to combinatorial problems,” Nature 369, p. 40, 1994.

[10]. D. S. Chouhan, and R. P. Mahajan, “An architectural framework for encryption & generation of digital signature using DNA cryptography,” Int. Conf. on Computing for Sustainable Global Development (INDIACom), pp. 743-748, IEEE, 2014.

[11]. E. M. S. Hossain, A. K. Md. Rokibul, M. R. Biswas, and Y. Morimoto, “A DNA cryptographic technique based on dynamic DNA sequence table,” In 19th Int. Conf. on Computer and Information Technology (ICCIT), pp. 270-275, IEEE, 2016.

[12]. G. Xiao, M. Lu, L. Qin, and X. Lai, “New field of cryptography: DNA cryptography,” Chinese Science Bulletin, Vol. 51, No. 12, pp. 1413-1420, 2006.

[13]. R. L. Stears, T. Martinsky, and M. Schena, “Trends in microarray analysis,” Nature medicine, 9.1, pp. 140-145, 2003.

[14]. S. Goyat, and S. Jain, “A secure cryptographic cloud communication using DNA cryptographic technique,” in Int. Conf. on Inventive Computation Technologies (ICICT), Vol. 3. pp. 1-8, IEEE, 2016.

[15]. B. M. Krishna, G. Madhumati and H. Khan, “Dynamically Evolvable Hardware-Software Co-Design Based Crypto System Through Partial Reconfiguration,” Journal of Theoretical & Applied Information Technology, 95(10), 2017.