ENCRYPTION SYSTEM WITH INDEXING DNA CHROMOSOMES CRYPTOGRAPHIC ALGORITHM

Olga Tornea, Monica Borda, Tatiana Hodorogea, Mircea-Florin Vaida
The Faculty of Electronics, Telecommunications and Information Technologies,
Technical University of Cluj-Napoca, 26-28 George Baritiu Street
Cluj-Napoca, 400027, Romania
tornea.olga@yahoo.com, Monica.Borda@com.utcluj.ro, Tatiana.Hodorogea@com.utcluj.ro, Mircea.Vaida@com.utcluj.ro

ABSTRACT
DNA has a great cryptographic strength, its binding properties between nucleotides bases (A—T, C—G) offer the possibility to create self-assembly structures which are an efficient means of executing parallel molecular computations. Our work is based on the development of the new encryption system with indexing DNA chromosomes cryptographic algorithm using MATLAB Bioinformatics Toolbox. This toolbox offers support for developing DNA cryptographic operations and providing more secure cryptographic algorithms. The algorithm is based on the idea to use DNA chromosomes as one-time-pad structures and index them in order to encrypt the plaintext message. Our work is based on the complexity of the development of the new encryption system with indexing DNA chromosomes cryptographic algorithm.

KEY WORDS
DNA Encryption (DNAE) System, Central Dogma of Molecular Biology (CDMB), One-Time-Pad, Bio-nanotechnology, Indexing Chromosomes

1. Introduction

Why we need data security is already well-known. Do we need to find alternative, more secure encryption techniques for protecting sensitive data? With current network, Internet, and distributed systems, cryptography has become a key technology to ensure the security of today’s information infrastructure. A cryptographic system that an attacker is unable to penetrate even with access to infinite computing power is called unconditionally secure. The mathematics of such a system is based on information theory and probability theory. When an attacker is theoretically able to intrude, but it is computationally infeasible with available resources, the cryptographic system is said to be conditionally secure. The mathematics in such systems is based on computational complexity theory. To design a secure cryptographic system is a very challenging. A cryptographic system has one or more algorithms which implement a computational procedure by taking a variable input and generating a corresponding output. DNA cryptography is based on Adleman’s research of DNA computing [1]. Basic procedures of DNA OTP encryption schemes are given by [2]. DNA consists of two complementary strands. Each strand is made of a series of units called “nucleotides”. In this algorithm a DNA strand is meant as a place with indexes to the real message. The principle of indexing used in this algorithm is presented in [3]. If an algorithm's behaviour is completely determined by the input, it is called deterministic, and if its behaviour is not determined completely by input and generates different output each time executed with the same input, it is probabilistic. A distributed algorithm in which two or more entities take part is defined as a protocol including a set of communicational and computational steps. Each communicational step requires data to be transferred from one side to the other and each computational step may occur only on one side of the protocol. The goal of every cryptographer is to reduce the probability of a successful attack against the security of an encryption system – to zero. Probability theory provides the answer for this goal. Our work is based on the complexity of the development of the new encryption system with indexing DNA chromosomes cryptographic algorithm, an unconditionally secure and probabilistic DNAE System. DNAE is designed to have applications in textual and image information security.

2. The Encryption Protocol

Adleman began the new field of bio-molecular computing research. His idea was to use DNA biochemistry for solving problems that are impossible to solve by
conventional computers, or that require an enormous number of computation steps. The DNAE technique simulates the CDMB steps: transcription, splicing, and translation process. The time complexity of an attack on a message of length $n$, is $O(2^n)$. DNA computing takes advantages of combinatorial properties of DNA for massively-parallel computation.

In our work we used a cryptosystem with symmetric key named One-Time-Pad (OTP). It is an algorithm where each key is used just once where from the name of One-Time-Pad. OTP encryption uses a large non-repeating set of truly random key letters. Each pad is used exactly once, for only one message. The sender encrypts the message and then destroys the used pad. As it is a symmetric key cryptosystem, the receiver has an identical pad and uses it for decryption. The receiver destroys the same pad after decrypting the message. New message means new key letters. A ciphertext message is equally likely to correspond to any possible plaintext message. Cryptosystems which use a secret random OTP are known to be perfectly secure [4].

Introducing DNA into the common symmetric key cryptography, it is possible to follow the pattern of symmetric key cryptosystem, while also exploiting the inherent massively-parallel computing properties and storage capacity of DNA in order to perform the encryption and decryption using OTP keys [5]. The resulting encryption algorithm which uses DNA medium is much more complex than the one used by conventional encryption methods.

We developed an encryption algorithm which uses OTP as symmetric key and real chromosomal sequence as OTP. We extracted chromosomal sequence from public available data bases [6] and used it for implementation of this algorithm (Figure 1 and 2).

---

**3. Message Encryption**

Plaintext message was transformed in bits and after that in DNA format. We used a text message for encryption so an encryption unit was a character and in ASCII cod it was represented on 7 bits, or in case it was an image a pixel was an encryption unit and it could be represented on 8 bits at least. Transformation from a 2-letter (0, 1) alphabet to a 4-letter (A, C, G and T) alphabet was done using 2 bits to represent a letter:

- A – 00
- C – 01
- G – 10
- T – 11

Using this substitution a character was represented on 4 letters which is equivalent to a byte.

Using Matlab functions we obtained decimal ASCII cods of the plaintext message, and transformed them in binary form, each character on 8 bits. After that, using functions from Bioinformatics Toolbox we transformed our message from binary to DNA alphabet.

Each character was transformed in a 4-letter DNA sequence and searched in the chromosomal sequence, used as OTP. We extracted chromosomal sequence from public available data bases [6] and used it for implementation of this algorithm (Figure 1 and 2).
Figure 3 Exemplification of the OTP scanning process for message encryption

For each character was obtained an array of indexes in chromosomal sequence. Number of indexes for a character depends on how often the character’s DNA sequence retrieved in the chromosomal sequence. For each character was chosen a random index from its array of indexes. We obtained the final encrypted message: an array of random indexes, one for each character. Below is presented example of the message encryption with implementation results.

Example of implementation results:

Message: “secret”
ASCII cods: 115 101 99 114 101 116

For each character was chosen a random index from its array of indexes using Matlab function. Below are established positions of random indexes inside character’s arrays:

115 70th index 23811
101 26th index 13981
99 7th index 8011
114 57th index 21195
101 57th index 32741
116 158th index 25264

Final encrypted message is:
23811 13981 8011 21195 32741 25264

4. Message Decryption

At message decryption is used the same OTP as at encryption, because it is a symmetric key algorithm. The key is Homo sapiens FOSMID clone ABC14-50190700J6, from chromosome x complete sequence. First we read this sequence using functions from Bioinformatics Toolbox:

```
FASTAData = fastaread('homo_sapiensFosmid_clone.fasta')
```

Each index from received encrypted message was used to point in chromosomal sequence:

```
SeqNT=FASTAData.Sequence(i:i+3)
```

Using these pointers we extracted for each character a 4-bases DNA sequence. This variable was transformed in numerical value, using transformation offered by Matlab Bioinformatics Toolbox (A-1, C-2, G-3, T-4). As transformation starts with 1, at encryption to each digit was added a unit and after that it was transformed in base (example, “00” binary → 0 digit → 0+1 → A). At decryption from each obtained digit was subtracted a unit and after that transformed in 2 bits:

```
Example:
CCC (bases) → 2221(digits) → 2-1, 2-1, 2-1, 1-1 → 1110 (digits) → 01 01 01 00 (bits)
```

Obtained binary numbers are the ASCII cods of the recovered message characters.
5. Complexity of Secure DNA Encryption System

With an OTP, an adversary has no information about how to cryptanalyze the ciphertext, since every key is equally likely to correspond. Cryptosystems which use a secret random OTP are known to be perfectly secure and are applicable primarily for transmission of ultra-secure information. The problem of this cryptosystem is that the key letters have to be truly random and the key sequence can not be reused ever again.

Here is where the advantages of using DNA in cryptography came. We used at implementation as the key a human chromosome, but any genetic sequences, of any living matter can be used as a secret key for encryption. We can exploit great storing capabilities and variety of DNA sequences for the usual OTP cryptosystem.

This encryption algorithm treats also the problem of the vulnerability to frequency attack. For the same character from the plaintext message we obtained a number of different indexes which are used as values in the ciphertext by a random choice. This solve the problem of frequency distribution of letters in a ciphered message [7]. The same character, for instance “e” will appear in ciphertext as 13981, or 32741, or any other index which was found for this character in the chromosome. Another advantage of this algorithm is that at encryption of another message, indexes for each character will be different from the previous encryptions values. The same character will appear in ciphertext under different values at encryption of two different messages. Chromosome or any DNA sequence which was chosen to be the encryption key dictates what kind and how many indexes for ciphertext will have a character.

6. Conclusion

Based on the ideas presented in [1], [2], and [3] an original DNA cryptographic algorithm was performed. We use one-time-pad principle and DNA chromosomes storing capabilities for message encryption. Implementation was performed in Matlab using Bioinformatics Toolbox and genetic database maintained by NCBI. One single chromosome from any species is composed from thousand of bases and it is perfect to be used in this algorithm to address the characters from the plaintext. Each character from the plaintext message was transformed into a unique sequence of 4 bases and searched in the chromosome, used as OTP. A random index of a character in chromosome becomes part of the ciphertext. The strength of this algorithm is based on the secrecy of the OTP and protection from frequency attack. The aim of this paper is to find useful and practical DNA cryptographic algorithm and to study its applicability in DNA technology. Laboratory implementations are possible (microarray technology [8]), but are still expensive and time consuming. Despite of this, simple and effective algorithms are necessary in order to bring DNA computing on digital level and use it on a large scale.

Acknowledgements

This paper was funded by the Romanian Agency UEFISCU within the PN II, IDEI no. 909/2007 research grant. Research Project PN2, “Alternative Security Technologies for Network Applications”- supported by the National Authority of Romania.

References