A Symmetric Encryption Algorithm based on DNA Computing

Fatma E. Ibrahim
Computer Science Department, Faculty of Computers and Informatics, Benha University Benha,Egypt

M. I. Moussa
Computer Science Department, Faculty of Computers and Informatics, Benha University Benha,Egypt

H. M. Abdalkader
Information System Department, Faculty of Computers and Information, Menofia University Shebien, Egypt

ABSTRACT

Deoxyribo Nucleic Acid (DNA) computing is a new method of simulating the bimolecular structure of DNA and computing by means of molecular biology. DNA cryptography is a new field which has been explored worldwide. The concept of using DNA computing in the fields of cryptography and steganography has been identified as a possible technology, which may bring forward a new hope for unbreakable algorithms. This paper proposed a new DNA cryptographic algorithm which used the key features of DNA and amino acid coding to overcome limitations of the classical One Time Pad (OTP) cipher. A significant feature of the proposed algorithm is that; it is considered an encryption and hiding algorithm at the same time. The proposed algorithm also enhances the security level of OTP cipher. An evaluation for the proposed algorithm is performed according to randomness testing by using the National Institute of Standards and Technology (NIST) test. The study showed that the proposed algorithm had better performance with respect to time, capacity and robustness compared to previous studies.

General Terms

Security, Information Hiding.

Keywords

DNA Cryptography, Amino Acid, OTP, NIST Statistical Test.

1. INTRODUCTION

Network technologies are using for sending or receiving various kinds of digital data over the internet. Some of these data may be secret information which is a candidate to unauthorized access. A variety of techniques have been used to keep the unauthorized user away, such as cryptography and data hiding. The characteristics of DNA computing, massive parallelism, huge storage and ultra-low power consumption opened the door for researchers to utilize it in many fields especially in security. DNA computing, in the literal sense, is the use of DNA molecules, the molecules which encode genetic information for all living things, in computers. DNA computing is currently one of the fastest growing fields in both Computer Science and Biology, and its future looks extremely promising. DNA computing is a new method of simulating the bimolecular structure of DNA and computing by means of molecular biology. The first pioneering step done in the field of using DNA computing returns back to Leonard Adleman [1]. He used DNA to solve a well-known mathematical problem, called the directed Hamilton Path problem, also known as the ”traveling salesman” problem. He believes that we can use DNA computing to solve even the most difficult problems that require massive amount of parallel computing. Since then, scientists have produced a number of developments in this area, both theoretical and practical.

One potential key application of large scale computation system is DNA based cryptography. A new scheme which described a symmetric DNA-based cipher approach was introduced in [2]. The investigation conducted in that paper was based on a conventional symmetric encryption algorithm called “Yet Another Encryption Algorithm” (YAEA). The main target of that scheme was to introduce the concept of using DNA computing in the fields of cryptography in order to enhance the security of cryptographic algorithms. In [3] a new scheme that introduced the concept of using DNA and Amino Acid encoding in order to solve the limitations in old Playfair cipher has been proposed. This scheme turned the researchers to use DNA and Amino Acid with other weak encryption techniques to make them more robust and powerful. The first scheme of using DNA in the field of steganography was introduced in [4]. DNA encoded message is camouflaged within the enormous complexity of human genomic DNA and then further concealed by confining this sample to a microdot. Three data hiding methods were introduced based upon DNA sequence: the insertion method, the complementary pair method and the substitution method. In these methods; the secret message is embedded into a reference DNA sequence resulting in a new reference sequence with data hidden [5].

A different novel data hiding method based on DNA coding and using word document as host file was proposed in [6]. Firstly, encode the plaintext using DNA coding instead of using 8-bit ASCII. Secondly, generate two random aided DNA sequences to encrypt and conceal the cipher sequence. Finally, hide the cipher sequence into a word document by substituting the least significant 2-bit of the three color components. A reversible data hiding scheme based on histogram technique was proposed to solve the weaknesses of Shiu et al.’s schemes [7]. The idea in that scheme was to transform the DNA sequence into a binary string and then combines several bits into a decimal integer. These decimal integers are used to generate a histogram. Afterwards, the proposed scheme uses a histogram technique to embed secret data.

For watermarking an application of watermarks based on DNA sequences to identify the unauthorized use of genetically modified organisms (GMOs) protected by patents has been demonstrated. This application was based on transforming the character form of a message or any form of an image to the form of bits, and then this binary form can be transformed to DNA form through many encoding techniques [8].

The main target of this paper is to propose a new algorithm for encrypting data using key features of DNA computing and OTP cryptography. The proposed algorithm overcame the OTP limitations and increased its security level. The evaluation of the proposed scheme was carried out using the randomness testing, NIST. The security analysis and experimental results indicate better performance and low computational requirements for the proposed algorithm. The rest of this paper is organized as follows: Section2 briefly discussed OTP approach and its related work. Section3 described the proposed algorithm. Section4 introduced the experiment results and discussed the security analysis using NIST. Finally, in Section 5 contained the conclusion.

2. THE OTP APPROACH

Several ideas of using DNA strands in OTP cryptography have been presented [9, 10, 11]; however, experimental results and performance analysis have not been discussed enough. Other algorithms based on Double Transposition technique have been proposed to enhance security of OTP cipher. However, these were more complex and difficult to implement [12]. OTP is known as the only theoretically unbreakable cryptosystem. There are several conditions that must be met by user of OTP cipher: random key (pad) is truly random, never reused, and kept secret [13]. It is defined as follow:

OTP Step:

1. Generate a long fresh new random secret key (SK).
2. Subtract the secret key SK from the plaintext M to create cipher text C.
3. C = (M-SK) (mod26).
4. To decrypt cipher text C add it to the secret key SK.
5. M = (C+SK) (mod26).
6. The system as presented is thus symmetric.

OTP offers complete security, but in practice it has some limitations; (i) the key must be as long as message so as Not to be repeated, (ii) the key must be truly random, (iii) plaintext is any combination of specific Letters and symbols and (iv) generating a large number of random keys is no problem but printing, distributing, storing these keys are problems. In this paper a new scheme for OTP cryptography based on DNA computing was proposed. The proposed scheme enhances the security level of the classical OTP cipher and overcomes its limitations by utilizing characteristics of DNA and Amino Acid coding.

3. THE PROPOSED SCHEME

In binary computing field, the binary digital coding is the most fundamental used method, where anything can be encoded by 0 or 1. In any DNA sequence, there are four kind of bases, which are ADENINE (A) and THYMINE (T) or CYTOSINE(C) and GUANINE (G), which can be encoded based on 0 or 1 by four digits: 0=A(00), 1=C(01), 2=G(10), 3=T(11).

This binary coding is used to transfer the binary secret message to DNA sequence. Then from DNA sequence to Amino Acid form. There are 64 possible 3-letter combinations of the DNA coding units T, C, A and G, which are used either to encode one of these amino acids or as one of the three stop codons that signals the end of a sequence.

While DNA can be decoded unambiguously, it is not possible to predict a DNA sequence from its protein sequence. Because most amino acids have multiple codons, a number of possible DNA sequences might represent the same protein sequence. The number of amino acids available is 20 in addition to 1 start and 1 stop. So the remaining letters which are *(B, J, O, U, X, Z)* will share many codons from other amino acids. The DNA coding is applied over the message based on Table1 in which the maximum number of codons will be 4 instead of 6. According to the new distribution in that table, it gives 26 letters with the corresponding codons. A simple mapping is used to transfer letters to digits and then go through the OTP step. Assign for each letter of alphabet a number between 0 and 25 where $A = 0$, $B = 1,\ldots, Z = 25$. The resulting amino acid form of cipher text is transferred back to DNA using Table1. As each amino has more than one DNA codon a random codon is chosen to represent each amino acid. Randomness here increases strength of proposed scheme as it is very difficult and confusion to predict the codons used.

3.1 Encryption

The sender uses the following algorithm to encrypt the secret message.

Algorithm 3-1: Encryption algorithm

Input :	The plaintext, a secret key
Output:	Faked DNA sequence represents encrypted message
Step1.	Convert plaintext *[P]* to binary form *[BP]*.
Step2.	Transfer *[BP]* binary form to DNA sequence *[DP]*.
Step3.	Transfer DNA sequence *[DP]* to Amino Acids *[AP]* using Table1 and keep track with AMBIG number.
Step4.	Apply One Time Pad Step (AP, SK) to get amino acids of cipher *[AC]*.
Step5.	Transfer *[AC]* to DNA of cipher text *[DC]*.
Step6.	Finally the sender appended AMBIG number to *[DC]* to get the final cipher text *[C]*.

Both sender and receiver have copies of secret key which must be used only once and being destroyed, so the receiver has to reverse the encryption process. The only problem for receiver is how to construct DNA form of plain text from Amino Acid sequence. The "AMBIG" number was used here to correctly get the DNA codon for each amino acid. The resulting DNA sequence was transferred to binary and recovers the original plaintext.

Table 1. Amino acids, their single-letter data-base codes (slc), and their corresponding dna codons

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

3.2 Decryption

The receiver uses the following algorithm to get the plaintext.

Algorithm 3-2: Decryption algorithm

Input : Faked DNA sequence

Output: The plaintext

Step1. Extract AMBIG number and *[DC]* from *[C]*.

Step2. Transfer *[DC]* to Amino acid *[AC]*.

Step3. Apply One Time Pad Step (AC, SK) to get Amino Acid for plaintext *[AP]*.

Step4. Transfer *[AP]* to DNA sequence using AMBIG to get [DP].

Step5. Transfer *[DP]* to binary form *[BP]* using the binary coding.

Step6. Convert *[BP]* to characters to get the plaintext *[P]*.

4. SECURITY ANALYSIS AND EXPERIMENTAL RESULTS

In this section the security analysis and the simulation results for the proposed scheme are discussed in details.

4.1 Security Analysis

The main characteristic that identifies any encryption algorithm is its ability to secure the protected data against attacks. The NIST statistical testing used to evaluate the secrecy of the proposed scheme. The NIST Test Suite is a statistical package consisting of 15 tests that were developed to test the randomness of binary sequences produced by either hardware or software. The NIST statistical testing as they appear in the tables for simulation results namely are; The Approximate Entropy Test, The Block Frequency Test, The Cumulative Sums Test, The Discrete Fourier Transform (DFT) Test, The Frequency Test, The Linear Complexity Test, Tests for the Longest-Run-of-Ones in Block, The Non Overlapping Template Matching Test, The Overlapping Template Matching Test, The Random Excursions Test, The Random Excursions Variant Test, The Binary Matrix Rank Test, The Runs Test, The Serial Test and The Universal Test. For each test we compute what so called P-value; this value used to determine whether the tested bit stream is random or not. For any bit stream to be random its P-value must be greater than 0.01. Very small P-values would support non-randomness for given measure that less than 0.01.

According to NIST statistical testing the more randomness binary sequence is the more secrecy it is, but as the final cipher here is presented in a DNA form this rule will be quietly different. Running NIST statistical testing on real DNA sequences; Eight DNA sequences downloaded from NCBI database and mention in Table2 proved that DNA isn't random [15]. So the evaluations here focused on proving that the final cipher behaves like any real DNA sequence not to prove that it is random. Table2 introduced the results of P-value for each real DNA sequence with the fifteen tests. Table3 introduced the results of applying NIST statistical tests on the final cipher generated from the proposed scheme for each message size. According to the similarities between the results from Table2 and Table3 we conclude that the final cipher is so close to any real DNA sequence. NIST statistical tests used to evaluate the robustness of the proposed scheme. It proved that the final cipher of the proposed scheme looks like any real DNA sequence, this ensure the secrecy of the proposed algorithm. Also it very difficult for anyone to differentiate between any real DNA sequence and the faked one resulting from the proposed scheme. Running NIST statistical testing with DNA sequence also defines new ranges for P-value of each test with DNA sequence. So dealing with DNA isn't similar to dealing with any other binary bit stream.

4.2 Experimental Results

This section describes the experimental results carried out to evaluate the performance of the proposed scheme. We use messages with various sizes to test the proposed scheme, the estimated storage size in Kilobytes. The experiments are conducted using Intel(R) Core(TM) i5-2430M CPU, 2.40 GHz,64 bit processor with 4 GB of RAM. The simulation program is compiled using NetBeans 7.1.1 for java windows application under windows7. The simulation results for the proposed scheme is basically depend on how much time it takes to encrypt messages with various sizes. Fig.1 illustrates experimental results and time taken to encrypt each piece of plaintext in milliseconds. Fig.1 also illustrates experimental results for other two DNA cryptographic algorithms. The first one is DNA based playfair algorithm [3] which enhanced the performance of the classical playfair algorithm using DNA computing.

The second is DNA based playfair and insertion algorithm which encrypt plaintext first and then hide it into a DNA reference sequence [14]. The results from Fig.1 show better performance for the three algorithms according to time taken to encrypt and hide data with various sizes. Table 4

summarizes these performance results in details. Time taken to encrypt messages using the proposed algorithm is less than that in [3] and [14]. According to these results DNA and Amino Acid encoding can be used to enhance the security level of other conventional cryptographic algorithms.

Table 2. Results of NIST testing with real DNA sequences

DNA/Test	NIST Tests														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
AC153526	0.00	0.00	0.00	0.00	0.00	0.87	0.00	0.00-0.59	0.00	0.00	0.00	0.23	0.00	0.00	0.00
AC166252	0.00	0.00	0.01-0.02	0.52	0.82	0.69	0.00	0.00-0.95	0.00	0.00	0.00	0.00	0.00	0.00	Error
AC167221	0.00	0.00	0.00	0.00	0.00	0.74	0.00	0.00-0.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00
AC168874	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.00-0.92	0.00	0.00	0.00	0.93	0.00	0.00	0.00
AC168897	0.00	0.00	0.00	0.00	0.00	0.39	0.00	0.00-0.77	0.00	0.00	0.00	0.31	0.00	0.00	0.00
AC168901	0.00	0.00	0.00	0.00	0.00	0.85	0.00	0.00-0.99	0.00	0.00	0.00	0.07	0.00	0.00	Error
AC168907	0.00	0.00	0.00	0.00	0.00	0.28	0.00	0.00-0.93	0.00	0.00	0.00	0.03	0.00	0.00	Error
AC168908	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00-0.59	0.00	0.00	0.00	0.23	0.00	0.00	0.00

Table 3. Results of NIST testing with the final DNA sequence result from the proposed algorithm

Msg/Test	NIST Tests														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1KB	0.00	0.00	0.00	0.07	0.00	0.60	0.00	0.00-0.99	0.04	0.00	0.00	0.00	0.00	0.00-0.76	Error
10KB	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00-0.99	0.00	0.00	0.00	0.00	0.00	0.00	Error
20KB	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00-0.96	0.00	0.00	0.00	0.00	0.00	0.00	Error
50KB	0.00	0.00	0.00	0.00	0.00	0.34	0.00	0.00-0.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100KB	0.00	0.00	0.00	0.00	0.00	0.19	0.00	0.00-0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00
150KB	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.00-0.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00
200KB	0.00	0.00	0.00	0.00	0.00	0.84	0.00	0.00-0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00
300KB	0.00	0.00	0.00	0.00	0.00	0.23	0.00	0.00-0.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 4. Performance results for the three encryption algorithms

Input size of plaintext (in KB)	The proposed scheme		DNA based playfair algorithm [3]		DNA based playfair and insertion algorithm [14]	
	Plaintext after removing spaces	**Total processing time (in milliseconds)**	**Plaintext after removing spaces**	**Total processing time (in milliseconds)**	**Plaintext**	**Total processing time (in milliseconds)**
1	853B	65.875	846B	15.625	1024	1.870
10	8599B	191.125	8124B	203.125	10340	52.671
20	17,070B	342.375	16599B	390.625	20480	210.298
50	41,833B	895.125	41781B	1562.500	51200	1619.648
100	83,752B	2224.25	83910B	5656.250	102400	7436.146
150	126,476B	4081.75	127098B	12906.25		
200	167,416B	7293.125				
300	251,801B	19190.0				

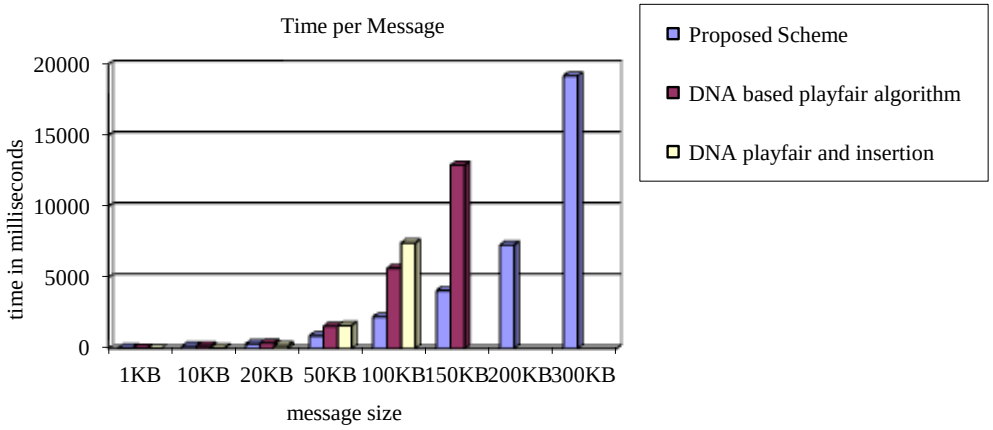

Fig. 1. Time per message

5. CONCLUSION

This paper proposed how DNA computing can be used to enhance the security level of classical OTP cipher. As the final cipher is presented in form of DNA sequence; this makes it difficult for intruder to detect whether or not there is a message hidden in these DNA sequence. Also the proposed scheme solved many limitations of OTP cipher which make it practically limited. The plaintext now can include any character not only English Alphabet. For the secret key it is generated using pseudo-random number generator, but as the proposed scheme pass through many transfers; this make it difficult for anyone to identify that key. Sometimes when encrypting long to messages some parts of the key need to be repeated to complete encryption process. This wasn't a problem as DNA and Amino Acid encoding encapsulate the processes of encryption and decryption. The proposed scheme ensured protecting the secure data against any attacks. Running NIST statistical testing with DNA proved practically that DNA isn't random. According to this fact a new ranges for the P-value of each test have been identified. These new ranges must be considered when testing any DNA cryptographic algorithm using NIST statistical testing.

6. REFERENCES

[1] Leonard Adleman,"Molecular Computation of Solutions to Combinatorial Problems", Science, 266:1021-1024, November 1994.

[2] Sherif T. Amin, , Magdy Saeb and Salah El-Gindi, "A DNA based Implementation of YAEA Encryption Algorithm", International Conference on Computational Intelligence (CI 2006), San Francisco, Nov. 20, 2006.

[3] Mona Sabry et al.,"A DNA and Amino Acids-Based Implementation of Playfair Cipher", International Journal of Computer Science and Information Security (IJCSIS), Vol. 8 No. 3, 2010.

[4] TAYLOR Clelland Catherine, Viviana Risca and Carter Bancroft,"Hiding Messages in DNA Microdots", Nature Magazine Vol. 399, June 10, 1999.

[5] H. J. Shiu, K. L. Ng, J. F. Fang, R. C. T. Lee and C. H. Huang, "Data hiding methods based upon DNA sequences", Information Sciences,vol.180, no.11, pp.2196-2208, 2010.

[6] Hongjun Liua, Da Lin and Abdurahman Kadir, "A novel data hiding method based on deoxyribonucleic acid coding", Computers and Electrical Engineering, vol. 39, pp.1164–1173, 2013.

[7] Ying-Hsuan Huang, Chin-Chen Chang and Chun-Yu Wu, "A DNA-based data hiding technique with low modification rates", Multimedia Tools and applications. Springer Science+Business Media, LLC 2012.

[8] Dominik Heider, Angelika Barnekow,"DNA-based watermarks using the DNA-Crypt algorithm", BMC Bioinformatics , Heider and Barnekow; licensee BioMed Central Ltd 2007.

[9] A. Gehani, T. LaBean and J. Reif," DNA-Based Cryptography", In: Jonoska, N., P˘aun, G., Rozenberg, G. (eds.) Aspects of Molecular Computing. LNCS, vol. 2950, pp. 167–188. Springer, Heidelberg 2003.

[10] J. Chen, "A DNA-Based Biomolecular Cryptography Design". In: 2003 IEEE International Symposium on Circuits and Systems, vol. 3, pp. 822–825 (2003).

[11] Miki Hirabayashi, Hiroaki Kojima and Kazuhiro 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.

[12] Sonia Dhull, Vinod Saroha, "Enhancing Security of One Time Pad Cipher by Double Columnar Transposition Method", International Journal of Advanced Research in Computer Science and Software Engineering,Volume 3, Issue 3, March 2013.

[13] William Stallings. "Cryptography and Network Security", Third Edition, Prentice Hall International, 2003.

[14] 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.

[15] Website, NCBI Database: <http://www.ncbi.nlm.nih.gov/>.