# Enhanced Information Security using DNA Cryptographic Approach

**Abhishek Majumdar, Meenakshi Sharma**

***Abstract*— In present days the transmission of data in a secured manner is a big issue. During the transmission different kinds of attack may happen and affect the data. Due to that reason lot of researchers are still working to provide better and secured cryptographic algorithms. The DNA Cryptography is a new and promising area to achieve higher information security, where the characteristics of human DNA molecules are followed as the DNA have complex structure and features. In this paper a couple of 128 bit publicly available DNA sequences are taken to form the secret keys. Moreover a better level of message encryption technique is proposed where two rounds encryption has been carried out among the plain text and the generated two secret keys and produce a cipher DNA sequence with appending some extra information within it.***

***Index Terms*— DNA sequence, Final Cipher, Key, Nucleotide**

## I. INTRODUCTION

Cryptography means to convert a known text into any coded format that can only be understood by the intended users. Several cryptographic approaches are proposed till now by thousands of researchers for several years. They performed various kinds of complex mathematical computations on the data to enhance the security of the information. But still more effective cryptographic algorithms are required to make information more confidential and secure. DNA based encryption method is one of the recent technique in cryptographic field that can provide higher security of the information. The DNA actually stands for Deoxyribo Nucleic Acid. A DNA is nothing but a double helix made up of two strands where each strand can consists either a purine or a pyramidine base. The purine bases are adenine (A) and guanine (G), while the pyrimidines bases are thymine (T) and cytosine (C), also known as the 4 basic nucleotide bases of DNA. Today, DNA based cryptography is taken as a most promising area of research by several researchers due to having the complex structural features and several special characteristics of the DNA. Out of them some used DNA computing, while some other incorporated biological properties of DNA strands and DNA sequence in their algorithms. In this paper a DNA based cryptographic approach is proposed where two publically available 128 bit DNA sequence is shared among the sender and the receiver and is used for the secret key generation that will be used the encryption phase. This secret key works on each plaintext blocks in two rounds by following an ordered manner and generate the cipher text and later on it will also converted to a DNA sequence with appending some extra bits of information within it, so that it will only be retrieved by the intended receiver.

**Manuscript Received on July 2014.**

**Abhishek Majumdar**, Department of Computer Science & Engineering, Sri Sai College of Engineering & Technology, Badhani, India.

**Meenakshi Sharma**, Department of Computer Science & Engineering, Sri Sai College of Engineering & Technology, Badhani, India.

## II. RELATED WORKS

Lots of DNA based encryption methods are proposed by several researchers, where several kind of encryption process was depicted. H. Z. Hsu, R. C. T. Lee et al. have given a idea in which they used some special properties of DNA sequences to encrypt data and for that they discussed three methods, and for each method, they secretly select a reference DNA sequence [1]. Amal Khalifa and Ahmed Atito et al. proposed a method of data hiding where the data is encrypted using amino acid and DNA based playfair cipher and also use complementary rules to hide the resultant cipher text in a DNA sequence [2]. Mohammad Reza Abbasy, Pourya Nikfard et al. used a sort of indexing method over the complementary DNA sequence [3]. Sabari Pramanik, Sanjit Kumar Setua et al. proposed an encryption scheme in which to encrypt the plain text a single stranded DNA string was taken as the secret key depending on the length of the plain text. Moreover they send the plain text as several DNA plain text packets by attaching the packet sequence number with each packet [4]. Suman Chakraborty, Sudipta Roy et al. had incorporated an idea of DNA based image encryption using soduko solution matrix to perform some computations on behalf of the message [5]. Nirmalya Kar, Atanu Majumder et al. had proposed a more secure and reliable encryption scheme by using the technologies of DNA sequence and DNA synthesis. They used three keys for encrypting the message along with a new method of key generation and key sharing. Instead of directly sharing the key, a session key holding the information regarding the actual encryption key was shared among two parties [6]. The researchers are implementing to enhance the security of the cipher text by appending extra coded information with it at different location of the cipher text [8]. Bibhash Roy, Atanu Majumder et al.has proposed an encryption method in which two levels of encryption took place that was concerned with how DNA sequencing can be used in the field of cryptography [7]. Researchers like Xing Wang, Qiang Zhang et al. derived a new way to show how cryptography works with DNA computing, it can transmit message securely and effectively. They have used RSA algorithm along with DNA computing theory [9].

## III. PROPOSED SYSTEM

The proposed method in this paper provides a secured & reliable data transmission. Here the overall method is done by 3 sub phases; these are the key generation, data encryption and the use of DNA encoding to provide better level of security.

### *A. Key Generation and Selection*

In this first phase, at first the sender will select two 128 bit DNA sequence randomly from publicly available DNA sequences. These two selected DNA sequences will produce the encryption keys after performing a large number of computations on it. The selected 128 bit DNA Sequences form a 256 bit DNA sequence by applying the method where each base of a DNA sequence is concatenated with the base in the same position in the another chosen DNA sequence. For example, let, the two randomly chosen DNA sequence is termed as P and Q respectively. Then take a 256 bit buffer and stores each result of [Concate(Pi, Qi) ; for 1 < i <128] , where each Pi and Qi may be any nucleotide base (A,T,C,G).As a result a Completely different 256 DNA sequence is induced. Then this 256 bit DNA sequence is divided into 64 blocks consists of 4 nucleotide bases each. After that every block is checked through the Table I, where for every possible 4 base combinations (total 44 = 256 combinations) the corresponding row and column base index (BI) is listed in an array and also the row and group base index (GBI) is listed in another array. In Table I, there exist 8 group base indices, where every group base index contains either 4 row or 4 column base indices. So, it can be possible to have the same base index pair of more than one 4base DNA strand but having different group index pairs.

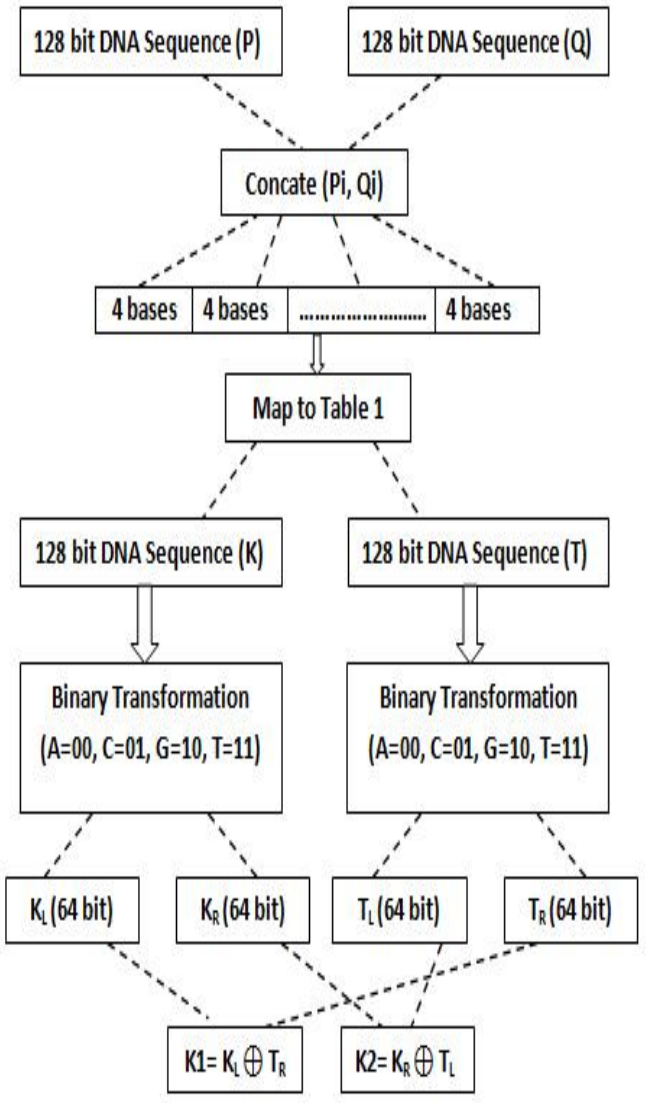


**Fig. 1 Process of key generation**

Similarly every 4 base combinations will produce a base index pair and a group base index pair. As a result at last the taken 256 bit DNA sequence will be divided into two 128 bit separate DNA sequences.

**TABLE I**
**GROUP BASE INDEX AND BASE INDEX TABLE OF 256 COMBINATIONS OF 4 DNA BASES**

| GBI | | A | | | |
|---|---|---|---|---|---|
| | BI | A | T | C | G |
| A | A | TTTT | TCTT | TATT | TGTT |
| | T | TTTC | TCTC | TATC | TGTC |
| | C | TTTA | TCTA | TATA | TGTA |
| | G | TTTG | TCTG | TATG | TGTG |
| T | A | TTCT | TCCT | TACT | TGCT |
| | T | TTCC | TCCC | TACC | TGCC |
| | C | TTCA | TCCA | TACA | TGCA |
| | G | TTCG | TCCG | TACG | TGCG |
| C | A | TTAT | TCAT | TAAT | TGAT |
| | T | TTAC | TCAC | TAAC | TGAC |
| | C | TTAA | TCAA | TAAA | TGAA |
| | G | TTAG | TCAG | TAAG | TGAG |
| G | A | TTGT | TCGT | TAGT | TGGT |
| | T | TTGC | TCGC | TAGC | TGGC |
| | C | TTGA | TCGA | TAGA | TGGA |
| | G | TTGG | TCGG | TAGG | TGGG |
| GBI | | T | | | |
| | BI | A | T | C | G |
| A | A | CTTT | CCTT | CATT | CGTT |
| | T | CTTC | CCTC | CATC | CGTC |
| | C | CTTA | CCTA | CATA | CGTA |
| | G | CTTG | CCTG | CATG | CGTG |
| T | A | CTCT | CCCT | CACT | CGCT |
| | T | CTCC | CCCC | CACC | CGCC |
| | C | CTCA | CCCA | CACA | CGCA |
| | G | CTCG | CCCG | CACG | CGCG |
| C | A | CTAT | CCAT | CAAT | CGAT |
| | T | CTAC | CCAC | CAAC | CGAC |
| | C | CTAA | CCAA | CAAA | CGAA |
| | G | CTAG | CCAG | CAAG | CGAG |
| G | A | CTGT | CCGT | CAGT | CGGT |
| | T | CTGC | CCGC | CAGC | CGGC |
| | C | CTGA | CCGA | CAGA | CGGA |
| | G | CTGG | CCGG | CAGG | CGGG |
| GBI | | C | | | |
| | BI | A | T | C | G |
| A | A | ATTT | ACTT | AATT | AGTT |
| | T | ATTC | ACTC | AATC | AGTC |
| | C | ATTA | ACTA | AATA | AGTA |
| | G | ATTG | ACTG | AATG | AGTG |
| T | A | ATCT | ACCT | AACT | AGCT |
| | T | ATCC | ACCC | AACC | AGCC |
| | C | ATCA | ACCA | AACA | AGCA |
| | G | ATCG | ACCG | AACG | AGCG |
| C | A | ATAT | ACAT | AAAT | AGAT |
| | T | ATAC | ACAC | AAAC | AGAC |
| | C | ATAA | ACAA | AAAA | AGAA |
| | G | ATAG | ACAG | AAAG | AGAG |
| G | A | ATGT | ACGT | AAGT | AGGT |
| | T | ATGC | ACGC | AAGC | AGGC |
| | C | ATGA | ACGA | AAGA | AGGA |
| | G | ATGG | ACGG | AAGG | AGGG |
| GBI | | G | | | |
| | BI | A | T | C | G |
| A | A | GTTT | GCTT | GATT | GGTT |
| | T | GTTC | GCTC | GATC | GGTC |
| | C | GTTA | GCTA | GATA | GGTA |
| | G | GTTG | GCTG | GATG | GGTG |
| T | A | GTCT | GCCT | GACT | GGCT |
| | T | GTCC | GCCC | GACC | GGCC |
| | C | GTCA | GCCA | GACA | GGCA |
| | G | GTCG | GCCG | GACG | GGCG |
| C | A | GTAT | GCAT | GAAT | GGAT |
| | T | GTAC | GCAC | GAAC | GGAC |
| | C | GTAA | GCAA | GAAA | GGAA |
| | G | GTAG | GCAG | GAAG | GGAG |
| G | A | GTGT | GCGT | GAGT | GGGT |
| | T | GTGC | GCGC | GAGC | GGGC |
| | C | GTGA | GCGA | GAGA | GGGA |
| | G | GTGG | GCGG | GAGG | GGGG |

Now let the resultant two DNA sequences are termed as K and T and these K and T are further simply subdivided into 2 parts of 32 bit of DNA sequences. Let for K the 64bit parts are KL and KR respectively and for T 64bit parts are TL and TR respectively. Now generate the keys K1 and K2 as:

$$K1= K_L \oplus T_R$$

$$K2= K_R \oplus T_L$$

These 2 keys of 64 bit each viz. K1 and K2 will be used in the phase of message encryption. The key selection and generation phase is depicted in Fig 1.

**Algorithmic steps:**

**Input:** Randomly chosen two 128 bit DNA sequences.
**Output:** Two 64 bit encryption keys.
**Step 1**-Let the randomly chosen two 128 bit DNA sequences are:

P='TACCACGTCGTGTCCCAGGACCATACGGTGAA CGTAAACGCTTAAAATTTAGGGCTCCCAGTCG'

Q='TTAAAGTCCGCCATATTGGAAGTCGCAAAAG TACGTACGGCTCCCTATATCGCGTTCCAAACCA'

**Step 2-** After blocking of every 4 adjacent nucleotides:

P = 'TACC ACGT CGTG TCCC AGGA CCAT ACGG TGAA CGTA AACG CTTA AAAT TTAG GGCT CCCA GTCG'

Q= 'TTAA AGTC CGCC ATAT TGGA AGTC GCAA AAGT ACGT ACGG CTCC CTAT ATCG CGTT CCAA ACCA'

**Step 3-** After each Concate (Pi, Qi) operation the generated 256 bit DNA sequence is:

DNA seq = 'TTAT CACA AACG GTTC CCGG TCGC TACT CACT ATGG GGAA CACG ATTC AGCC GCGC TAGA AGAT CAGC TGAT AAAC CGGG CCTT TCAC ACAT AATT TATT ACGG GCGG CTTT CCCC CAAA GATC CCGA'

**Step 4-** After mapping the DNAseq Table I, two separate 128 bit DNA sequences named K and T is produced.

K =' AA CC GC TA GT TT AC AC GA CG GC TA TG TT CC AG TC AG TC GG AT TT AT AC AC GT GT AA TT CC TC CT'

T = 'CA TT TC AG GT GA TA TT GC CG TT AC TC GG GA CC GT CA CC GT AT CA CC AC AA GC GG AT TT CT AG GT'

**Step 5-** Binary Substitution:

(A-00, T-11, C-01, G-10)

K='0000010110011100101111110001000110000110100 1110011101111010100101101001011011010001111110011000100011011101100001111010111010111'

T='0100111111010010101110001100111110010110111 1000111011010100001011011010001011011001101000101000100001001101000111111011100101011'

**Step 6-** Divide the K and T into two 64 bit blocks each.

$K_L$='0000010110011100101111110001000110000110100111001110111101010010'

$K_R$='1101001011011010001111110011000100011011101100001111010111010111'

$T_L$='0100111111010010101110001100111110010110111100011101101010000101'

$T_R$='1011010001011011001101000101000100001001101000111111011100101011'

**Step 7-** Generation of two 64 bit Encryption keys.

K1= $K_L \oplus T_R$= 1011000111000111100010110100000010001111001111110001100001111001

K2= $K_R \oplus T_L$= 1001110100001000100001111111111010001101010000010010111101010010

### *B. Message Encryption*

In this phase, first of all any kind of file is chosen that is to be transmitted. It may either a simple text file or any document and multimedia file. After that the byte values of the input file are extracted. These unsigned byte values are called Plaintext that participate in the encryption process. Then these byte values will be transformed into 8-bit binary. The Plaintext is then divided into a number of 64 bit blocks, where each block will go through the encryption process with the two round keys K1 and K2 generated in the key generation phase. Now each 64 bit block of Plaintext termed as PT is subdivided into four 16 bit blocks of Plaintext. Similarly the 64 bit round 1 key K1 is also subdivided into four 16 bit blocks, then perform an EX-OR operation between the plaintext PT and the round1 encryption key K1 and generate the intermediate ciphertext ICT. Here the EX-OR operation that is performed is different than the directly performing EX-OR operation. For this here, every 16 bit blocks of each plaintext and the round key are further subdivided into 2 parts of 8 bit each. Then every 8 bit subpart of in every 16 bit part of plaintext is EX-ORed to the diagonal 8 bit subpart in the corresponding 16 bit round key and store it as a new subpart of the intermediate ciphertext. Similarly the every part of both plaintext and the selected key is EX-ORed and generate a new 64 bit intermediate ciphertext ICT and it will be acted as input for the second round. In the round2 the same operation is carried out between the induced intermediate ciphertext ICT and the round2 key K2. The resultant value of round2 is termed as final cipher text (F_CT). The EX-OR operation among the 8bit subparts of both the plaintext/intermediate cipher text and the selected round encryption keys is depicted in Fig. 2.

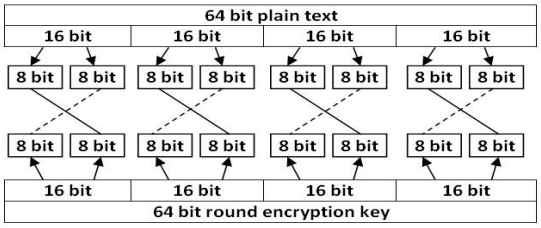


**Fig. 2 Proposed EX-OR operation**

Now the induced 64 bit F_CT is divided into 4 parts having 16bit each. All of these 4 parts are transformed into their hexadecimal form by a special operation. For this initially the 16 bit 1st part is taken and stored in 'tmp'. After that divide each part by 16 and convert the remainder into its equivalent hexadecimal form and keep it in CT1.Divide the result once again by 16 keeping hex form of the remainder in CT2. Do until the result is less than 16. Make together all the CTi in order to get the cipher text in hex form.

**TABLE III HEXADECIMAL TO DNA CONVERSION**

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

At last of this message encryption phase the cipher text parts in hex form are combined and perform a DNA encoding operation as given in Table II, in which for every hex digit there is a corresponding DNA representation of 2 DNA bases (With 2 bases at most 42 =16 combinations can be made).As a result finally the plain text is transformed into a DNA sequence which is never possible to understand by the intruders. Fig. 3 depicts the overall proposed message encryption process.

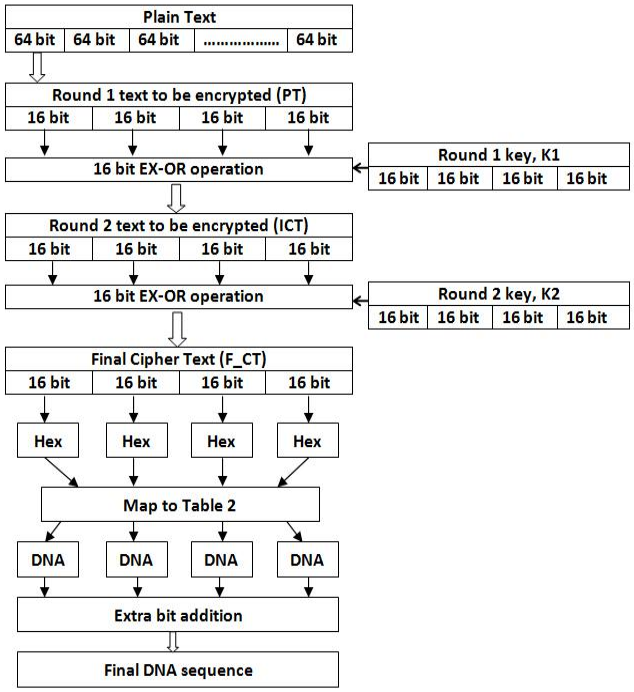


**Fig. 3 Proposed message encryption method**

**Algorithmic steps:**

**Input:** 2 keys K1 and K2 of 64 bit each; A file that has to be transmitted.

**Output:** A DNA sequence.

**Step 1-** Let, a 64 bit block of the plaintext be, PTi ='1001 0011 0110 0111 1011 1101 1001 1100 1111 0101 0010 1101 0010 1101 1010 0011'

**Step 2-** PT is subdivided to Four 16 bit parts:

P1= 1001 0011 0110 0111

P2= 1011 1101 1001 1100

P3= 1111 0101 0010 1101

P4= 0010 1101 1010 0011

8 bit subparts of P1, P2, P3, P4:

$P1_L$= 1001 0011; $P1_R$= 0110 0111

$P2_L$= 1011 1101; $P2_R$= 1001 1100

$P3_L$= 1111 0101; $P3_4$= 0010 1101

$P4_L$= 0010 1101; $P4_R$= 1010 0011

**Step 3-** Round 1:

Selected encryption Key = Key1.

K1 is subdivided to Four 16 bit parts:

K11= 1011 0001 1100 0111

K12= 1000 1011 0100 0000

K13= 1000 1111 0011 1111

K14= 0001 1000 0111 1001

8 bit subparts of key1:

$K11_L$= 1011 0001; $K11_R$= 1100 0111

$K12_L$= 1000 1011; $K12_R$= 0100 0000

$K13_L$= 1000 1111; $K13_R$= 0011 1111

$K14_L$= 0001 1000; $K14_R$= 0111 1001

Temporary variables T11, T12, T13, T14;

Compute-

$T11= Concate [(P1_L \oplus K11_R), (P1_R \oplus K11_L)]$;

= 0101 0100 1101 0110

$T12= Concate [(P2_L \oplus K12_R), (P2_R \oplus K12_L)]$;

= 1111 1101 0001 0111

$T13= Concate [(P3_L \oplus K13_R), (P3_R \oplus K13_L)]$;

= 1100 1010 1010 0010

$T14= Concate [(P4_L \oplus K14_R), (P4_R \oplus K14_L)]$;

= 0101 0100 1011 1011

Compute Intermediate Ciphertext –

ICT=Concate (T11, T12, T13, T14)

= 0101 0100 1101 0110 1111 1101 0001 0111 1100 1010 1010 0010 0101 0100 1011 1011

**Step 4-** Round 2:

Temporary variables T21, T22, T23, T24

Selected encryption Key = Key2.

Plain text PT =ICT.

8 bit subparts of P1, P2, P3, P4:

$P1_L$= 0101 0100; $P1_R$= 1101 0110

$P2_L$= 1111 1101; $P2_R$= 0001 0111

$P3_L$= 1100 1010; $P3_4$= 1010 0010

$P4_L$= 0101 0100; $P4_R$= 1011 1011

8 bit subparts of key2:

$K21_L$= 1001 1101; $K21_R$= 0000 1000

$K22_L$= 1000 0111; $K22_R$= 1111 1110

$K23_L$= 1000 1101; $K23_R$= 0100 0001

$K24_L$= 0010 1111; $K24_R$= 0101 0010

Compute-

$T21= Concate [(P1_L \oplus K21_R), (P1_R \oplus K21_L)]$;

= 0101 1100 0100 1011

$T22= Concate [(P2_L \oplus K22_R), (P2_R \oplus K22_L)]$;

= 0000 0011 1001 0000

$T23= Concate [(P3_L \oplus K23_R), (P3_R \oplus K23_L)]$;

= 1000 1011 0010 1111

$T24= Concate [(P4_L \oplus K24_R), (P4_R \oplus K24_L)]$;

= 0000 0110 1001 0100

Compute Final Ciphertext –

F_CT=Concate (T21, T22, T23, T24)

= 0101 1100 0100 1011 0000 0011 1001 0000 1000 1011 0010 1111 0000 0110 1001 0100

**Step 5-** Division of F_CT into four parts of 16 bit.

F_CT1= T21; F_CT2= T22; F_CT= T23; F_CT= T24

**Step 6-** Transformation to hexadecimal form.

For, $F_CT1= (0101\ 1100\ 0100\ 1011)_2$

$tmp=F_CT1= (23627)_{10}$

CT1= 23627% 16 = 11 = B (hex form).

tmp= 23627/ 16 = 1476,

CT2= 1476% 16 = 4 = 4 (hex form).

tmp= 1476/ 16 = 92,

CT3= 92% 16 = 12 = C (hex form).

tmp= 92/ 16 = 5,

CT4= 5 = 5 (hex form).

So, **F_CT1=B4C5.**

Now for, F_CT2= $(0000\ 0011\ 1001\ 0000)_2$

$tmp=F_CT2= (912)_{10}$

CT1= 912% 16 = 0 = 0 (hex form).
tmp= 912/ 16 = 57,
CT2= 57% 16 = 9 = 9 (hex form).
tmp= 57/ 16 = 3,
CT3= 3 = 3 (hex form).
So, **F_CT2=093.**
Now for, F_CT3= $(1000\ 1011\ 0010\ 1111)_2$
tmp=F_CT3= $(35631)_{10}$
CT1= 35631% 16 = 15 = F (hex form).
tmp= 35631/ 16 = 2226,
CT2= 2226% 16 = 2 = 2 (hex form).
tmp= 2226/ 16 = 139,
CT3= 139% 16 = 11 = B (hex form).
tmp= 139/ 16 = 8,
CT4= 8 = 8 (hex form).
So, **F_CT3=F2B8.**
Now for, F_CT4= $(0000\ 0110\ 1001\ 0100)_2$
tmp=F_CT4= $(1684)_{10}$
CT1= 1684% 16 = 4 = 4 (hex form).
tmp= 1684/ 16 = 105,
CT2= 105% 16 = 9 = 9 (hex form).
tmp= 105/ 16 = 6,
CT3= 6 = 6 (hex form).
So, **F_CT4=496.**
**Step 7-** Extra bit padding
Pad number of digits prior to each F_CTi.
F_CT1=**4**B4C5
F_CT2=**3**093
F_CT3=**4**F2B8
F_CT4=**3**496
**Step 8-** DNA encoding as per Table II.
F_CT1=**4**B4C5 = TACGTAGATT
F_CT2=**3**093= AGAACTAG
F_CT3=**4**F2B8= TAGGACCGCA
F_CT4=**3**496= AGTACTTC
This padding is due to get the correct information back at receiver side as the DNA sequence parts vary in their lengths.
**Step 9-** Final DNA sequence
Club together all the F_CTi :
F_DNA= TACGTAGATTAGAACTAGTAGGACCGCA AGTACTTC

## IV. CONCLUSION

In this paper the proposed approach is more secure than any other cryptographic algorithm. The key used in this algorithm is a couple of 28 bit DNA sequence to generate the secret encryption keys after a number of computations. As there approximately 55 million publicly available DNA sequences, so it is almost impossible to an intruder to predict this sequence. The message encryption approach is also better than the available cryptographic algorithms based on the DNA due to using some special operations performed on the data. Thus it will very much difficult for the intruders to apply different cryptanalysis on the cipher text.

## REFERENCES

1. H.Z. Hsu and R.C.T.Lee, "DNA Based Encryption Methods", The 23rd Workshop on Combinatorial Mathematics and Computation Theory, National Chi Nan University Puli, Nantou Hsies, Taiwan 545, April 2006.
2. Amal Khalifa and Ahmed Atito. "High-Capacity DNA-based Steganography", In the 8th International Conference and informatics and Systems (INFOS2012),IEEE,May.2012.
3. Mohammad Reza Abbasy, Pourya Nikfard, Ali Ordi, Mohammad Reza Najaf Torkaman, "DNA Base Data Hiding Algorithm", In: International Journal on New Computer Architectures and Their Applications.2012.
4. Sabari Pramanik, Sanjit Kumar Setua, "DNA Cryptography", In: ICECE,2012,pp.551-554.IEEE.2012. doi:10.1109/ICECE.2012.6471609
5. Suman Chakraborty, Sudipta Roy, Prof. Samir K. Bandyopadhyay, "Image Steganography Using DNA Sequence and Sudoku Solution Matrix". In: International Journal of Advanced Research in Computer Science and Software Engineering,Feb 2012.
6. Nirmalya Kar, Atanu Majumder, Ashim Saha, Anupam Jamatia, Kunal Chakma, Dr. Mukul Chandra Pal,"MobileHealth'13", July 29, 2013, Bangalore, India.ACM(2013).
7. Bibhash Roy, Atanu Majumder," An Improved Concept of Cryptography Based on DNA Sequencing",In: International Journal of Electronics Communication and Computer Engineering. Vol-3, Issue-6 (Nov2012).
8. Bibhash Roy, Gautam Rakshit, Pratim Singha, Atanu Majumder, Debabrata Datta, "An improved Symmetric Key Cryptography with DNA Based Strong Cipher", ICDeCom-2011, BIT Mesra, Ranchi, Jarkhand, India, Feb 2011.
9. Xing Wang, Qiang Zhang," DNA computing-based cryptography", In Proc. of the 2009 IEEE International Conference, ISBN: 978-1-4244-3867-9/09.
10. Behrouz A.Forouzen, Debdeep Mukhopadhyay, "Cryptography and Network Security",2nd edition,Tata McGraw Hill Education Pvt.Ltd.

## AUTHORS PROFILE

**Abhishek Majumdar**, is pursuing M.Tech in Computer Science & Engineering at SSCET, Badhani, Punjab, India under Punjab Technical University, Jalandhar. His ongoing research area is on DNA based Cryptography and Steganography and his areas of interest are Network Security, image processing, and Bioinformatics.

**Meenakshi Sharma**, is working as Associate Professor in Department of Computer Science & Engineering, SSCET, Badhani, Punjab, India. She has more than 16 years of teaching experience. Her areas of interest are Parallel Computing, Cloud Computing, and Network Security.