15

Modern Technologies Used for Security of Software Applications

Tatiana Hodorogea[1] and Ionas Szilard Otto[2]
[1]*University of Basel,*
[2]*Bogdan-Voda University, Cluj-Napoca,*
[1]*Switzerland,*
[2]*Romania*

1. Introduction

Nowadays information systems security services involve more complexity because of there heterogeneity involving very big threats and attacks on such kind of networks, which are widely spread, open and interconnected. The security attacks and the technologies to exploit security attacks are growing continuously.

The importance of providing and maintaining the data and information security across networks is a major enterprise business activity, resulting in a big demand and need to ensure and maintain information security.

Cryptographic algorithms for confidentiality and authentication play a major importance role in nowadays information security.

With current network, Internet, and distributed systems, cryptography has become a key technology to ensure the security of today's web-Software Applications. 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. The goal of every cryptographer is to reduce the probability of a successful attack against the security of an encryption system – to zero and the probability theory provides the answer for this goal.

The aim and objective of this chapter is the development of a DNA Cryptographic Keys Based on Evolutionary Models, for the integration in our DNAProvider as Java Cryptographic Extension (JCE) with DNA Encryption (DNAE) system for use in security of our developed Web-based Software Applications.

Java Cryptography Extension (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. JCE offers a provider implementation plus API packages providing support for key agreement, encryption, decryption and secret key generation. The security provider interface the means by which different security implementations may be plugged into the security package as message digests, encryption, digital signatures and keys, through JCE, JSSE and authentication through JAAS. Thus, JCE support allowed us to provide our independent implementation of DNA Cryptographic Keys Based on Evolutionary Models used for Security of Web-based Business Processes.

As Public-Key algorithms are based on mathematical functions rather than on substitution and permutation involving the use of two separate keys, in contrast to symmetric encryption, which uses only one key we developed, implemented and tested the Security System Software Applications based on the Central Dogma of Molecular Biology (CDMB), where we derived DNA Cryptographic Keys based on evolutionary models. Our cryptographic system has one or more algorithms which implements 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*.

Our work was based on the complexity of developing, as a subset of JCE, an unconditionally secure DNAE System as part of our security provider, named DNAProvider, (Hodorogea, Ionas, 2011).

Our work is based on Deriving DNA Cryptographic Keys Based on Evolutionary Models for Security of Software Applications.

Biotechnological Methods as recombinant DNA have been developed for a wide class of operations on DNA and RNA strands.

When aligning the DNA sequences of the same gene from related species, there will usually be differences between the sequences because of evolution and because of the degeneracy of the genetic code. Based on evolutionary models we extract and align the DNA sequences of the same gene from related chosen species with respect to human DNA Sequences. The alignment in the evolutionary system pipeline is realized with ProbCons tool, which is a pair-hidden Markov model-based on progressive alignment algorithm that primarily differs from most typical approaches in its use of maximum expected accuracy. After aligning our extracted DNA Sequences with ProbCons we derive the private/public pair DNA cryptographic keys based on evolutionary models mathematical functions. The molecular evolution model assigns probabilities to multiple-alignment columns in terms of the the philogenetic tree branches and is time dependent of frequency selections. Based on Kimura-Ohta theory Halpern and Bruno who have shown that mutation limit can be determined by substitution rates in terms of the mutation rates and equilibrium frequencies. Our work described in this chapter was based on the complexity of deriving DNA Cryptographic Keys Based on Evolutionary Models for Security of Software Applications.

2. Data security and cryptography

Networks are based on a number of network level equipments and servers as:

- **Dynamic host configuration protocol (DHCP)**, server dynamically assigns an IP address.
- **Domain name system (DNS),** server translates a domain name (URL) into an IP address.

- **Network address translation (NAT**), performs translation between private and public addresses.
- **E-mail server** supports electronic mailing
- Internet/Intranet/Extranet Web servers
- **Access points (AP),** giving wireless equipments access to wired network.
- **Virtual LAN (VLAN)** which virtually separate flows over the same physical network, so that direct communications between equipments from different VLANs could be restricted and required to go through a router for filtering purposes
- **Network access server (NAS) / Broadband access server (BAS),** gateways between the switched phone network and an IP-based network
- **Intrusion detection system (IDS) / Intrusion prevention system (IPS)** used to detect intrusions based on known intrusion scenario signatures.

More than 20 years information security considers confidentiality, integrity and availability, known as CIA as the base of information security. Cryptography gives us all of these services, linked with transmitted or stored data.

Considering GRID computing security where the heterogeneous resources are shared and located in different places belonging to different administrative domains over a heterogeneous network, additional security requirements must be satisfied compare to classical network security.

A GRID is a software toolbox and provides services for managing distributed software resources. Securing information in GRID computing encompasses verifying the integrity of the message against malicious modification, authenticating the source of a message and assuring the confidentiality of the message being sent.

The key points of information security are:

- Confidentiality- keeping the data secret
- Integrity - keeping the data unmodified
- Authentication - certifies the source of the data
- Non-repudiation - the process of the sent data can’t be negated

Confidentiality implies the prevention to disclosure information by individuals and unauthorized systems.

In information security integrity implies the impossibility of data modification without the authorization and keeping the data unchanged. Authentication means the knowledge of the source, from where the data was received.

The information needs to be available when necessary. The assurance of availability implies the prevention of denial of service attacks.

Communication between GRID entities must be secure and confidentiality must be ensured for sensitive data, from communication stage, to potential storage stage. Problems of integrity should be detected in order to avoid treatment faults, availability is directly linked to performance and cost in GRID environment.

Cryptographic algorithms for confidentiality and authentication play a major importance role in nowadays information security.

2.1 Security services, threats and attacks

The main threats to WLAN networks are the radio waves since the radio waves broadcast, without respect to neither walls nor other limit. Denial of service (DoS) makes the network ineffective. It is easy to jam a radio network and network becomes unusable. By the use of rush access the network is overloaded with malicious connection request. Tools are able to detect this kind of traffic and help network administrator to identify and locate the origin.

Intrusions threats are most common attacks where the intrusion is done via client station and protection is the same as for wired networks, the use of firewall.

The most critical attack that aims to take the control of network resources of the enterprise is the network intrusion and in this case Wi-Fi dedicated intrusion detection systems (IDS) are efficient against such attacks.

With falsification of access points the hacker fetches the traffic on the network and the security protection from such attacks is by detecting abnormal radio transmission in unexpected areas.

Security protections can be applied to WLAN: network monitoring is a good defense to observe the network to be informed if something strange happens.

The intrusion detection system (IDS) is used against network intrusions. IDS correlates suspect events, tries to determine if they are due to an intrusion.

Traffic monitoring prevents against spoofing due to permanence observing of the Wi-Fi traffic in order to detect any inconsistent situations.

Network engineering is another security mechanism for network protection. It is strongly recommended to deploy WLAN using switches instead hubs and to control the traffic between wired networks. WLAN dedicated switch manages radio, networking and security functions and access points are used only as emitters and receptors providing a better protection against attacks. The firewalls manage protections at addressing level by providing filters and log connections, managing access control list (ACL) which are used for access filtering and monitor the connections. The firewalls must be installed in a DMZ, VPN authentication with encryption mechanisms activated. The use of VLAN must be done in order to split the network for the isolation of strategic data from the radio network. For this VLAN must be deployed on a dedicated virtual LAN structure where network contains several VLANs and each associated to a WLAN subnet with own SSID. All VLANs must be connected on the WLAN switch.

Encryption is the security mechanism at the application level by its use if the information is intercepted is unusable. In this scope standard protocols like transport layer security (TLS) may be used. Authentication is done by a login password sequence and link between client and server is secured by TLS, authentication is done via a local authentication database.

MAC addresses filtering is a non cryptographic security feature uses the unique link layer (MAC) address of the WLAN network card and identifies legitimates users.

One of security feature based on cryptography is wired equivalent privacy (WEP), defined in the initial IEEE 802.11 standard and provides authentication and encryption with 40-128 bit key length. The key should be changed in all nodes and in the access points, frequently and simultaneously.

Because of WEP weakness, the IEEE designed a protocol named 802.11i, known as WPA 2 (Wi-Fi Protected Access 2). Temporal key integrity protocol (TKIP) is used for generating per-packet keys for the RC4 ciphering used. The key is called temporal because is changed frequently and is combined with the sender's MAC address using the exclusive OR-operation. Resulting in the usage of different keys for upstream and downstream transmissions,

Two types of security mechanisms are known: first type is the one which are implemented in a certain protocol layer. Second type of the security mechanisms are not related to protocol layers or any security services.

Cryptography encrypts the data by the mean of using encryption security mechanism.

Encryption security mechanism is an encryption algorithm which encrypts and decrypts the data, transforming it into unreadable format.

The encryption mechanism depends on encryption keys being used (zero or more) and encryption algorithm. After the readable data is cryptographically transformed, digital signature is appended to it as a second security mechanism. *Digital signature security mechanism* proves the integrity of the data, the source of the data and protects the information send against forgery.

The access right to information and resources is realized thought the third security mechanism known as: *access control security mechanism*.

For preventing traffic analysis attempts the bits are inserted into the gaps of the data stream and this constitutes the *traffic padding security mechanism*.

Data security model represents a secure transfer of information across information channel (internet), between two principals: sender and receiver, by the use of communication protocols. Data security model implies the protection of data against confidentiality and authentication threats coming from an opponent. Security related transformation is needed to satisfy these conditions of data protection during transfer through information channel. Encryption transforms the message in an unreadable format, by the opponent. The additional code is added to the secret information based on the content of the message and this way the identity of the sender is verified.

3. DNA cryptography model

With current network, Internet, and distributed systems, cryptography has become a key technology to ensure the security of today's information infrastructure.

Biotechnological Methods as recombinant DNA have been developed for a wide class of operations on DNA and RNA strands. Bio Molecular Computation (BMC) makes use of biotechnological methods for doing computation and splicing operations allow for universal computation.

The first applications of DNA-based cryptography systems using biotechnologies techniques included: methods for 2D data input and output by use of chip-based DNA micro-array technology and transformation between conventional binary storage media via (photo-sensitive and/or photo emitting) DNA chip arrays

Lately DNA Cryptosystem using substitution and biotechnologies have been developed: *Substitution* one-time-pad encryption: is a substitution method using libraries of distinct pads, each of which defines a specific, randomly generated, pair-wise mapping. The decryption is done by similar methods. The *Input is a* plaintext binary message of length n, partitioned into plaintext words of fixed length.

Substitution One-time-pad, a table randomly mapping all possible strings of plaintext words into cipher words of fixed length, such that there is a unique reverse mapping and the e*ncryption is done by* substituting each i-th block of the plaintext with the cipher word given by the table, and is decrypted by reversing these substitutions. Using long DNA pads containing many segments, each segment contains a cipher word followed by a plaintext word and the cipher word, acts as a hybridization site for binding of a primer. Cipher word is appended with a plaintext word to produce word-pairs. The word-pair DNA strands are used as a lookup table in conversion of plaintext into cipher text.

*One-time-pad DNA Sequence with l*ength n, contains d = n/(L1+ L2+ L3) copies of repeating unit *Repeating unit* made up of:

1. Bi = a cipher word of length L1 = c1log n
2. Ci = a plaintext word length L2= c2log n

Each sequence pair uniquely associates a plaintext word with a cipher word and the Polymerase acts as a "stopper" sequence of length L3 = c3.

To generate a set of oligonucleotides corresponding to the plaintext/cipher and word-pair strands, ~Bi used as polymerase primer and *extended* with polymerase by specific attachment of plaintext word Ci. The *Stopper sequence* prohibits extension of growing

DNA strand beyond boundary of paired plaintext word.

Methods for Construction of DNA one-time pads are based on the biotechnologies rather than bioinformatics and present difficult to achieve both full coverage and yet still avoiding possible conflicts by repetition of plaintext and cipher words.

This methods make use of DNA chip technology for random assembly of one-time pads.

The advantages are that are currently commercially available (Affymetrix) chemical methods for construction of custom variants are well developed.

Other method also based on biotechnologies is so called method *DNA chip Method* for Construction of DNA one-time pads where is used an array of immobilized DNA strands and multiple copies of a single sequence are grouped together in a microscopic pixel which is optically addressable. Using the technology for synthesis of distinct DNA sequences at each (optically addressable) site of the array and combinatorial synthesis conducted in parallel at thousands of locations, prepared of oligonucleotides of length L, the 4L sequences are synthesized in 4n chemical reactions.

As an Example: 65,000 sequences of length 8 use 32 synthesis cycles and 1.67x107 sequences of length 10 use 48 cycles. The construction of DNA One-time pads based on biotechnologies was first developed by the pioneer in this field (Adleman 1997).

XOR One-time-pad (Vernam Cipher) Cryptosystem based on biotechnologies One-time-pad: S is a sequence of independently distributed random bits

M is a plaintext binary message of n bits resulting in the following cipher text ,

Ci = Mi XOR Si for = 1,…,n .

*Decrypted bits, u*se commutative property of XOR Ci XOR resulting in:

Si = (Mi XOR Si) XOR Si= Mi XOR (Si XOR Si)= Mi.

DNA Implementation of XOR One-time-pad Cryptosystem:

The *plaintext messages is* one test tube of short DNA strands

The *encrypted message is* another test tube of different short DNA strands

Encryption by XOR One-time-pad maps these in a random and reversible way such as plaintext is converted to cipher strands and plaintext strands are removed. For the *efficient* DNA encoding Adleman proposed to use *modular base 4 as* DNA has four nucleotides. Encryption constitutes the addition of one-time-pad elements modulo 4 and decryption is the subtract one-time-pad elements modulo.

Details of DNA Implementation of XOR One-time-pad Cryptosystem based on biotechnologies:

Each plaintext message has appended a unique *prefix index tag* of length L indexing it.

Each of one-time-pad DNA sequence has appended unique *prefix index tag* of same length L, forming *complements* of plaintext message tags. Using recombinant DNA bio techniques such as annealing and ligation in order to *concatenate into a single DNA strand* each corresponding pair of a plaintext message and a one-time-pad sequence resulting in *enciphered by bit-wise XOR computation and* fragments of the plaintext are converted to cipher strands using the one-time-pad DNA sequences, and plaintext strands are removed.

The reverse decryption is similar using commutative property of bit-wise XOR operation.

BMC Methods to effect bit-wise XOR on Vectors. This method can adapt BMC methods for binary addition and similar to bit-wise XOR computation can disable carry-sums logic to do XOR

*BMC techniques for Integer Addition were implemented by (*Guarnieri, Fliss, and Bancroft 96), first BMC addition operations (on single bits) by (Rubin el al 98, OGB97, LKSR97, GPZ97) permit chaining on n bits.

Addition by *Self Assembly* of DNA tiles was exploited by (Reif, 97) and (LaBean, 99)

*XOR by Self Assembly of DNA tiles (*LaBean, 99**):** *XOR by Self Assembly of DNA tiles includes that f*or each bit Mi of the message, construct sequence ai that represents the ith bit.

Scaffold strands for binary inputs to the XOR are the usage of linkers to assemble the message M's n bits into scaffold strand sequence *a1, a 2 … a n.*

The One-time-pad is further portion scaffold strand *a*' *1a*' *2… a*'*n* and is created from random inputs add output tiles, the annealing give self assembly of the tiling.

The next step: adding ligase yields to the reporter strand:

R = a 1 a 2 … a n.a' 1 a' 2… a'n.b 1 b 2 … b n, where b i = a i XOR a'i, for i = 1,…,n.

In the next step the reporter strand is extracted by biotechnique of melting away the tiles, smaller sequences, and purifying it, contains concatenation of input message, encryption key, ciphertext.

Before the final last step using a marker sequence the ciphertext can be excised and separated based on its length being half that of remaining sequence. In the last step ciphertext is stored in a compact form.

These increasing importances of information security and the protection of human privacy rights as Confidentiality lead me to develop new security solutions based on modern technologies: Bioinformatics and Biotechnology.

In this work we present a technical process for protecting data assets such as personal medical information using Bioinformatics and a DNA cryptography technique based on bioinformatics rather then biotechnologies in this bioinformatics technique 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 developed a bioinformatic mechanism to generate encrypt-decrypt keys by taking into consideration specifics of the cryptography method and the individual's DNA genome analysis.

Our work was based on the complexity of developing, as a subset of JCE, an unconditionally secure DNAE System as part of our security provider, named DNAProvider, (Hodorogea, Ionas 2011).

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.

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).

4. Complexity of DNA encryption system as a subset of Java cryptography extension

Java Cryptography Extension (JCE) was developed as an extension package which includes implementation for cryptographic services.

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, (Fig.1)

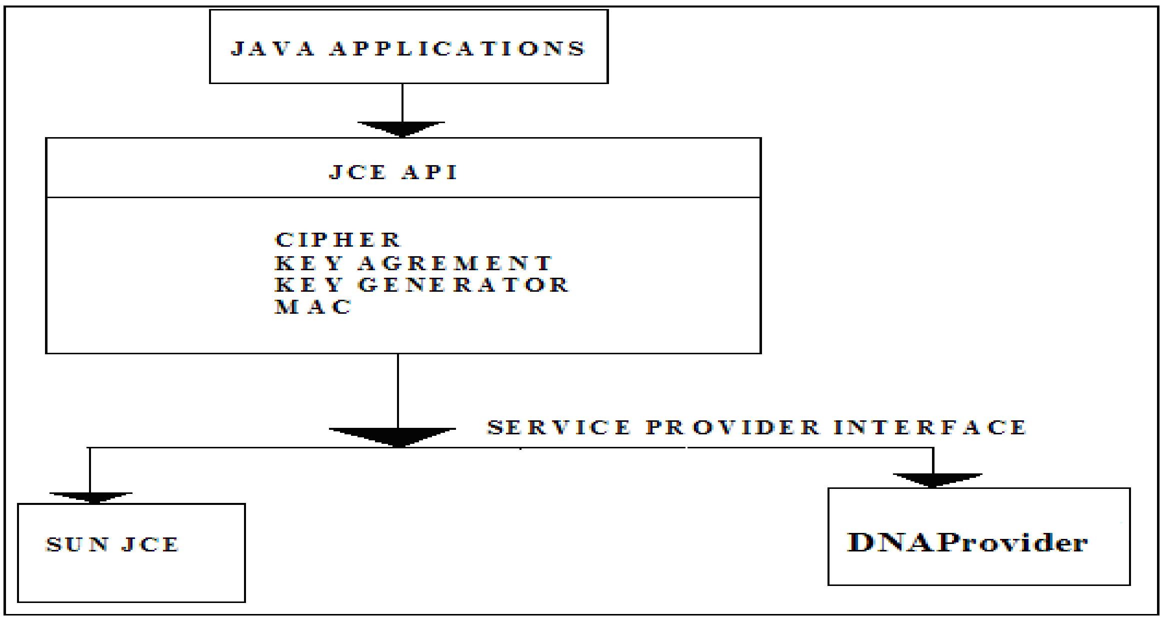


Fig. 1. Java Cryptography Extensions architectural model with unconditional secure DNA Encryption as part of our security provider (DNAProvider)

This support allows us to provide our independent implementation of DNAE System, based on the CDMB (Central Dogma of Molecular Biology).

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 requested functionality of the DNA Provider.

The security provider interface the means by which different security implementations may be plugged into the security package as message digests, encryption, digital signatures and keys, through JCE, JSSE and authentication through JAAS. Thus, JCE support allowed us to provide our independent implementation of DNA Cryptographic Keys Based on Evolutionary Models used for Security of Web-based Business Processes.

The classes necessary to handle secret keys come only with JCE. Keys and certificates are normally associated with some person or organization, and the way in which keys are stored, transmitted, and shared is an important topic in the security package.

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,(Fig.2).

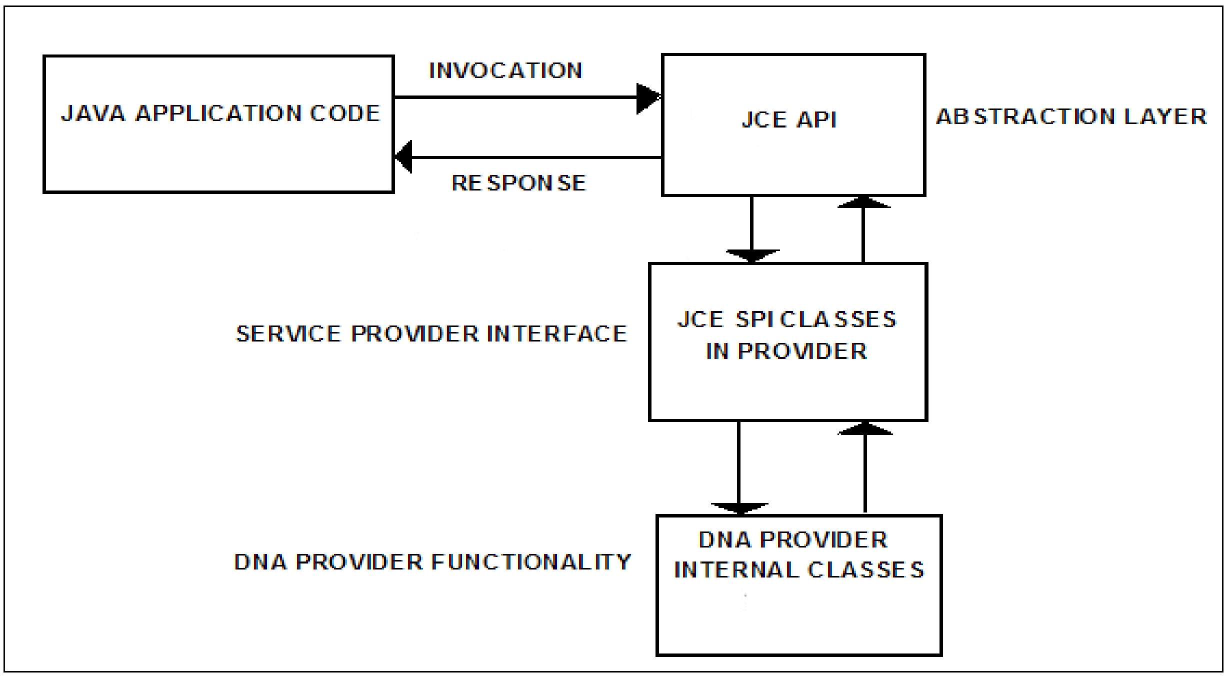

JAVA APPLICATION CODE
INVOCATION
RESPONSE
JCE API
ABSTRACTION LAYER
SERVICE PROVIDER INTERFACE
JCE SPI CLASSES IN PROVIDER
DNA PROVIDER FUNCTIONALITY
DNA PROVIDER INTERNAL CLASSES

Fig. 2. Invocation of DNAProvider for providing requested functionality

The security provider abstracts two ideas: engines and algorithms. An Engine Class defines an abstract cryptographic service, without its concrete implementation. The goal of the security provider interface is to allow an easy mechanism where the specific algorithms and their implementations can be easily changed or substituted. The architecture including all of this contains:

Engine classes, these classes come with the Java virtual machine as part of the core API.

Algorithm classes, at the basic level, there is a set of classes that implement particular algorithms for particular engines.

A default set of these classes is provided by the supplier of the Java platform. Other third-party organizations or individual can supply additional sets of algorithm classes. These classes may implement one or more algorithms for one or more engines.

Going to provide my own set of classes to perform security operations, I must extend the Provider class and register that class with the security infrastructure

Provider class is abstract, none of its methods are abstract, I need do is subclass the Provider class and provide an appropriate constructor.

The basic implementation of a DNAProvider security provider is:

```
public class DNAProvider extends Provider
{
public DNAProvider( )
{
super("DNAProvider", 1.0, "DNA Security Provider v1.0");
}
}
```

Here we define the skeleton of a DNAProvider that is going to provide certain facilities based on Central Dogma of Molecular Biology(CDMB).

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 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 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.

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. 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. 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.

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.

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 DNA encryption protocol

Resent research considers the use of the Human genome in cryptography and the famous DNA one-time-pad encryption schemes utilizing the indexed of random key string was first developed by Ashish Gehani, Thomas H. LaBean and John H. Reif.

At the lowest level, a genome can be described as a long string of nucleotides. It could be compared to a very long text made of four letters (strings of DNA). All living organisms consist of cells and in each cell there is the same set of chromosomes. Chromosomes are strings of DNA and serve as a model for the whole organism made from genes, which are made from blocks of DNA. Complete set of genetic material (all chromosomes) is called genome. The assumption of evolutionary models is that biological systems have evolved from the same origin, constantly reusing some basic building blocks and through the cycles of mutation and selection that constitute evolution, new functions have been created by reusing pieces of already existing DNA machinery. If we consider this problem in terms of sequences, this means that two sequences responsible for similar functions may be different, depending on how long they have been diverging. Many of the problems in bioinformatics and more specifically in sequence alignment are said to be NP complete as the number of potential solutions rises exponentially with the number of sequences and their length and the solution cannot be found in polynomial time and space. A sequence alignment is the representation of two sequences in a way that reflects their relationship and if the alignment is designed to reflect phylogenetic relationships, the residues will be aligned when they originate from the same residue in the common ancestor. If a given sequence lacks one residue, a gap will be inserted in its place at the corresponding position, in an evolutionary model context, a null sign means that a residue was inserted in one of the sequences or deleted in the other while the sequences were diverging from their common ancestor.

As Public-key algorithms are based on mathematical functions rather than on substitution and permutation and involves the use of two separate keys, in contrast to symmetric encryption, which uses only one key. When aligning the DNA sequences of the same gene from related species, there will usually be differences between the sequences because of evolution.

We developed a Unique Process System Pipeline Evolutionary Models of deriving DNA Cryptographic

Keys Sequences by deriving the DNA private/public keys from human genome analysis by computing the philogenetic tree relating and the branch length during evolution for chosen species. The molecular evolution model assigns probabilities to multiple-alignment columns in terms of the the philogenetic tree branches and is time dependent of frequency selections. Based on Kimura-Ohta theory Halpern and Bruno, have shown that mutation limit can be determined by substitution rates in terms of the mutation rates and equilibrium frequencies.

Models of DNA evolution were first proposed in 1969 by Jukes and Cantor, assuming equal transition rates and equal equilibrium frequencies for all bases.

In 1980 Kimura-Ohta introduced a model of DNA Evolution with two parameters: one for the transition and one for the transversion rate.

To estimate evolutionary distances in terms of the number of nucleotide substitutions and the evolutionary rates when the divergence times are known by comparing a pair of nucleotide sequences. There are two types of differences when homologous sites are occupied by different nucleotide bases and both are purines or both are pyrimidines. The difference is called Transition type when one of the two is a purine and the other is a pyrimidine then the difference is called transversion type.

Let P and Q be the fractions of nucleotide sites, showing between two sequences compared the transition and transversion type differences, then:

The Evolutionary Distance per Site is:

$$K = -(1/2)\ln\{(1-2P-Q)\} \tag{1}$$

The Evolutionary Rate per Year is then given by:

$$k = K/(2T) \tag{2}$$

T is the time since the divergence of the two sequences. If only the third codon positions are compared, then *the Synonymous Component of Evolutionary Base Substitutions per Site* is:

$$K'_S = -(1/2)\ln(1-2P-Q) \tag{3}$$

In biology, a substitution model describes the process from which a sequence of characters changes into another set of traits.

Each position in the sequence corresponds to a property of a species which can either be present or absent.

4.2 The technique of deriving DNA cryptographic keys based on evolutionary models

We developed and implemented a software tool for aligning the DNA Cryptographic Keys Sequences of the same gene from related chosen species with respect to Human DNA Sequences.

The alignment in the evolutionary system pipeline of DNA Cryptographic Keys Sequences was realized with trained ProbCons tool which is a pair-hidden Markov model-based on progressive alignment algorithm, that primarily differs from most typical approaches in its use of maximum expected accuracy.

As Public-key algorithms are based on mathematical functions rather than on substitution and permutation and involves the use of two separate keys, in contrast to symmetric encryption, which uses only one key. When aligning the DNA sequences of the same gene from related species, there will usually be differences between the sequences because of evolution, (Ochman, 2003). Some of these will lead to differences in the amino acids of the encoded protein (non-synonymous changes). Because of the degeneracy of the genetic code leave the protein unchanged (synonymous, or silent changes). If Ka/Ks< 1 *Purifying (negative) selection,* most proteins are well adapted to carry out their function change would not lead to the creation of selective advantage. If *Ka/Ks >1 Diversifying (positive)*, selection has acted to change the protein and if *Ka/Ks= 1 Neutral evolution*, (Mustonen, Lässig, 2005). After aligning our extracted DNA Sequences with ProbCons tool, we derive the private/public pair DNA cryptographic keys based on evolutionary models and based on mathematical functions.

ProbCons is a tool for generating multiple alignments of protein sequences. It uses a combination of probabilistic modeling and consistency-based alignment techniques and has achieved the highest accuracies of all alignments methods. The basic for ProbCons algorithm is the computation of pairwise posterior probability matrices, P(xi ~ yi |x, y), which give the probability that one should match letters *xi* and *yi* when aligning two sequences *x* and *y*. ProbCons uses a simple probabilistic model that allows for efficient computation of this probabilities. Given a set of sequences ProbCons computes the posterior probability matrices for each pair of sequences and computes the expected accuracy of each alignment.

As Public-key algorithms are based on mathematical functions rather than on substitution and permutation and involves the use of two separate keys, in contrast to symmetric encryption, which uses only one key. When aligning the DNA sequences of the same gene from related species, there will usually be differences between the sequences because of evolution.

We developed a Unique Process System Pipeline Evolutionary Models of deriving DNA Cryptographic

Keys Sequences by deriving the DNA private/public keys from human genome analysis by computing the philogenetic tree relating and the branch length during evolution for chosen species. The molecular evolution model assigns probabilities to multiple-alignment columns in terms of the the philogenetic tree branches and is time dependent of frequency selections. Based on Kimura-Ohta theory Halpern and Bruno, have shown that mutation limit can be determined by substitution rates in terms of the mutation rates and equilibrium frequencies.

For every alignment column, we calculated the likelihood under two evolutionary models: a "foreground" and a "background" model.

The background model assumes a rate model (Felsenstein 1981), parameterized by the branch lengths of the phylogenetic tree:

w is a vector of nucleotide frequencies, with w_α the frequency of nucleotide α,

$r_{\alpha\beta}$ -the rate of substitution from base β to base α which is proportional to w_α, independent of β.

For every background evolution models we have a corresponding foreground model. The difference between the foreground model and background model is that the background model assumes that all positions undergo substitutions from base β to base α at the same rate $r_{\alpha\beta} \propto w_\alpha$.

The foreground model I assume that, at a given position i, the substitution rates $r^i_{\alpha\beta} \propto w^i_\alpha$ are altered due to specific selection preferences for certain bases at this position, parameterized by nucleotide frequencies w^i_α.

The parameters w^i_α, at each position are unknown, integrated out of the likelihood.

For each alignment column of the reference species, in intergenic regions and in genes, we calculate the ratio *R*, representing the likelihoods of foreground and background evolutionary models.

Halpern and Bruno in 1998 estimated the evolutionary distances from coding sequences taking into account protein-level selection to avoid relative underestimation of longer evolutionary distances.

The equilibrium frequencies determine the maximum dissimilarity expected for highly diverged but functionally and structurally conserved sequences and crucial for estimating long distances (Molina, Nimwegen 2008).

Halpern and Bruno introduced a codon-level model of coding sequence evolution in which position-specific amino acid equilibrium frequencies were free parameters. They demonstrated the importance and feasibility of modeling such behavior as the model produced linear distance estimated over a wide range of distances. Some alternative models underestimated long distances, relative to short distances.

If r is the rate of substitution from a base a to a base b at position i, μ is the rate of mutation from a to b and w is the equilibrium frequency of nucleotide i, at this position, (Halpern AL, Bruno WJ, 1998).

Following Golding and Felsenstein (1990), Halpern and Bruno (1998) who have shown that mutation limit of the standard Kimura-Ohta theory, one can uniquely determine substitution rates in terms of the mutation rates and the equilibrium frequencies w^i_α if $r^i_{\alpha\beta}$ is the rate of substitution from β to α at position i, $\mu_{\alpha\beta}$ the rate of mutation from β to α, and w_{α}^{i} the equilibrium frequency of α at this position, we have (Halpern and Bruno 1998).

We derive the private/public pair DNA cryptographic keys based on evolutionary models and based on mathematical functions.

We started with extracting from public available database all orthologus DNA coding sequences for all genes, from related species with respect to Human Genome sequences. A genome of a reference species in our case is Human Genom (hg18) and two more additional genomes are: Taurus Genome (bosTau3) and Dog Genome (canFam2). We extracted the DNA sequences for 29.000 genes which equals to 44103 pages in printable format. Using a trained parameter set for ProbCons tool we aligned all orthologus DNA coding sequences of our choosen species for all genes with respect to Human DNA coding sequences.

ProbCons achieved the highest accuracies of all multiple alignments methods as it uses probabilistic modeling and consistency-based alignment techniques.

We computed the philogenitic tree for our chosen species and the branch length during evolution, (Fig. 4) with respect to human genome (hg18).

A Software Application, reeds the tree, computes the pairwise alignment, computes the branches of the tree for our

chosen mammalian species. Public-key algorithms are based on mathematical functions, rather than on substitution and permutation and involve the use of two separate keys in contrast to symmetric encryption, (Fig. 3).

In Table 1, Second Column (C2) model represents the computed DNA Public Keys, with respect to Colum C1 and assumes substitution rate model which is calculated by the branch lengths of the phylogenetic tree and a vector of nucleotide frequencies, (Table 1) and represents the public keys.

Given the transition probabilities and given a phylogenetic tree we calculated the ratio for an alignment column C3/C2, which is the product over transition probabilities for each branch of the tree we summed over all possible nucleotides for internal nodes calculated by recursive algorithm introduced by Felsenstein.

The first column C1 represents all possible three base sequences with respect to human species. Second Colum (C2) model, with respect to C1 assumes substitution rate model which is calculated by the branch lengths of the phylogenetic tree and a vector of nucleotide frequencies, and represents the public key. The third Colum (C3) assumes that at a given position, the substitution rates are altered during due to specific selection preferences for a certain base. The last Colum is the ratio C3/C2 and represents the private key, (Table 1).

Using the same model and desired length of bases from the first column we can derive the public/private keys used in Java KeyStore with respect to human or desired number of species. Resulting in new set of public/private *DNA Cryptographic Keys for our Java DNA KeyStore usage.*

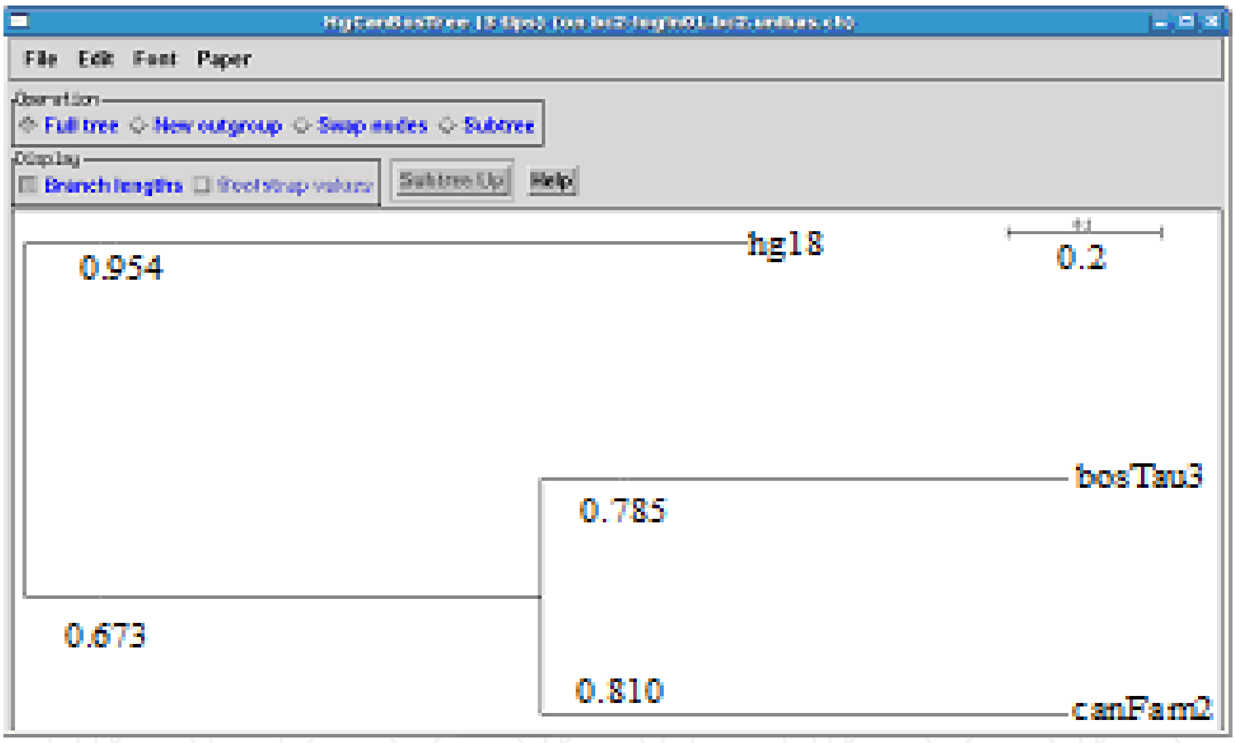
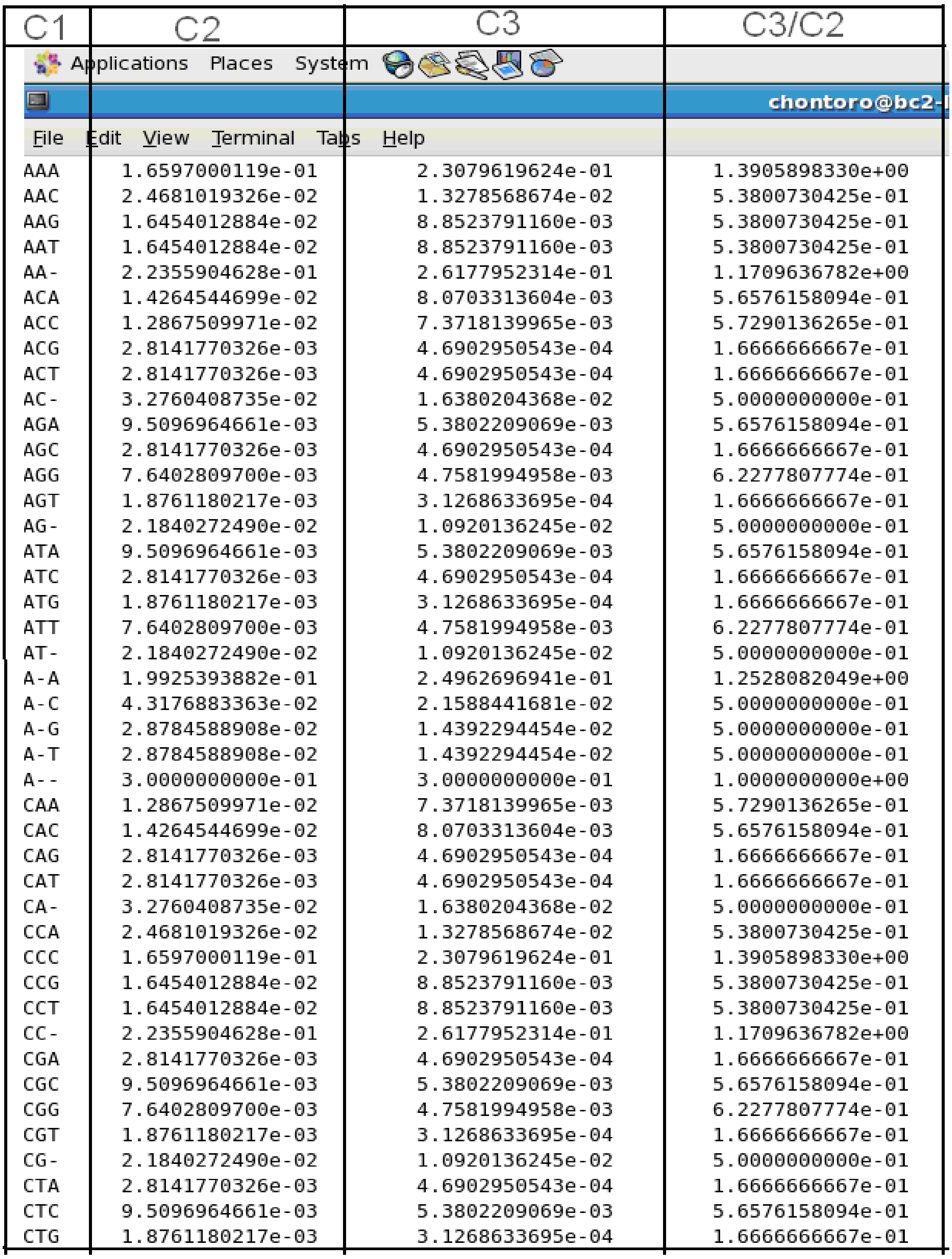


Fig. 3. Computed philogenetic tree and the branch length



C1	C2	C3	C3/C2
AAA	1.6597000119e-01	2.3079619624e-01	1.3905898330e+00
AAC	2.4681019326e-02	1.3278568674e-02	5.3800730425e-01
AAG	1.6454012884e-02	8.8523791160e-03	5.3800730425e-01
AAT	1.6454012884e-02	8.8523791160e-03	5.3800730425e-01
AA-	2.2355904628e-01	2.6177952314e-01	1.1709636782e+00
ACA	1.4264544699e-02	8.0703313604e-03	5.6576158094e-01
ACC	1.2867509971e-02	7.3718139965e-03	5.7290136265e-01
ACG	2.8141770326e-03	4.6902950543e-04	1.6666666667e-01
ACT	2.8141770326e-03	4.6902950543e-04	1.6666666667e-01
AC-	3.2760408735e-02	1.6380204368e-02	5.0000000000e-01
AGA	9.5096964661e-03	5.3802209069e-03	5.6576158094e-01
AGC	2.8141770326e-03	4.6902950543e-04	1.6666666667e-01
AGG	7.6402809700e-03	4.7581994958e-03	6.2277807774e-01
AGT	1.8761180217e-03	3.1268633695e-04	1.6666666667e-01
AG-	2.1840272490e-02	1.0920136245e-02	5.0000000000e-01
ATA	9.5096964661e-03	5.3802209069e-03	5.6576158094e-01
ATC	2.8141770326e-03	4.6902950543e-04	1.6666666667e-01
ATG	1.8761180217e-03	3.1268633695e-04	1.6666666667e-01
ATT	7.6402809700e-03	4.7581994958e-03	6.2277807774e-01
AT-	2.1840272490e-02	1.0920136245e-02	5.0000000000e-01
A-A	1.9925393882e-01	2.4962696941e-01	1.2528082049e+00
A-C	4.3176883363e-02	2.1588441681e-02	5.0000000000e-01
A-G	2.8784588908e-02	1.4392294454e-02	5.0000000000e-01
A-T	2.8784588908e-02	1.4392294454e-02	5.0000000000e-01
A--	3.0000000000e-01	3.0000000000e-01	1.0000000000e+00
CAA	1.2867509971e-02	7.3718139965e-03	5.7290136265e-01
CAC	1.4264544699e-02	8.0703313604e-03	5.6576158094e-01
CAG	2.8141770326e-03	4.6902950543e-04	1.6666666667e-01
CAT	2.8141770326e-03	4.6902950543e-04	1.6666666667e-01
CA-	3.2760408735e-02	1.6380204368e-02	5.0000000000e-01
CCA	2.4681019326e-02	1.3278568674e-02	5.3800730425e-01
CCC	1.6597000119e-01	2.3079619624e-01	1.3905898330e+00
CCG	1.6454012884e-02	8.8523791160e-03	5.3800730425e-01
CCT	1.6454012884e-02	8.8523791160e-03	5.3800730425e-01
CC-	2.2355904628e-01	2.6177952314e-01	1.1709636782e+00
CGA	2.8141770326e-03	4.6902950543e-04	1.6666666667e-01
CGC	9.5096964661e-03	5.3802209069e-03	5.6576158094e-01
CGG	7.6402809700e-03	4.7581994958e-03	6.2277807774e-01
CGT	1.8761180217e-03	3.1268633695e-04	1.6666666667e-01
CG-	2.1840272490e-02	1.0920136245e-02	5.0000000000e-01
CTA	2.8141770326e-03	4.6902950543e-04	1.6666666667e-01
CTC	9.5096964661e-03	5.3802209069e-03	5.6576158094e-01
CTG	1.8761180217e-03	3.1268633695e-04	1.6666666667e-01

Table 1. Public/Private DNA Cryptographic Keys

5. Conclusion

Considering GRID computing security where the heterogeneous resources are shared and located in different places belonging to different administrative domains over a heterogeneous network, additional security requirements must be satisfied compare to classical network security. Communication between GRID entities must be secure and confidentiality must be ensured for sensitive data, from communication stage, to potential storage stage. Cryptographic algorithms for confidentiality play a major importance role in nowadays information security.

Our work described in this chapter was based on the complexity of developing the cryptographic package provider, named DNAProvider as Java Cryptographic Extension (JCE), where we derive the DNA Cryptographic Keys Based on Evolutionary Models for Security of Software Applications, extending the JCE by implementing faster and more secure DNA Encryption (DNAE) system based on the Central Dogma of Molecular Biology (CDMB). Sun Microsystems certified and signed our DNAProvider as Java Cryptographic Extension (JCE) with DNA cryptographic algorithm. We got the Code Signing Certificate from Sun Microsystems for our DNAProvider as Java Cryptographic Extension (JCE) with DNA cryptographic algorithm which is available for 5 years, until with the reference #679, when renewing it in 2013.

In our future research work we intend to integrate our developed system pipeline of deriving DNA Cryptographic Keys Based on Evolutionary Models implemented and tested at University of Basel, Switzerland, in our DNAProvider as Java Cryptographic Extension (JCE) with DNA Encryption (DNAE) system for use in security of our developed Web-based Business Processes Software Applications. We aim to use DNA Provider with unconditional secure DNAE system to ensure security of today's web-based business processes. as e-commerce and Internet banking. (Hodorogea, Ionas, 2011).

6. Acknowledgment

This research work is supported by the Company INNOVA BIOTECH, Cluj-Napoca, Romania.

7. References

Abad, C., Taylor, J., Sengul, C., Yurcik, W., Zhou,Y., & Rowe, K. (2003). Log correlation for intrusion detection: A proof of concept. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC 2003).* Los Alamitos, CA: IEEE Computer Society Press.

Almgren, M., & Jonsson, E. (2004). Using active learning in intrusion detection. In *Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW'04).* Los Alamitos, CA: IEEE Computer Society Press.

Anderson, J. P. (1980). *Computer security threat monitoring and surveillance* (Tech.l Rep.). FortWashington, PA: James P. Anderson.

Alberts C., Audrey D., "Managing Information Security Risks: The OCTAVESM Approach ", Addison Wesley Professional, July 09, 2002.

Bace, R., & Mell, P. (2001). *Intrusion detection systems.* NIST special publication in intrusion detection systems. Retrieved from http://csrc.nist gov/publications/nistpubs/800-31/sp800-31.pdf

Beznosov, K. (2004). *On the benefits of decomposing policy engines into components.* Third Workshop on Adaptive and Reflect Middleware, Toronto, Canada.

Blobel, B. (2001). The European TrustHealth project experiences with implementing a security infrastructure. *International Journal of Medical Informatics, 60,* 193-201.

Blobel, B., Hoepner, P., Joop, R., Karnouskos, S., Kleinhuis, G., & Stassinopoulos, G. (2003). Using a privilege management infrastructure for secure Web-based e-health applications. *Computer Communication*, *26*(16), 1863-1872.

Blobel, B. (2004). Authorisation and access control for electronic health record system. *InternationalJournal of Medical Informatics, 73,* 251-257.

Hodorogea T., Ionas O., (2011), "Security of Business to Business and Business to Customer Software Applications Based on the Central Dogma of Molecular Biology (CDMB) and Evolutionary Models", IEEE Explore (ITI) 2011, International Conferince on Information Technology Interfaces, June, 2011, Cavtat, Croatia.

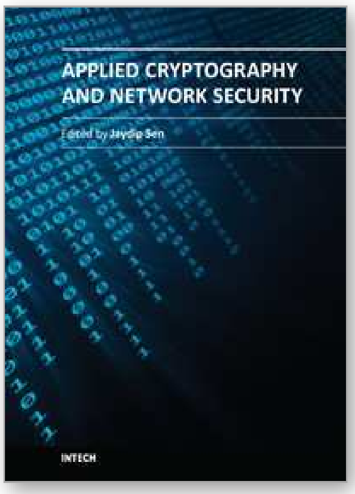
Halpern, A.L. and Bruno, W.J. 1998. Evolutionary distances forprotein-coding sequences: Modeling site-specific residue frequencies.Mol. Biol. Evol. 5: 910–917.

Halligan, D.L., Eyre-Walker, A., Andolfatto, P., and Keightley, P.D. 2004, Patterns of evolutionary constraints in intronic and intergenic DNA of *Drosophila*. *Genome Res.* 14: 273–Rajewsky, N., Socci, N.D., Zapotocky, M., and Siggia, E.D. 2002. The evolution of DNA regulatory regions for proteo-gamma bacteria by interspecies comparisons. *Genome Res.* 12: 298–308

Rogozin, I.B., Makarova, K.S., Natale, D.A., Spiridonov, A.N., Tatusov, R.L., Wolf, Y.I., Yin, J., and Koonin, E.V. 2002. Congruent evolution of different classes of non-coding DNA in prokaryoticgenomes.Nucleic Acids Res. 30: 4264–4271. doi:2001. Codon bias at the 3_-side of the initiation codon is correlated

van Nimwegen, E. 2003. Scaling laws in the functional content of genomes. Trends Genet.

van Nimwegen, E. 2004. Scaling laws in the functional content of genomes: Fundamental constants of evolution In Power laws, scale-free networks and genome biology (eds. E. Koonin et al.), pp.236–253 Landes Bioscience, Austin, TX.



Applied Cryptography and Network Security
Edited by Dr. Jaydip Sen

ISBN 978-953-51-0218-2
Hard cover, 376 pages
Publisher InTech
Published online 14, March, 2012
Published in print edition March, 2012

Cryptography will continue to play important roles in developing of new security solutions which will be in great demand with the advent of high-speed next-generation communication systems and networks. This book discusses some of the critical security challenges faced by today's computing world and provides insights to possible mechanisms to defend against these attacks. The book contains sixteen chapters which deal with security and privacy issues in computing and communication networks, quantum cryptography and the evolutionary concepts of cryptography and their applications like chaos-based cryptography and DNA cryptography. It will be useful for researchers, engineers, graduate and doctoral students working in cryptography and security related areas. It will also be useful for faculty members of graduate schools and universities.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821