International Conference on Mathematics, Statistics & Information Technology
(ICMSIT 2016), 20-22 Dec. 2016, Tanta, Egypt.

Information hiding using artificial DNA sequences based on Elliptic Curve

Eman I. Abd El- Latif *eman.mohamed@fsc.bu.edu.eg*
Department of Mathematics,
Faculty of Science, Benha University,
Benha, Egypt

Mahmoud I. Moussa *mahmoud.mossa@fci.bu.edu.eg*
Department of Computer Sciences,
Faculty of Computers and Informatics,
Benha University, Benha, Egypt

Abstract

Cryptography is one of the major concerned areas of computer networks and data security. An efficient direction of providing data security can be termed as DNA based on cryptography. In this paper, a new algorithm, to hide secret messages in a sequence of DNA, is proposed. This algorithm is based on the notion of elliptic curve cryptography. Two selected DNA sequences from DNA database are used. Our algorithm is similar to the well-known S-DES algorithm.

Keywords: DNA Cryptography, DNA Sequence, DES, ECC.

1. Introduction

DNA cryptography is hiding a data in terms of DNA sequences, which can be done using several DNA technologies with the biological methods. In recent years, the use of data hiding approach has become popular in transmitting secret messages. Data hiding approaches based on the DNA sequence attracted much attention to avoid malicious intruding and fulfill a safe transmission. The DNA sequence is made up of four different types of bases, Guanine-G, Adenine-A, Thymine-T and Cytosine-C. Each base is attached to a sugar molecule and a phosphate molecule. Together, a base, sugar, and phosphate are called a nucleotide

The order of these bases determines the information available for building and maintaining an organism. DNA has much more storage capacity which is equal to (1gm=10^8 TERA). It means small amount of DNA can stores world's information. DNA cryptography method is one of the new techniques in cryptographic field that can provide higher security of the information. Cryptographic schemes use one key to encrypt and decrypt the message is referred as symmetric-key. A few well-known

symmetric key cryptography examples are; the Data Encryption Standard (DES), Triple-DES (3DES) and the Advanced Encryption Standard (AES). The Cryptographic schemes use two keys; one is used for encryption "called public key" and the other one for decryption "called private key" are referred as asymmetric key encryption. In the majority of cryptographic applications in practical systems, symmetric and asymmetric algorithms are used together to construct hybrid schemes.

To deal with data hiding problem, many cryptographic schemes were proposed with cryptographic based on DNA such as a symmetric DNA-based cipher approach [1],[2]. The secret message was encrypted using RSA algorithm and then hidden in DNA sequence using complementary character [1]. A new scheme to hide two secret bits from a message by replacing one character in DNA sequence is described recently. The researchers used a kind of mapping between one complementary rule and the two secret bits to hide the message and then send the fake DNA sequence [2]. An algorithm is proposed by using software point of view for implementing data hiding based on DNA sequences. Both of DNA's features and binary coding technology beside complementary pairing rules are needed. Data hiding is started by applying three different and separate steps to prepare cipher message [3]. A session keys are shared between the sender and the receiver instead of sharing the actual keys between them. These keys contain the information about the actual key that is used for encrypting the message. This DNA sequence is one of the key to encrypt the message in next step and then get the fake DNA sequence. Add some extra bits at the beginning and at the ending of the faked DNA sequence and send the total form of DNA sequence to the receiver[4],[5]. New method provides a secure and reliable data transmission has 3 sub-phases. They are the key generation, data encryption and the use of DNA encoding. The key generation is done by selected two 128-bit DNA sequence randomly from publicly available DNA sequences. These two selected DNA sequences will produce two encryption keys after performing a large number of computations on it. The data encryption technique is proposed where two rounds encryption has been carried out among the plain text and the generated two secret keys. A DNA encoding is converted every hex digit into a corresponding DNA representation of 2 DNA bases [6]. Eman and M. I. Moussa have proposed an algorithm based on two dimension chaotic system and DNA sequence. This algorithm uses the two dimensional chaotic map to generate two artificial DNA sequences S_1 and S_2. The sender uses the first sequence S1 for the encryption and uses the second sequence S2 to hide the cipher message randomly in a real third sequence S_3, which is selected from DNA database [7]. Neil Koblitz and Victor Miller [8] discovered Elliptic curve cryptography (ECC) in 1985. ECC is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. One of the main benefits in comparison with non-ECC cryptography is the same level of security provided by keys of smaller size. Fatma and M. I. Moussa

introduced a new data hiding algorithm based on deoxyribonucleic acid (DNA) sequence, where a DNA coding is used to encode plaintext instead of the classical 8-bit ASCII coding. The algorithm based on two DNA sequences [9]. Based on the properties hiding data in DNA sequence has been attracting much attention and research work has been carried out to propose several new methods [10], [11], [12], [13], [14].

The paper is organized as follow. Section 2 briefly introduces elliptic curves over finite field F_p . Section 3 presents the proposed algorithm to hide a secret message. The experimental and comparisons results are given in section 5; section 6 introduces the security analysis. Finally, section 7 is the conclusion.

2. Elliptic Curves over F_p

In order to apply the theory of elliptic curves to cryptography, we need to look at elliptic curves whose points have coordinates in F_p.

Let $p \geq 3$ be a prime number. Let E be an elliptic curve, defined over a finite field by the equation

$$y^2 = x^3 + ax + b \quad mod\ p \qquad (1)$$

Where $a, b \in F_p$ satisfying $4a^3 + 27b^2 \neq 0$
The points on E with coordinates in F_p are denoted by:

$$E(F_p) = \{(x, y): x, y \in F_p\ satisfy\ \ y^2 = x^3 + ax + b\} \cup \{\infty\}$$

3. The Proposed Algorithm

In this paper, we proposed an algorithm to hide a secret message (M) in DNA sequence to increase the security during transmission of data. The proposed algorithm consists of the following steps:

1. Generate all the pairs (x_i, y_i) satisfying equation (1) by choosing a prime number p and $a, b \in F_p$
2. Choose two variables c , q (where q is a prime number and $c > 1$), and apply equation (2).

$$f_i(x_i, y_i) = \exp(-\frac{|x_i - y_i|^2}{2c^2}) \qquad (2)$$

3. Apply equation (3) to generate a sequence of integers

$$h_i = 10^{14} * f_i\ \ mod\ q \qquad (3)$$

4. $sort\ (h_i)$. Put the result in an array A, and apply $z_i = A[i] + i$

5. Select a DNA sequence S_1 from DNA database.
6. For each z_i, we select the element in the position z_i in the DNA sequence $S_1$to generate K_1

$$K_1 = S_{1,z_i}$$

7. Locate the second repeated characters in K_1 and indicate them by bold.
8. Apply the complementary rule to the bold character. We can apply the following complementary rule: $(AT)(CA)(GC)(TG)$ .The resulting sequence is the second key (K_2).
9. Sender converts M into binary form using ASCII code.
10. Each two binary bit converted into DNA sequence based on table 1.

Binary form	nucleotide
00	A
01	C
10	G
11	T

Table 1: Binary Representation of Nucleotides

11. Sender encrypts the message (in the previous step) through two levels. In the first level, we use K_1, and in the second level, we use K_2.This encryption is similar to the well-known S-DES algorithm.
12. Select another DNA sequence S_2 from DNA database.
13. For each z_i, we hide character from encryption message (step 11) in $S_2$to generate the fake DNA sequence.
14. Send fake DNA sequence to the receiver.

3.1 Complementary Rules

There are six major possible complementary rules: $(AT)(TC)(CG)(GA)$, $(AT)(TG)(GC)(CA)$, $(AC)(CT)(TG)(GA)$, $(AC)(CG)(GT)(TA)$, $(AG)(GT)(TC)(CA)$, $(AG)(GC)(CT)(TA)$

In this paper, we used the following complementary rule: $(AT)(TC)(CG)(GA)$

3.2 Generation of Keys

In the algorithm 1, we generate two keys that are used in encryption algorithm. The algorithm consists of the following steps:

1. We used equation (1) to generate a sequence of pairs(x_i, y_i).
2. These pairs are used in equation (2).
3. Based on equation (2), a set of completely random sequence is generated between 0 and 1.
4. This sequence is converted into a sequence of integer by equation (3). We note that, some values of h_i appeared two or more time.
5. We sort h_i and store them in an array A and then apply $z_i = A[i] + i$
6. For each z_i, we select the element in the position z_i in the DNA sequence $S_1$to produce K_1 ($K_1 = S_{1,z_i}$)
7. The second key (K_2) is generated by applying the complementary rule on the second repeated letter in K_I.

Algorithm 1: Keys Generation

Input: DNA sequence (S_1), the proposed system and complementary rules.

Output: K_1 and K_2.

Step 1: Choose two prime numbers p and q.

Step2: Use equation (1) to generate a sequence of pairs (x_i, y_i)

Step 3: $for\ i = 1\ to$ number of pairs

$$f_i(x_i, y_i) = \exp(-\frac{|x_i - y_i|^2}{2c^2}), \qquad c > 1 \qquad (2)$$

$$h_i = 10^{14} * f_i \ mod\ q \qquad (3)$$

$end\ for$

Step 4: $sort(h_i)$ and put the result in array A

Step 5: $for\ i = 1\ to\ length\ (h)$

$z_i = A[i] + i$

$end\ for$

Step 6: Calculate $K_1 = S_{z_1}.S_{z_2}.S_{z_3} \ldots . S_{z_i}$

Step 7: Apply the complementary rule on the second repeated letter in $K_1$to generate K_2

3.3 Generation of XOR table and S_0-BOX

The XOR gate is a digital logic gate that executes an exclusive or. The output is "true" if either, but not both, of the inputs are "true". The output is "false" if both inputs are "false" or if both inputs are "true. An approach to recollect XOR is "the output is 1 if the inputs are different, but 0 if the inputs are the same". The binary XOR operator has the accompanying truth table 2.

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

Table 2: binary XOR truth table

We convert XOR table from binary system into DNA system that contains only four nucleotides (A, C, G, T) by the following steps:

1) Convert nuclide A to 00, C to 01, G to 10 and T to 11

2) Make exclusive or between each two binary bit according to table 2

For example:

$A \oplus C$ = 00 $\oplus$ 01=01= C

$G \oplus C$ = 10 $\oplus$ 01=11= T

Continuing in this fashion, we end up with a complete table. The results are listed in table 3.

	A	C	G	T
A	A	C	G	T
C	C	A	T	G
G	G	T	A	C
T	T	G	C	A

Table 3: XOR of DNA

In cryptography, an S-box (substitution-box) is a fundamental segment of symmetric key algorithms which performs the substitution. There are three requirements regarding the values in the S-box. First, the distributions of outputs must be checked for uniformity to protect against the Davies' Attack. Second, the outputs must have no linearity in their function to the input. Third, there must be unique values in every row of the S-box. Fixed tables are typically utilized as in the Simplified Data Encryption Standard (S-DES) as appeared in table 4.

We need to change S_0 –BOX of S-DES into S_0 –BOX DNA by replacing **0** by ***A***, **1** by ***C***, **2** by ***G***and **3** by ***T*** to generate table 5

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3

3	3	1	0	2

Table 4: S_0 –box of S-DES

	A	C	G	T
A	C	A	T	G
C	T	G	C	A
G	A	G	C	T
T	T	C	A	G

Table 5: S_0 –box of DNA

3.4 Message Cryptography Algorithm

The sender converts the message M into binary form by ASCII code.
So, there are two levels (level 1 and level 2) to get the encrypted message as shown in figure 1 and figure 2. Level 1 use K_1, and level 2 use K_2.

LEVEL 1: Encrypt the secret message using K_1

Step 1: Convert message M into ASCII binary code. Convert each two binary bits into DNA sequence by using table 1.

Step 2: Divide M into $M(L)\ and\ M(R)$

$$M = M(L)\,.M(R)$$

Step 3: To produce IC_1:

$$IC_1 = M(R) \oplus K_1$$

Step 4: To produce IC_2:

IC_2= Apply S_0-box into IC_1

The S_0-box operates as follows: The first input bit specifies a row of the S_0-box, and the second input bit specifies a column of the S_0-box and so on.

Step 5: Calculate $N = z_i \bmod L$, where L is the length of IC_2 in the encryption algorithm

Step 6: Read N from left to right where each number appears twice to generate R.

Step 7: To get IC_3: $IC_3 = expand\ IC_2 using\ R$

Step 8: To get IC_4: $IC_4 = M(L) \oplus IC_3$

Step 9: To produce IC_5: $IC_5 = IC_4.\ M(R)$

Step 10: To generate IC_6: $IC_6 = M(R).\ IC_4$

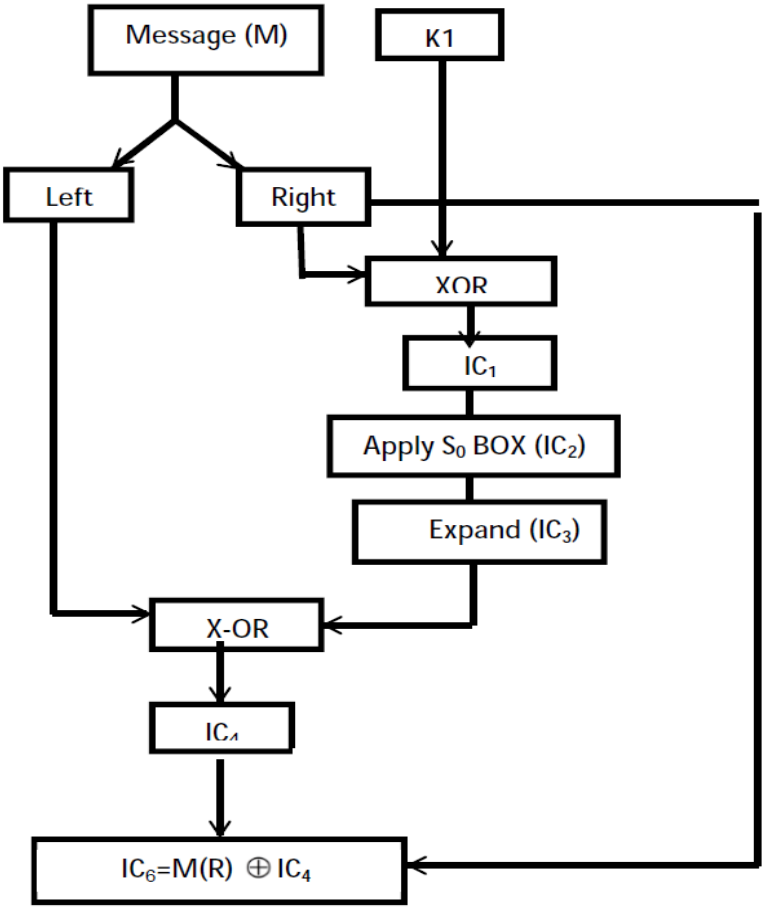

Figure 1: Message encryption diagram of level 1

LEVEL 2: Encrypt the secret message using K_2

Step 1: Divide $IC_6 into\ IC_6(L)\ and\ \ IC_6(R)$

Step 2: $Apply\ IC_2^{`} = IC_6(R) \oplus K_2$

Step 3: $Apply\ IC_3^{`} = S_0 - \text{box into } (IC_2^{`})$

Step 4: $Apply\ IC_4^{`}$= Expand $IC_3^{`}$

Step 5: $DO\ IC_5^{`}= IC_4^{`} \oplus IC_6(L)$

Step 6: $Apply\ IC_6^{`} = IC_5^{`} . IC_6(R)$

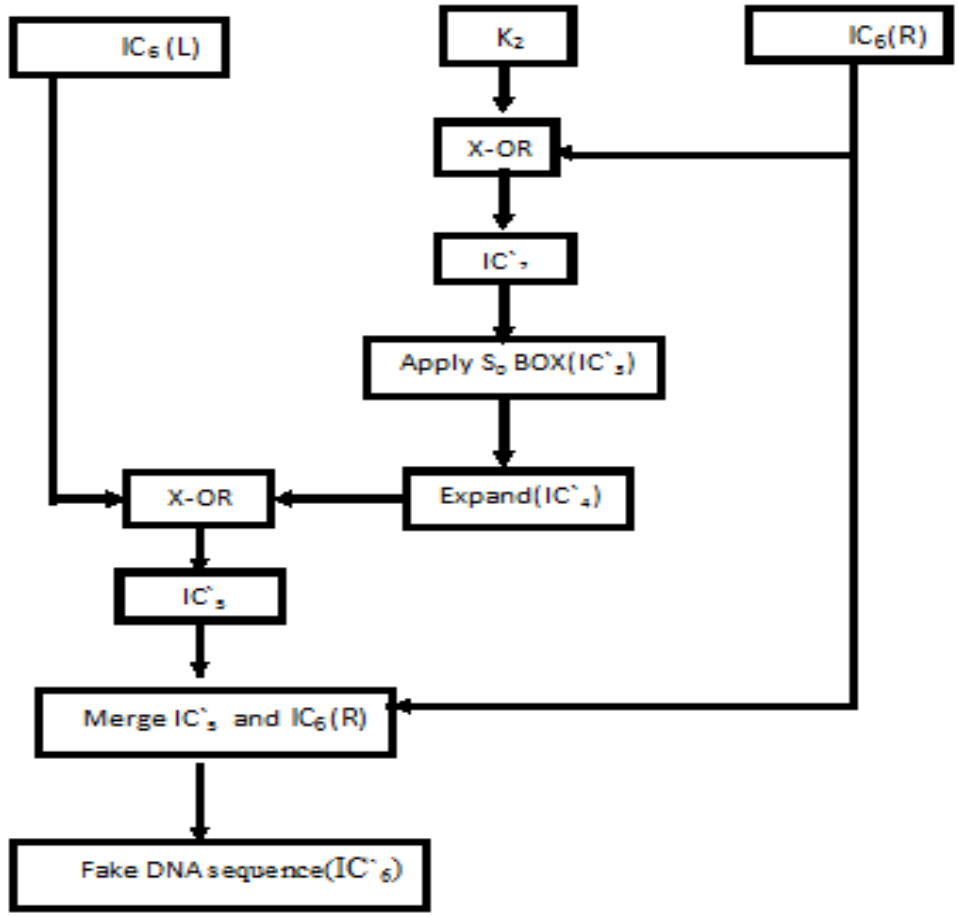

Figure 2: Message encryption diagram of level 2

3.5 Message Hiding Algorithm

In algorithm 3, we select S_2 from DNA database to hide the encrypted message.
Based on integer numbers (z_i), we replaced each nuclide from S_2 by nuclide from artificial DNA sequence ($IC_6^{`}$). Finally, send fake DNA sequence to the receiver.

Algorithm 3: Message Hiding Algorithm (a,b,p,q)

Input: S_2 and $IC_6^{`}$
Output: fake DNA sequence with hidden encrypted message
Step 1: $for\ i = 1\ to\ length(IC_6^{`})$
$replace\ S_2(z_i) to\ be\ IC_6^{`}(i)$
$end\ for$
Step 2: send fake DNA sequence to the receiver.

3.6 Data Recovery Algorithm

The sender sends the fake DNA sequence without any other DNA sequence or any sequence of number to the receiver.

The receiver extracts the artificial DNA sequence from fake DNA sequence, and then applying decryption steps in artificial DNA to get the original message.

Algorithm 4: Data Recovery Algorithm

Input: Fake DNA sequence and equation (2)

Output: Original message (M)

Step 1: Extract artificial DNA sequence ($IC_6^{`}$) according to numbers z_i

Step 2: Divide $IC_6^{`}$ into two halves, $IC_6^{`}(R)$, $IC_6^{`}(L)$

Step 3: To produce $IC_2^{`}$:

$$IC_2^{`} = K_2 \oplus IC_6^{`}(R)$$

Step 4: To produce $IC_3^{`}$:

$$IC_3^{`} = \text{Apply } S_0 \text{ –box in } IC_2^{`}$$

Step 5: $IC_4^{`} = Expand\ IC_3^{`}$

Step 6: To generate $M(R)$:

$$M(R) = IC_4^{`} \oplus IC_6^{`}(L)$$

Step 8: To get IC_1:

$$IC_1 = K_1 \oplus M(R)$$

Step 9: Apply $IC_2 = S_0$ – box in IC_1

Step 10: Apply $IC_3 = Expand\ IC_2$

Step 11: To produce $M(L)$:

$$M(L) = IC_3 \oplus IC_6^{`}(R)$$

Step 12: To get the original message M

$$M = M(L).M(R)$$

4. Numerical Example

In this example, we explain the proposed hiding algorithm in four steps.

- The first step is used to generate $K_1 and\ K_2$.
- The second step and the third step are used to encrypt the secret message using $K_1 and\ K_2$.
- The final step is used to hide the encrypted message in S_2.

Step1: **Keys Generation**

1. With $a = 5, b = 5\ and\ p = 17$, the elliptic curve equation over F_{17} is;

$$y^2 = x^3 + 5x + 5$$

The set of points which satisfy this equation are: (3,8) (3,9) (4,2) (4,15) (5,6) (5,11) (6,8) (6,9) (7,3) (7,14) (8,8) (8,9) (10,1) (10,16) (12,5) (12,12) (15,2) (15,15) (16,4) (16,13)

2. When $c = 2\ and\ q = 37,$ we get
   h_i ={ 16,8,14,30,5,8,14,18,23,9,26,5,20,8,9,26,19,26,3,18}
3. $sort\ (h_i)$ and put the result in array A and then add the value of $A[i]$ to the index i to generate z_i
   z_i= { 4,7,8,12,13,14,16,17,23,24,27,30,31,33,35,39,43,44,45,50}
4. Choose S_1=ATCGAATTCGGGCTGAGTCACAATTCGCG CTGAGTGAACC
5. K_1= $S_1(z_i)$ = GT**T**GCTAGATGCTATC
6. Apply the complementary rule on K_1 according to the injective map to get K_2 = GT**A**GCTAGATGCTATC

Step2: **Encrypt the secret message using K_1**

1. Let the message (M) = $hi = 0110100001101001 = CGGACGGC$
2. Divide M into left $M(L)$ and right $M(R)$
   $M(L) = CGGA,\ \ M(R) = CGGC$
3. $IC_1 = M(R) \oplus K_1 = TCCT$
4. $IC_2 = S_0$ – box (IC_1)
   - For S_0 – box :Row T , column C →output = C
     Row C , column T →output= A
   - Combining both result from S_0 – box we get : CA
5. IC_3 = Expand $IC_2 = CACA$
6. $IC_4 = IC_3 \oplus M(L) = AGTA$
7. $IC_6 = M(R)\ .\ IC_4 = CGGC\ AGTA$

Step3: Encrypt the secret message using K_2

1. Divide IC_6 into left $IC_6(L)$and right $IC_6(R)$
   $IC_6(L) = CGGC\ \ ,\ \ IC_6(R) = AGTA$
2. $IC_2^{`} = IC_6(R) \oplus K_2 = GCTA$
3. $IC_3^{`} = S_0$ – box $\left(IC_2^{`}\right)$
   - For S_0 – box :Row G , column C→output = G
     Row T, column A →output= T
   - Combining both result from S_0 – box we get : GT
4. $IC_4^{`} = Expand\ IC_3^{`} = GTGT$
5. $IC_5^{`} = IC_4^{`} \oplus IC_6(L) = TCAG$
6. $IC_6^{`} = IC_5^{`}.IC_6(R) = TCAG\ AGTA$

Step 4: Message Hiding

z_i = { 4,7,8,12,13,14,16,17,23,24,27,30,31,33,35,39,43,44,45,50}
S_2= TACCACGTCGTGTC CCA GGACCATACGGTGAACGTAAA CGCTTAAA ATTTAGGGCTCCCAGTCG

After hiding message we get the fake DNA sequence:

Fake DNA sequence = TACTACCACGTGAGCTA GGACCATACGGT GAACGTAAACGCTTAAAATTTAGGGCTCCCAGTCG

5. Experimental Results

A series of experiments carried out to assess the performance of the proposed algorithm (capacity, payload and bpn). In table 5, as appeared, eight DNA sequences are used as the test sample in the first column. These DNA sequences are publicly available and can be obtained by accessing the National Center for Biotechnology Information database (NCBI). The third column demonstrates the number of nucleotides before hiding the secret message.

The fourth column shows the number of nucleotides after hiding the secret message. The fifth column demonstrates the remaining length of new sequence after extracting out the reference DNA sequence.

The bpn columns demonstrate the number of bytes hidden per nucleotide. Sixth column of the same table shows results of using the algorithm in [9] on the same eight DNA sequences. But the last column shows results of applying the proposed algorithm.

Capacity and payload demonstrate that the length of the fake reference DNA sequence is not expanded, and bpn is within [0.202, 0.294]. The number of bits hidden per character in the proposed algorithm is higher than the number in the previous algorithm.

Locus	**Specifies definition**	**No. of nucleotides**	**Capacity C**	**Payload**	**bpn= |M|/C [9]**	**bpn of proposed algorithm**
AC153526	Mus musculus10 BAC RP23-383C2	200,117	200,117	0	0.071	0.220
AC166252	Mus musculus6 BAC RP23-100G10	149,884	149,884	0	0.072	0.294
AC167221	Mus musculus10 BAC RP23-3P24	204,841	204,841	0	0.070	0.215
AC168874	Bostaurus clone CH240-209N9	206,488	206,488	0	0.075	0.213
AC168897	Bostaurus clone CH240-190B15	200,203	200,203	0	0.077	0.220
AC168901	Bostaurus clone CH240-18511	191,456	191,456	0	0.073	0.230
AC168907	Bostaurus clone CH240-19517	194,226	194,226	0	0.075	0.227

AC168908	Bostaurus clone CH240-95K23	218,028	218,028	0	0.078	0.202

Table 5: The experimental results of the proposed algorithm

6. Security Analysis

In this section, the strength and the robustness of the proposed algorithm are based on the following:

- There are roughly 163 million DNA sequences available publicly. Thus, the probability of an attacker making a successful guess is $\frac{1}{1.63\times10^8}$. Another DNA sequence which is used to hide the secret message, so the probability of the attacker making a successful guess for the second selection is $\frac{1}{1.63\times10^8}$.
- Binary coding rule: as mentioned, the sender is free to select any equivalent binary form for every nucleotide. It means that A can be '00', '01', '10', or '11'; C can be '00', and so on. In other words, all the binary coding rules are 4×3×2×1=24. So, the likelihood of making the correct guess by the attacker is $\frac{1}{24}$.
- There are six possible complementary rules. The probability of an attacker to make a successful guess for the complementary rule is $\frac{1}{6}$.

The final probability of guessing the secret message is $(\frac{1}{1.63\times10^8})^2 \mathbf{x} \left(\frac{1}{6}\right) x \frac{1}{24}$

Based on the experimental results on DNA sequences, we can conclude that the proposed scheme has a high, stable embedding rate without expanding the length of the reference DNA sequence, and it achieves an appropriate tradeoff between embedding capacity and robustness.

7. Conclusion

In the encryption algorithm proposed here, the communicating parties concur upon to utilize elliptic curve cryptography. The security of the elliptic curve cryptography relies on the difficulty of finding the value of initial parameters. The elliptic curve parameters for cryptographic schemes ought to be carefully chosen in order to resist all known

attacks. Before encryption the message, we must generate two keys. The first key is generated by used ECC and first DNA sequence. The second key is generated by indicated the second characters repeated in first key and then establish a kind of injective mapping between one character and one complementary rule. The secret message is encryption in two levels using S-DES. Finally, hide the encrypted message in another DNA sequence. Hence, the method of encryption proposed here provides adequate security against cryptanalysis.

References

[1] B. A. Mitras and A. K. Aboo, "Proposed Steganography Approach Using Dna Properties", vol.14, 2012.

[2] C. Guo, C.-C. Chang, and Z.-H. Wang, "A new data hiding scheme based on DNA sequence," Int J Innov Comput Inf Control, vol. 8, pp. 1-11, 2012.

[3] M. R. Abbasy, P. Nikfard, A. Ordi, and M. R. N. Torkaman, "DNA Base Data Hiding Algorithm," International Journal of New Computer Architectures and their Applications (IJNCAA), vol. 2, pp. 183-192, 2012.

[4] N. Kar, A. Majumder, A. Saha, A. Jamatia, K. Chakma, and M. C. Pal, "An improved data security using DNA sequencing," in Proceedings of the 3rd ACM MobiHoc workshop on Pervasive wireless healthcare, vol.10, pp. 13-18,2013.

[5] B. Roy and A. Majumder, "An Improved Concept of Cryptography Based on DNA Sequencing," IJECCE, vol. 3, pp. 1264-1267, 2012.

[6] A. Majumdar and M. Sharma, "Enhanced Information Security using DNA Cryptographic Approach," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 4, pp. 72-76, 2014.

[7] Eman I. Abd El- Latif, and M. I. Moussa, “Chaotic Information- hiding Algorithm based on DNA”, International Journal of Computer Applications vol. 122, no. 10, pp. 41-45, July 2015.

[8] V. Miller, "Use of elliptic curves in cryptography," Advances in Cryptology—CRYPTO’85 Proceedings, pp. 417-426, 1986.

[9] Fatma E. Ibrahim, M. I. Moussa, and H. M. Abdalkader, “Enhancing the Security of Data Hiding Using Double DNA Sequences”, presented at Industry Academia Collaboration Conference (IAC), 6-8 April, Cairo, Egypt, 2015.

[10] E. I. Fatma, I. M. Mahmoud and S. A. Hatem, "A Symmetric Encryption Algorithm based on DNA Computing," International Journal of Computer Applications, vol. 97, no. 16, pp. 41-45, 2014.

[11] D. Bhattacharyya and S. K. Bandyopadhyay, "Hiding Secret Data in DNA Sequence‖," International Journal of Scientific &Engineering Research, vol. 4, 2013.

[12] Jin-Shiuh Taur, Heng-Yi Lin, Hsin-Lun Lee, Chin-Wang Tao,” Data hiding in DNA sequences based on table lookup substitution”, International Journal of Innovative Computing, Information and Control, vol. 8 , no.10(A), pp. 6585–6598, 2012

[13] E. Bashier, G. Ahmed, H.-A. Othman, and R. Shappo, "Hiding Secret Messages using Artificial DNA Sequences Generated by Integer Chaotic Maps," International Journal of Computer Applications, vol. 70, pp. 1-5, 2013.

[14] C.-C. Chang, T.-C. Lu, Y.-F. Chang, and R. Lee, "Reversible data hiding schemes for deoxyribonucleic acid (DNA) medium," International Journal of Innovative Computing, Information and Control, vol. 3, pp. 1145-1160, 2007.