COMPLEXITY OF DNA ENCRYPTION SYSTEM AS A SUBSET OF JAVA CRYPTOGRAPHY EXTENSION

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

ABSTRACT
Java Cryptographic Extension (JCE) offers support for developing cryptographic package providers, allowing us to extend the JCE by implementing faster or more secure cryptographic algorithms. By the same means we shall provide our independent implementation of a DNA Encryption (DNAE) system, based on the Central Dogma of Molecular Biology (CDMB).
In this work we present a technical process for protecting data assets such as personal medical information using a DNA cryptography technique in which a person's own blood mineral levels serve as a seed for selecting, transmitting, and recovering his sensitive personal data. As we know that the management of security keys remains a challenge, we also propose a mechanism to generate encrypt-decrypt keys by taking into consideration specifics of the cryptography method and the individual's blood analysis.
Our work is based on the complexity of developing, as a subset of JCE, an unconditionally secure DNAE System as part of our security provider, named DNAProvider. We intend to use DNA Provider with the DNAE System in medical applications to ensure security of medical information

KEY WORDS
DNA Encryption (DNAE) system, Central Dogma of Molecular Biology (CDMB), probabilistic encryption.

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. If an algorithm's behavior is completely determined by the input, it is called *deterministic*, and if its behavior 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 developing an unconditionally-secure DNA Encryption System as part of DNA Provider. We aim to use DNA Provider with unconditional secure DNAE system in medical applications to ensure security of medical information. However, DNA Provider with DNAE will also have applications in many aspects of today's web-based business processes such as e-commerce, Internet banking, and email.

2. Java Cryptography Extension Architectural Model with DNA Encryption Security Provider

The goal of the security provider interface is to allow a means whereby specific algorithm implementations can be substituted for the default provider, SUN JCE. JCE was developed as an extension package which includes implementation for cryptographic services. JCE offers a provider implementation plus API packages providing support for key agreement, encryption, decryption and secret key generation. Thus, JCE offers support for

developing alternative cryptographic package providers. This support allows us to provide our independent implementation of DNAE System, based on the of the CDMB [11].

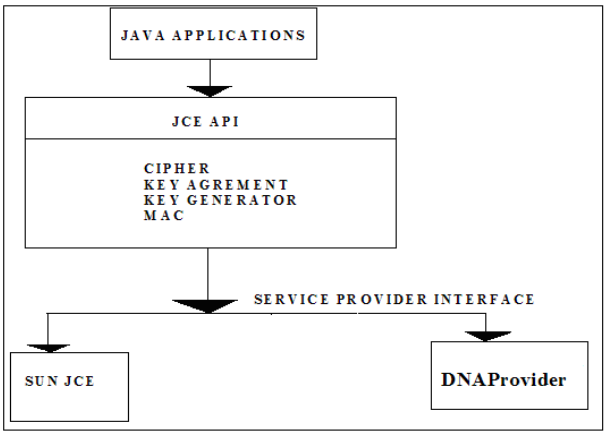

Fig. 1 Java Cryptography Extensions architectural model with unconditional secure DNA Encryption as part of our security provider (DNAProvider)

The application code calls the appropriate JCE API classes. The JCE API classes invoke the classes in a provider that implements the interface classes, JCE SPI. The JCE SPI classes, in turn, invoke the the requested functionality of the DNA Provider (Fig.2).

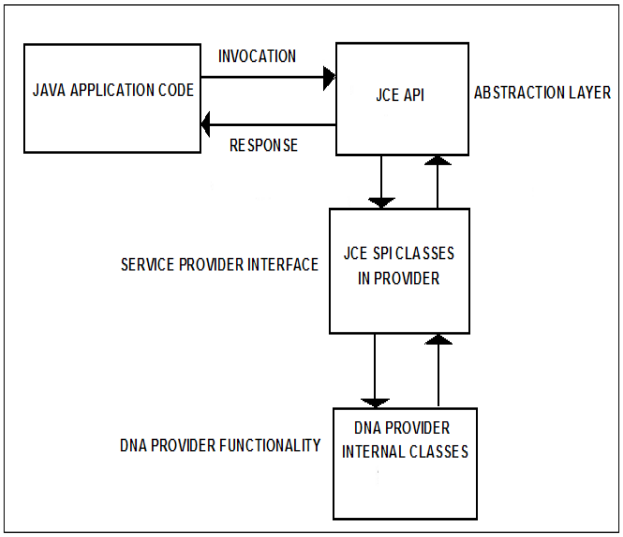

Fig. 2 Invocation of DNAProvider for providing requested functionality

When the Java Virtual Machine starts execution, it examines the user's properties to determine which security providers should be used. The user's properties are located in the file *java.security*, in which each provider is also enumerated. If users prefer to use DNAProvider as an additional security provider they can edit this file and add the DNA Provider. When the Security Class is asked to provide a particular engine and algorithm, it searches the listed providers for the first that can supply the desired operation.

3. Creating the DNA Security Provider with DNA Encryption

An Engine Class defines an abstract cryptographic service, without its concrete implementation. JCE 1.2.2 is provided as an optional package and adds engine classes such as: Cipher, KeyAgreement, KeyGenerator, MAC, and SecretKeyFactory. The application interfaces given by an engine class are implemented and referred to as the *Service Provider Interface*. We wrote the service implementation code and implemented the master class, which is a subclass of DNAProvider. As a next step, we prepared to request a code-signing certificate from SUN for testing. This code-signing certificate will be used to sign DNAProvider. The certificate is valid for five years for both testing and production. Prior to requesting the certificate we will continue work on the complexity of DNA Encryption Protocol, as well as the theoretical and practical possibilities it offers. To provide secure communications based on unconditionally-secure DNA Cryptography, we apply the ramifications of information theory and probability theory. Unconditional security, in a theoretical sense, can only be achieved if the entropy of the secret key is greater than or equal to the entropy of the plain-text message, [8]. Information theory and probability theory continue to have a deep impact on modern cryptography; for example, Quantum Cryptography applies the Heisenberg uncertainty principle of quantum physics to provide a secure channel.

3.1. Data Hiding in DNA

Recent research considers the use of the Human genome in cryptography. In 2000, the Junior Nobel Prize was awarded to a Romanian-American student, Viviana Risca, for her work in DNA steganography [10]. A DNA-encoded message is first camouflaged within the enormous complexity of human genomic DNA, and then further concealed by confining this sample to a microdot. A prototypical 'secret message' DNA strand contains an encoded message flanked by polymerase chain reaction (PCR) primer sequences. Denatured human DNA provides a very complex background for concealing a secret-message.

Risca, knowing both the secret-message DNA, PCR primer sequences and the encryption key could readily amplify the DNA and then proposed a mechanism to read and decode the message. We propose as a first step to encode the medical records of an individual in DNA data strand flanked by unique primer sequences, which we obtain in the process of deriving a DNA secret key from blood analysis [9]. The specific mineral levels and their deviation from normal values are considered as a first step. We then mix the message-encoded DNA strand

among other decoy DNA strands that will together be sent to a receiver through a public channel.

3.2. Biological Principles

Is DNA is a Genetic Program? This is a question posed by Rupert Sheldrake, a theoretical biologist that extends the Jung concepts from the collective human level to the biological and mineral levels, generally, to the whole universe [13]. August Weismann's theory of the germ-plasma (end of 19th century) is the ancestor of the present idea of genetic programming. The genetic program is assumed to be identical with DNA, the genetic biochemistry. The genetic information is coded in DNA and this code forms the genetic program.

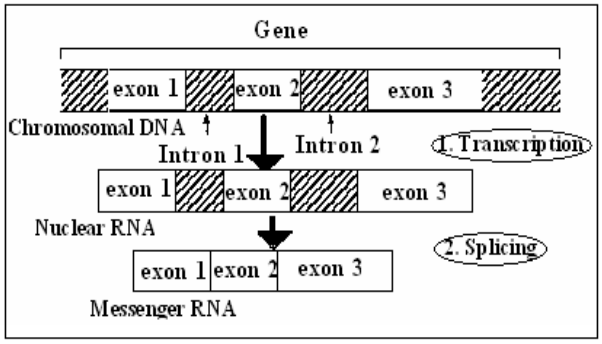

Fig. 4 Central Dogma of Molecular Biology

A DNA segment that constitutes a gene is read, starting from the promoter (starting position) of the DNA segment. The non-coding areas (introns) are removed according to certain tags. The remaining coding areas (extrons) are rejoined and capped. Then the sequence is transcribed into a single stranded sequence of mRNA (messenger RNA). The mRNA moves from the nucleus into the cytoplasm. In chromosomes, DNA acts as a template for the synthesis of RNA in a process called transcription. During RNA Synthesis and Processing in the transcription and splicing steps, introns are excised and extrons are retained to form mRNA, which will perform the translation work.

In the translation process, codons are translated into the amino acids according to the genetic code. The DNA form of information is scanned by a hypothetical operator, Stefani, to find the locations of the introns, which she then records [15]. She cuts out the introns according to the specified pattern so that the DNA form of data is translated into the mRNA form. The mRNA form then translates into the protein form of data according to the genetic code table (64 codons to 20 amino acids).

4. The DNA 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.

Introducing DNA cryptography into the common PKI scenario, it is possible to follow the pattern of PKI, while also exploiting the inherent massively-parallel computing properties of DNA bonding to perform the encryption and decryption of the public and private keys [6]. The resulting encryption algorithm used in the transaction is much more complex than the one used by conventional encryption methods.

To put this into the common description of secure data transmission and reception with respect to DNA cryptography, let us say Stefani is the sender, and Otto, the receiver. Stefani provides Otto her public key which will comprise someone's unique blood analysis [7]. The Public Key (PK) encryption technique splits the key into a public key for encryption and a secret key for decryption. As an example: Otto generates a pair of keys and publishes his public key, while only he knows his secret key. Thus, anyone can use Otto's public key to send him an encrypted message, but only Otto knows the secret key to decrypt it [16].

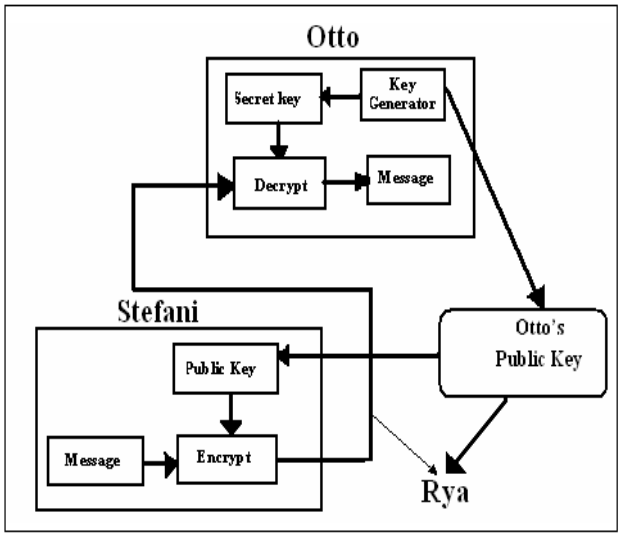

Fig. 5 Public Key Encryption

A secret DNA data strand contains three parts: a secret DNA data strand in the middle, and unique primer sequences on each side S1. Stefani uses the technique of deriving DNA private key from blood analysis. In this process, Stefani uses a program that associates a specific mineral to the nucleotide sequence based on the someone's medical results, which will constitute the unique primer sequences S1. Using an information conversion program, Stefani encodes the medical records in a DNA data strand flanked by unique primer sequences S1 and mixes it among other decoy DNA strands. According to the CDMB, during the process of transcription, Stefani removes the introns from the data-encoded DNA, resulting in encryption key 1, E1 (starting

and pattern codes of introns). Thus, E1 => C1 = E1(P), where P is plain-text and C is the cipher-text. Stefani translates the resulting spliced form of the data from which she derives Encryption key 2, E2 (codon-amino acid mapping). E2 => C = E2(C1) obtains the data-encoded protein after the translation process. Stefani sends Otto the keys E1 and E2 through a public channel. Then she sends Otto the encoded protein form of the data through a public channel. Otto uses the key E2 to recover the mRNA form of the data from the protein form of the data. Decryption key, D1 = E2 => P1=D1(C). Otto recovers the DNA form of the data in the reverse order that Stefani encrypted it. Decryption key, D2 = E1 => P = D2(P1). Otto identifies the secret data-carrying DNA strand using the program that associates the nucleotide sequence based on someone's blood mineral analysis. He obtains the unique primer sequences S1 that mark the beginning and end of the secret data DNA strand hidden among the decoy strands. In this last step, Otto uses the information conversion program and reads the medical record of the individual.

4.1 The Technique of Deriving DNA Private Keys from Blood Analysis

Considering the genetic studies of the French researcher Etienne Guillé, it is possible to identify the nucleotides sequences that correspond to some minerals. An example is the Gold sequence [2]. The minerals are able to influence the DNA by conformational changes and also by activating or inhibiting DNA coding sequences, (Table 1).

Table 1. Mineral-Nucleotide Correspondence Table

Mineral	Nucleotide Sequence
Fe	AT AG AC GG AA
Au	GA AT AG AC GC AA

As blood analysis results are specific for each person (Table 2), we can associate a mineral such as Calcium, Magnesium, etc., from the medical result, to a nucleotide sequence based on its concentration level. This nucleotide sequence based on the medical results of the specific person will constitute the unique primer sequences. However, as we generate a sequence for calcium, an intruder (Rya), knowing the calcium level, could possibly discover the nucleotide sequence.

Table 2 Example Blood Mineral Analysis

Mineral	Blood Analysis Result	Normal Level
Ca	2.81 mmol/l	2.25-2.7
Mg	0.89 mmo/l	0.75-1.05

Therefore, as an additional layer of security, we propose to associate to each measured mineral a corresponding mineral – which we will call here, a synergetic mineral pair. For example: Ca-Fe, (Table 3) or other special minerals established by the encryption process.

Table 3 Paired mineral table

Mineral	Paired synergetic minerals
Ca	Fe, Au, P
Mg	K, Ca, Na,

Then, we will substitute the nucleotide sequence for calcium with that of iron. For our purposes, this step serves as an intermediate substitution table for increased obscurity. Thus a person's data-carrying DNA strand will be flanked by primer sequences unique to that individual. We will do the association in such way that from every most recent blood analysis results, a new primer sequence will be generated. After generating the unique primer sequence, an n-base primer will result. As we know that the management of private keys remains a challenge, we will use each unique blood analysis as the basis for a secret key generation mechanism. The medical results will be of no use to an unauthorized person, and for an intruder, it would prove extremely difficult to read and detect the DNA strand that contains someone's medical history, without knowing the specific unique primer sequences of the specific person. Performing a quick search by a program we will get the chosen mineral (M).
Considering a dedicated medical application with security facilities developed by our research team, we are able to incorporate the blood analysis results of an individual into the security process using the following algorithm:
If the mineral level (L) does not correspond to a normal level (NL) and is equal to value: L=X.YZ mmol/l (1)
Then:
1st step: We associate X times the nucleotide sequence corresponding to the synergetic mineral of the selected mineral, obtaining a sequence S1.
2nd step: In the second step, to S1 we add Y+Z numbers of nucleotide from the original sequence of a synergetic mineral, resulting sequence S that will constitute the unique primer sequence.
Otherwise:
If all blood results are in a normal level we will choose a minimal encoding to generate the unique primer sequences because we don't have to increase the hiding process for an individual of normal health. In this case we will just associate the nucleotide sequence corresponding to a chosen mineral.

4.2 Complexity of Secure DNA Encryption System Based on Probabilistic Theory

Given $n \in N$ as a composite positive integer and $x \in Z^*n$, the Quadratic Residuosity Problem (QRP) is the problem of deciding whether $x \in QR_n$. If an arbitrary element of J_n *is* given, it is computationally difficult to decide whether it is a square or it is a pseudo-square modulo *n*.

Probabilistic DNA encryption can exploit this computational difficulty [3].

The DNA Encrypt algorithm based on CDMB, employed by probabilistic DNA encryption, must specify how a k-bit plain-text message $m = m_1 m_2 \ldots m_k$ is DNA-encrypted according to CDMB, so that only Otto, the recipient (or a person holding the Otto's private key), is able to decrypt it. The DNA Encryption algorithm needs to be a probabilistic one, to take as input, a public key (n, y) and a message m, generating the DNA cipher-text c as output. For every message bit m_i (i= 1, ... , k), the DNA Encrypt algorithm needs to choose $x_i \in R$ Z*n and compute: $c_i \equiv x^{2i} (\text{mod } n)$ if $m_i = 0$ and $y x^{2i} (\text{mod } n)$ if $m_i = 1$. In both cases, each message bit m_i must be DNA-encrypted with an element of Z* n. The resulting DNA cipher-text c is the k-tuple $c = (c_1, \ldots , c_k)$. For DNA c_i, the DNA decryption algorithm needs to evaluate $e_i = c_i / p_i$ and compute $m_i = 0$ if $e_i = 1$, otherwise, $m_i = 1$.

As the last step, the plain-text message m is set to $m = m_1 m_2 \ldots m_k$. The DNA Probabilistic encryption is semantically secure.

Probabilistic encryption as proposed by Goldwasser and Micali, is the best notion of security we currently have for asymmetric encryption systems [4].

From a theoretical point of view of the complexity in DNA Probabilistic encryption, there is not much to add. From a practical point of view, the complexity in DNA probabilistic encryption based on CDMB includes the fact that every plain-text message bit m_i is encrypted with an element of Z* n, resulting in a considerable message expansion. We need to improve the minimization of message expansion to a constant number of bits.

5. Conclusion

Nowadays, the software applications dedicated to different domains must respect many security elements. Considering a dedicated medical application (for alternative therapy) developed by our research team [16], we are attempting to adapt standard classical and new security facilities, including DNA cryptography. The DNA technology will be used to implement security facilities for software applications [17], including the use of blood analyses to generate encryption keys. We hope to find that our DNA Encryption System with DNA Cryptography, based on CDMB, probability theory, and information theory, implemented as a subset of the Java Cryptography Extension, may prove to be an unconditionally secure encryption system.

To improve the security coding technique we will consider a hybrid numerical association to the letters that compose the biological PCR strands.

An interesting mechanism is presented by Gregg Braden and may be used in a dedicated implementation in the future, [18].

Acknowledgements

The research activity will be supported by PN2 IDEI project, no. 1083, from National Authority for Scientific Research that will be realised during 3 years (2007-2010). The new idea that are presented in the paper comes from the spirituality domain [17] and I would like to express my acknowledgements to all "humans" that gave me the power to choose the spiritual direction in conjunction with my technical affinities.

References

[1.] Boneh, D., "Twenty Years of Attacks on the RSA Cryptosystem," Notices of the American Mathematical Society (AMS), Vol. 46, No. 2, 1999, pp. 203–213.

[2.] Bivolaru Gregorian, Enciclopedia naturista a elementelor minerale, Editura Shambala, 2003 (Romanian language)

[3.] Boneh, D., and H. Shacham, "Fast Variants of RSA," *CryptoBytes*, Vol. 5, No. 1, 2002

[4.] Goldwasser, S., and S. Micali, "Probabilistic Encryption," Journal of Computer and System Sciences, Vol. 28, No. 2, April 1984,

[5.] Garfinkel Simson, Web Security, Privacy & Commerce, 2nd Edition, O'Reilly Publisher, November 2001

[6.] Gehani Ashish; La Bean, Thomas H.; Reif, John H, "DNA-Based Cryptography," Department of Computer Science, Duke University, June 1999

[7.] Tatiana Hodorogea, Mircea-Florin Vaida, Alternate Cryptography Techniques, ICCC05, Miskolc-Lillafured, Hungary, 24-27 May 2005, Vol. 1, pp. 513-518

[8.] Shoup, V., "OAEP Reconsidered," *Proceedings of CRYPT0 '01*, Springer-Verlag, LNCS 2139,2001

[9.] Tatiana Hodorogea, Mircea-Florin Vaida, Blood Analysis as biometric selection of Public Keys, 7th International Carpathian Control Conference ICCC'2006, Ostrava – Beskydy, Czech Republic, May 29-31, 2006, pp. 675-678

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

[11.] Tatiana Hodorogea, Mircea-Florin Vaida, Security Provider with DNA cryptography algorithm as a subset of Java Security API, ICCC´2007, Slovakia,Technical University of Kosice, 24-27 may 2007.

[12.] Kahn D., The Codebrakers, McMillan, New York, 1967

[13.] Sheldrake Rupert, Mind, Memory, and Archetype: Morphic Resonance and the Collective Unconscious, web Sheldrake published papers

[14.] C. Striletchi, M. Vaida, Enhancing the Security of Web Applications, Information Technology Interfaces, ITI 2003, Proceedings of the 25th

International Conference on Knowledge Management and E-Commerce, pp. 463-468, June 2003, Cavtat, Croatia

[15.] Pointcheval, D., "How to Encrypt Properly with RSA," *CryptoBytes*, Vol. 5, No. 1, 2002, Norwood, MA, 2003.

[16.] Vaida Mircea-Florin, Tatiana Hodorogea, Coding Documents using Alternative Techniques, CSNDSP 2006 PROCEEDINGS, July 2006, Patras, Greece, pp.359-361

[17.] Vaida Mircea-Florin, Teaching Computers as a Human Spiritual Evolution, the 4th IASTED International Conference on Web-Based Education, WBE2005, February 21-23, 2005, Grindelwald, Switzerland, pp. 667-672

[18.] Gregg Braden, "Codul lui Dumnezeu: secretul trecutului, promisiunea viitorului" (The God Code), Ed. For You, Bucureşti, 2005