cryptography

MDPI

Article

A Cryptographic System Based upon the Principles of Gene Expression

Harry Shaw

NASA/Goddard Space Flight Center, Greenbelt, MD 20771, USA; harry.c.shaw@nasa.gov; Tel.: +13-012-868-680

Received: 21 October 2017; Accepted: 16 November 2017; Published: 21 November 2017

Abstract: Processes of gene expression such as regulation of transcription by the general transcription complex can be used to create hard cryptographic protocols which should not be breakable by common cipherattack methodologies. The eukaryotic processes of gene expression permit expansion of DNA cryptography into complex networks of transcriptional and translational coding interactions. I describe a method of coding messages into genes and their regulatory sequences, transcription products, regulatory protein complexes, transcription proteins, translation proteins and other required sequences. These codes then serve as the basis for a cryptographic model based on the processes of gene expression. The protocol provides a hierarchal structure that extends from the initial coding of a message into a DNA code (ciphergene), through transcription and ultimately translation into a protein code (cipherprotein). The security is based upon unique knowledge of the DNA coding process, all of the regulatory codes required for expression, and their interactions. This results in a set of cryptographic protocols that is capable of securing data at rest, data in motion and providing an evolvable form of security between two or more parties. The conclusion is that implementation of these protocols will enhance security and substantially burden cyberattackers to develop new forms of countermeasures.

Keywords: cybersecurity; biomimetic; encryption; confidentiality; gene expression; gene regulation

1. Introduction

Network security is a vital component of the design of any network. There are five main requirements to be addressed in developing a secure network: authentication, confidentiality, data integrity, non-repudiation, and access control. In vivo, biomolecular cellular systems of gene expression authenticate themselves through various means such as binding of transcription factors and promoter sequences. These factors also enforce access control. They have means of retaining confidentiality of the meaning of genome sequences through processes such as control of protein expression. They are capable of establishing data integrity and non-repudiation through transcriptional and translational controls. The motivation for developing this protocol architecture is to utilize these naturally occurring capabilities from biomolecular systems.

A suite of genomics and proteomics based authentication and confidentiality protocols will be demonstrated that augment traditional network security approaches with concepts from molecular biology via the regulation of gene expression. These protocols are agnostic to their implementation and can be incorporated into any existing network security protocol (Secure http, Secure Sockets Layer (SSL), Transport Layer Security (TLS), Internet Protocol Security (IPSec), etc.) or any future network security strategy. The protocols can be implemented for implementing web-based security strategies, digital signatures, digital rights management, and general purpose encryption for data in motion or data at rest.

These protocols will provide new challenges for network attackers by forcing them to work in both the information security domain and the molecular biology domain. Although no security

strategy is without vulnerabilities, the intent of this work is to present a completely new set of problems for network attackers [1]. The existing authentication and confidentiality protocols and processes are becoming more vulnerable to attacks and networks are becoming less secure. Simultaneously, the power of cryptanalysis against existing encryption methodologies is growing rapidly. However, the current infrastructure investment in these methodologies is too large to abandon. No alternate authentication infrastructure exists which can adequately replace the current methods.

In this protocol, there are tools from conventional cryptography that are used along with the principles of the regulation of gene expression. This paper concentrates on the biological aspects of the protocol.

1.1. Weak Points with the Current Security Approaches

Cryptanalysis techniques are very strong and improve with increases in computing capability. Security protocols that rely heavily upon algorithms such as the RSA algorithm have been attacked using adaptive chosen cipherattacks. These attacks involve simple power analysis and differential power analysis of smartcard implementations. Smartcards are also vulnerable to reverse engineering using chip level diagnostic testing. A smartcard attack involved leaking side channel implementation through its implementation of the Chinese Remainder Algorithm [2]. Protocols using the modular exponentiation approach can be attacked via a number of methods:

1. Timing attacks using the Chinese Remainder Algorithm and Montgomery's Algorithm. This timing attack works by enabling factorization of the RSA modulus n. It works if the exponentiation is carried out by the Chinese Remainder Algorithm and the multiplication of the prime factors is performed by Montgomery's Algorithm [3].
2. Analysis of short RSA exponents. This attack uses a continued fractions algorithm to make an estimate using the public key exponent, e and the modulus, p*q to make an estimate of the private key exponent, d. It relies on the fact that with e < p*q and GCD $(p-1, q-1)$ is small, d can be estimated [4].
3. Lattice basis reduction (LLL) algorithms. This type of attack can use a forged signature to recover RSA keys [5]. Lattice-based signatures are also vulnerable to fault attacks as demonstrated by Bindel et al. in 2016 [6].
4. General timing attacks on modular exponential algorithms. These attacks involve timing characterization of cryptographic functions such as RSA and others to correlate key computation cycles and timing to actual key values. [7]. This includes timing attacks on OpenSSL described in 2013 [8].

Protocols for performing authentication are vulnerable to social engineering. The use of two-phase authentication has been growing to enhance authentication reliability. However, two-phase authentication is still vulnerable to cyberattack especially as hosts such as on-line banking attempt to make their on-line services more user-friendly [9].

Additionally, the useful lifetime of cryptographic codes is unpredictable and the existence of network vulnerabilities due to lax implementation of existing security protocols continues to be a major problem as demonstrated by the growing number of successful cyberattacks against major institutions and governments [10].

1.2. DNA Cryptography and the Central Dogma

DNA cryptography using the central dogma of biology has been published that takes plaintext through a process of DNA→RNA→Amino Acid coding. Researchers in 2010 published a DNA-based Cryptography for Secure Mobile Networks scheme [11] in which binary plaintext is converted to a DNA text via a substitution code, introns are inserted into DNA text and a key is passed to the receiver over a secure channel to provide the details of the intron insertion. The new DNA text is transcribed into a mRNA code utilizing only the exons and the exons are translated into an amino acid protein

code requiring a second, secure transfer of translation data so that the receiver can decode the protein code back to the mRNA, mRNA back to DNA sequence, and the DNA sequence stripped of the introns and converted back to the original binary plaintext. The protein code can be transmitted over an open channel. There are many variations on this theme in the literature [12,13].

1.3. DNA Computing and Elliptic Curve Cryptography

A combination of DNA computing and Elliptic Curve Cryptography (ECC) has been described [14] for a powerful form of DNA encryption. It permits encrypted traffic over communication links, which may not be secure. Sender and receiver agree on an auxiliary base parameter as a pre-shared secret for the ECC process. A substitution code for the plaintext is performed to convert it to DNA text, which is converted to integers, followed by conversion to ECC curve points using Koblitz's algorithm. The curve points are encrypted with the ECC algorithm.

1.4. Other DNA Encryption Systems

Systems using DNA as a one-time code pad in a steganographic approach have been described [15]. In work by Gehani et al. they proposed use of DNA codes assembled from short oligonucleotide sequences, into one-time pads. They further assume that the one-time pads can be kept as a pre-shared secret. The approach relies on encoding the plaintext through a DNA substitution code or a bit-wise XOR function between the plaintext and the DNA sequence. They also propose that the language for creating the DNA ciphertext be disjoint from the plaintext. Gehani also proposes an approach with biological instantiation. The approach is compatible with using DNA one-time pads and custom DNA chips with complementary sequences to an encrypted sequence such that an encrypted image could be decrypted and revealed fluorescently.

A symmetric key block cipher approach using DNA transcription and translation has been demonstrated by Sadeg [16]. This work uses nomenclature of transcription and translation. The encryption algorithm generates ciphertext blocks 128 bits in length from a plaintext block of 128 bits and a key of 128 or 256 bits. There exist sub-key generators that provide nr + 2 keys, with nr equaling the number of iteration rounds. Additional symmetric block key cipher approaches using DNA cryptography continue to appear in the published literature [17].

An image compression and encryption system using a DNA-based alphabet [18] was demonstrated including a genetic algorithm-based compression scheme. This work is based upon the principles of fractal-based languages such as SCAN. This approach encodes data into large fields of $(n \times n)!$ pixel-like data and selects one of the $(n \times n)$ permutations as the ciphertext.

1.5. Cryptography on the Basis of Separation by Gel Electrophoresis

In the category of techniques that use a biological instantiation, researchers in 2000 proposed an optical technique in which a message was coded into DNA and subjected to gel electrophoresis to separate the DNA sequence into bands. The DNA message would also be mixed with nonsense DNA to create a different pattern of bands and the two could be subtracted from each other at the receiver to resolve the message [19].

1.6. DNA Watermarking

By using the natural redundancy of the amino acid codon system, messages can be coded into biologically functional genomic sequences without disrupting the ability of the code gene to be expressed. This algorithm permits a user to insert encrypted data into a genome of choice. Researchers in 2008 [20,21] created a system of DNA watermarks in which Genetically Modified Organisms (GMOs) could be tagged with a DNA message without disrupting the process of gene expression. It does this primarily by encoding the message into the third base in a triplet with synonymous codons. The natural redundancy of the codon system is such that the third base can sometimes be altered without changing the codon's meaning to another amino acid. This was elegantly demonstrated on

the *Vam7* gene in a mutant *Saccharomyces cerevisiae* strain CG783. They proved that the watermark mutation did not influence subsequent mRNA translation into protein. Additional research in DNA watermarking continues including work that uses codon postfix nomenclature [22].

2. Materials and Methods

2.1. Basis of the Cryptographic System Relying on Principles of Gene Expression

1. There exists a scheme to reversibly convert plaintext to DNA nucleotide codes. The methodology of the protocol allows users to utilize their own DNA coding scheme. It is also possible to use one of the DNA coding schemes developed by the author [23–26]. The plaintext to DNA conversion in [26] permits utilization of a wider set of DNA nucleotides than other coding schemes. Thus, a DNA codeword dictionary such as:

$$A_D = \{A, C, G, T, MeC, H, X\}, \tag{1}$$

   which represents the bases adenine, cytosine, guanine, thymine, the epigenetic marker methyl-cytosine, and mutagenic bases hypoxanthine, and xanthine can be implemented. The plaintext is coded into prefix-free binary codewords which are encrypted with a pre-shared key and converted to a DNA-based message as shown in [26]. The product is an unstructured sequence of nucleotide codes.
2. The DNA text is mapped into the structure of a gene complete with introns, exons, regulatory regions, etc. This output is called a ciphergene. This represents the level 1 encryption and the inverse operation is the level 1 decryption. The purpose of this coding from a security perspective is that a single sequence of letters from a small alphabet can be used to represent a large set of permutations of message combinations.
3. The ciphergene code is then operated on by a series of protein transcription factor codes that combine with their counterpart regulatory codes on the ciphergene to produce a new coded sequence that represents a coded transcriptional complex. The output of level 2 is the Pre-Transcriptional Complex and represents the level 2 encryption and the inverse operation is the level 2 decryption.
4. The third step is a series of operations that takes the Pre-Transcriptional Complex (PTC) code, which is operated on by protein and RNA polymerase codes resulting in a basal transcriptional complex code. The basal transcriptional complex code (BTC) is processed by algorithms and maps the code into a messenger RNA code, called the cipher-mRNA code. The cipher-mRNA now consists only of codons of the original DNA text message and is translated into a protein code, called the cipherprotein. The output of level 3 is the cipherprotein code that is transmitted from the sender to the receiver. The receiver applies the symmetric decryption keys to recover the cipher-mRNA and then performs all subsequent steps to reach level 2, level 1, and decoding to produce the plaintext.
5. The resulting codes for ciphergenes, cipher-mRNA (c-mRNA), and cipherproteins are subject to the processes of regulation of expression through operations on the codes. This can be done as pre- or post-transcriptional operations as well as pre- or post-translational operations such that these processes are utilized as part of the network security concept of operations. The scope of the protocols can be described in biological terms as the regulated transcription of genes to form messenger RNA followed by translation of the messenger RNA into proteins.

   Table 1 summarizes the steps in the encryption and decryption process.

Table 1. Genomic Proteomic encryption and decryption process.

Encryption Level	Input	Ouptut	Decryption Level	Input	Output
-	Plaintext	DNA text	3C	Cipherprotein	c-mRNA
1	DNA text	Ciphergene	3B	c-mRNA	BTC
2	Ciphergene	PTC	3A	BTC	PTC
3A	PTC	BTC	2	PTC	Ciphergene
3B	BTC	c-mRNA	1	Ciphergene	DNA text
3C	c-mRNA	Cipherprotein	-	DNA test	Plaintext

2.2. Coding of Sequences as Objects.

An object is defined as a genomic or proteomic sequence. It could be sequence defined at:

- the nucleotide base level (e.g., AGGCT . . .)
- the codon level, (AAG, TTA, CGC, . . .)
- transcription factor/ binding site (SP1, CCAT, AP2, . . .)
- protein transcription factor (TFIIA, TFIIB, . . .) and so forth.

Each object is drawn from the elements in a dictionary set associated with that object, for example:

- Nucleotides: N = {A, T, C, G, U, I, MeC, X, H}
- DNA Codons: DC = {ATT, ATC, ATA, CTT, CTC, CTA, CTG, TTA, TTG, GTT, GTC, GTA, GTG, TTT, TTC, ATG, TGT, TGC, GCT, GCC, GCA, GCG, GGT, GGC, GGA, GGG, CCT, CCC, CCA, CCG, ACT, ACC, ACA, ACG, TCT, TCC, TCA, TCG, AGT, AGC, TAT, TAC, TGG, CAA, CAG, AAT, AAC, CAT, CAC, GAA, GAG, GAT, GAC, AAA, AAG, CGT, CGC, CGA, CGG, AGA, AGG, TAA, TAG, TGA}
- Transcription factors: TF = {TFII, TBP, . . . }

Associated with each dictionary is a set of class elements that describe the function of an element in a given sequence. For the set of nucleotides, *N*, the classes might be:

C_N = {Promoter, Upstream Activator, TATA, Exon, Intron, . . . }.

The class defines the function of the element in a sequence at a given position. Each element of an object is mapped into a class. For example, all the nucleotides in the sequence from Figure 1 in the range of −275 to −200 would be mapped into a code in the UAS (Upstream activator sequence).

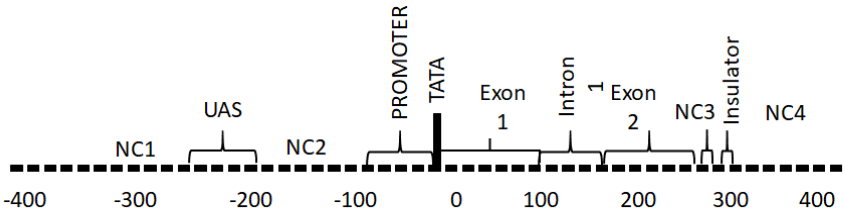

Figure 1. Biological gene structure ready for encryption.

Figure 1 depicts in a generic form some of the basic classes needed for transcriptional regulation of genes.

- Promoter. The promoter region is responsible for the binding of RNA polymerase, transcription factors and for the subsequent initiation of transcription.

- Upstream Activating Sequence. This is a region upstream of the transcriptional start site that binds transcription factor proteins required for transcription.
- Downstream Activating Sequence. This is a region downstream from the transcriptional start site that binds transcription factor proteins required for transcription.
- Exon. These regions contain the codons that are ultimately translated into proteins from messenger RNA.
- Introns. These are non-coding intervening regions between exons. Introns may also contain regulatory elements.
- TATA. This is a recognition sequence of bases (ATA(A/T)A(A/T)(A/G)) [27] that appears in some genes upstream of the transcription start site and binds TATA box binding proteins required for transcription. Not all genes have TATA boxes and some genes have non-canonical TATA boxes.
- Non-coding. These are regions without a specific function assigned.
- Insulator. The insulator is a regulator region that acts as a repressor of transcription of adjacent genes.

2.3. Coding the General Transcriptional Complex

Figure 2 provides a block diagram of the eukaryotic general transcriptional complex [28]. In Figure 2A, the ellipses depict the pre-transcriptional complex proteins without RNA Polymerase II. In Figure 2B, the completed transcriptional complex is shown. TFIIA, TFIIB, . . . , TFIIH are general transcriptional protein complexes and it is possible to decompose each of those elements in lower level elements. The boxes depict the regulatory sequences in a generic gene. Transcription can occur only when the transcription factors have bound to the proper locations of the regulatory sequences, thus allowing RNA Polymerase II to bind and initiate transcription. The network diagram of blocks depicts the required interaction of codes. Red dashed lines are protein–protein interactions. Black solid lines are protein–nucleic acid interactions.

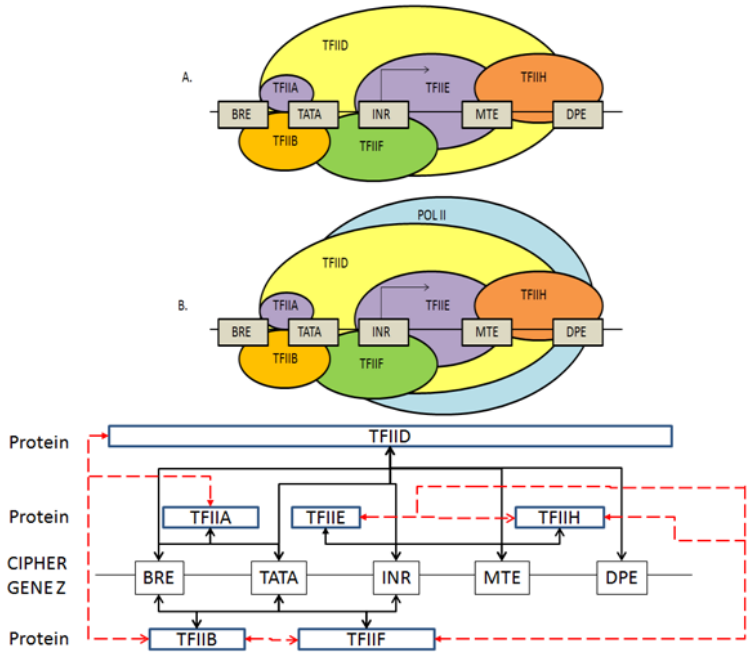

Figure 2. Coding the General Transcriptional Complex. (**A**) shows the pre-transcriptional complex of general transcription factor proteins bound to gene regulatory sequences; (**B**) shows the completed basal transcriptional complex with protein RNA Polymerase II bound to the pre-transcriptional complex; Below (**B**) is the view of the associations of (**A**) converted to a series of associations between transcription factors and other transcription factors as well as transcription factors to regulatory sequences.

The Method of Types [29] is used as a basis to create types corresponding to the required elements of the genomic and proteomic cryptographic codes. Let there be a set of all transcription factor codes, $\{tf_1, tf_2, tf_3, \ldots, tf_n\}$ and let a subset of these codes be assigned to the set of codes for TFII, $\{tfII_1, tfII_2, \ldots, tfII_m\}$. Let there be a set of all regulatory sequence codes, $\{r_1, r_2, \ldots r_j\}$ and let a subset of these codes be assigned to the codes for BRE $\{rBRE_1, rBRE_2, \ldots, rBRE_k\}$ and TATA $\{rTATA_1, rTATA_2, \ldots, rTATA_k\}$. There exists a condition of binding such that codes from BRE and TFIIA and TATA and TFIIA satisfy a condition at a binding threshold. There exists a set of joint binding probabilities for all of the required interactions. Using probability theory, we can express, for example, Equation (2) for the interaction of BRE, TFIIA and TATA with TFIIA.

$$J = P(\text{BRE} \cap \text{TFIIA}) \cap P(\text{TATA} \cap \text{TFIIA}) \tag{2}$$

Table 2 lists samples of the joint probabilities for protein–DNA binding in Figure 2.

Table 2. Sample of the required joint probabilities for binding of general transcription factor proteins to gene regulatory sequences.

Event
TFIIA∩BRE
TFIIA∩TATA
TFIIB∩BRE
TFIIB∩TATA
TFIIE∩INR
TFIIE∩MTE
TFIIF∩TATA
TFIIF∩INR
TFIID∩BRE
TFIID∩TATA
TFIID∩INR
TFIID∩MTE
TFIID∩DPE
TFIIH∩MTE
TFIIH∩DPE
TFIID∩TFIIA∩TATA
TFIIE∩TFIIF∩TFIIH

2.4. Coding for Control of Transcription Factor Binding

If we define the relationship between protein transcription factor and regulatory sequence in terms of jointly typical sets of the two sequences, then different levels of homology can be required in different authentication or confidentiality scenarios.

An example: Let Γ = {1, 2, 3, 4, 5}, a 5-tuple alphabet for gene regulatory sequences with type Γ_g consisting of equation set 2:

$$\begin{aligned} P_{g1} &= (2/10) = 0.2 \\ P_{g2} &= 0.4 \\ P_{g3} &= 0.1 \\ P_{g4} &= 0.1 \\ P_{g5} &= 0.2 \end{aligned} \tag{3}$$

The type class of Γ_g consists of all sequences within Γ with the same statistical distribution, as shown in equation set 3:

$$\begin{aligned} T(\Gamma_g) &= \{1122223455, 112225534, \ldots, 5543222211\} \\ |T(\Gamma_g)| &= \left(\tfrac{10!}{2!4!2!}\right) = 37,800 \end{aligned} \tag{4}$$

We can then define the code for a member of regulatory sequence BRE as g = 2,455,222,113 as a member of the type Γ which can contain all the codes for those regulatory sequences.

We can define metrics of sequences jointly typical to Γ such that a condition of binding occurs. Let Ψ = {0, 1, 2, 4, 5, 8, 9} 7-tuple alphabet of transcription factor codes for members of TFIIx (TFIIA, TFIIB, etc.). Let Ψ_{tf} consist of sets that conform to equation set 4:

$$\begin{gathered} P_{\Psi 0} = 1/10 \\ P_{\Psi 1} = 1/10 \\ P_{\Psi 2} = 2/10 \\ P_{\Psi 4} = 2/10 \\ P_{\Psi 5} = 1/10 \\ P_{\Psi 8} = 1/10 \\ P_{\Psi 9} = 2/10 \\ \left|T\left(\Psi_{tf}\right)\right| = \left(\frac{10!}{2!2!2!}\right) = 453,600 \end{gathered} \tag{5}$$

such a code as TFIID as *tf* = 5,089,292,414 fits the condition. We can define codes for different transcription factors of the family TFII. It is clear that we can define binding criteria as the mutual information between Ψ and Γ. Let *tf* and *g* have the following user-defined, pre-shared secret joint distribution as shown in Table 3.

Table 3. Joint distribution of gene regulatory sequences and transcription factor codes.

	tf	**0**	**1**	**2**	**4**	**5**	**8**	**9**
g		0.1	0.1	0.2	0.2	0.1	0.1	0.2
1	0.2	0.02	0.14	0.04	0	0	0	0
2	0.4	0	0	0.28	0.08	0.04	0	0
3	0.1	0	0	0.07	0.02	0.01	0	0
4	0.1	0	0	0	0.08	0.01	0.01	0
5	0.2	0	0	0	0	0.14	0.02	0.04

Define a new type, Ω, such that it conforms to the joint distribution of Γ and Ψ as shown in Table 4. Using the examples of *BRE* as g = 2,455,222,113 and TFIIA as *tf* = 5,089,292,414 and the output is a codeword complying with the statistical distribution shown in Table 4. The new codeword set is a set of prefix-free codes complying with the joint probability distribution in Table 3.

Code words of Type S are formed by combinations of the integers conforming to the joint probabilities. If the requirement for prefix-free codes is relaxed or the Table 4 coding methodology was susceptible to a frequency analysis of the codewords then tuples of type S in Ω in Table 4, could be replaced, for example with fractional parts of irrational numbers as shown in type T in Table 4. The selection rule would be a pre-shared secret between sender and receiver. The values could be taken from the fractional parts of hyperbolic sine and hyperbolic cosine functions. Code words of Type T are formed by combinations of the individual codewords in Ω.

Any process in transcription, translation including post-transcriptional modifications and posttranslational modifications can be coded using the techniques shown herein. Transcriptional regulation can be coded such that a threshold for binding interactions can be set by using the joint probabilities of binding. Figure 3 displays the expression of sufficient binding between a regulatory sequence and a protein where Figure 4 displays the expression of insufficient binding.

Table 4. Codewords for the joint distribution of gene regulatory sequences and transcription factors. Tuples of type S are generated to provide prefix-free codes using the combinations of types generated. Tuples of type T are converted to whole and fractional parts of the sinh and cosh functions applied to the type S codes.

Ω: i = {1, 2, 3, 4, 5} | | j = {0, 1, 2, 4, 5, 8, 9}	**Pij**	**Tuples of s in Type Ω**	**Tuples of T in Type Ω Using Sinh and Cosh Functions**
1, 0	0.02	3, 3	1001
1, 1	0.14	7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7	54,831,612,327,324
1, 2	0.04	5, 5, 5, 5	74,203
2, 2	0.28	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2	3,626,860,407,847,212,927,945,509,482
2, 4	0.08	1, 1, 1, 1, 1, 1, 1	15,430,806
2, 5	0.04	4, 4, 4, 4	27,308
3, 2	0.07	9, 9, 9, 9, 9, 9, 9	4,051,542
3, 4	0.02	89, 89	2244
3, 5	0.01	64	311
4, 4	0.08	81, 81, 81, 81, 81, 81, 81, 81	75,304,865
4, 5	0.01	826	372
4, 8	0.01	837	112
5, 5	0.14	62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62	78,334,370,730,000
5, 8	0.02	69, 69	4626
5, 9	0.04	68, 68, 68, 68	17,021

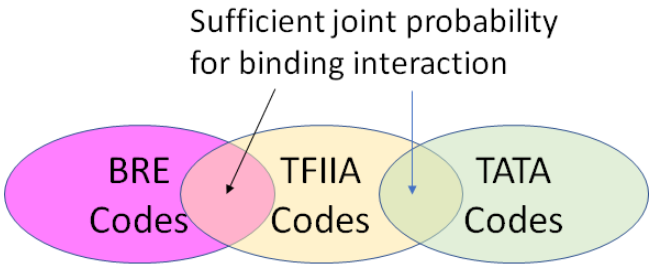

Figure 3. Binding of BRE to TFIIA and TATA to TFIIA. The network diagram in Figure 2 shows the associations between the B-recognition element (BRE) regulatory sequence and the TFIIA general transcription factor protein and the TATA regulatory sequence. Therefore there are non-zero codes for the joint probability of binding events.

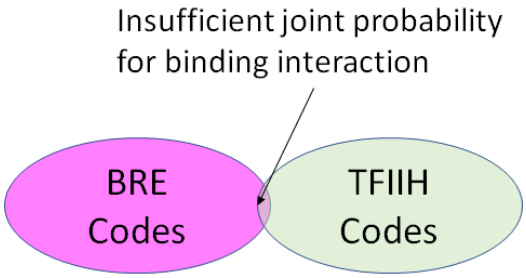

Figure 4. Non-binding of BRE to TFIIH. The network diagram in Figure 2 shows no association between the BRE regulatory sequence and the TFIIH general transcription factor. There is no binding between the two elements. It is possible to have no code assigned or a code representing a null association.

The same methodology used in coding the transcription process applies for coding the translation process. The user determines the level of fidelity from the actual biological processes to impose in the actual coding.

2.5. Features of the Genomic and Proteomic Security Protocol

- Every gene sequence used in the protocol is called a ciphergene, resides in a system called the ciphercolony, and is indexed by a ciphergene ID. The unauthorized disclosure of the ciphergene ID is a major vulnerability that must be prevented.
- The ciphergene ID points to all of the features unique to the expression of the gene. It is the single link to all of the information necessary to process and regulate transcription and translation for a given gene and message.
- Each output level of the protocol carries all the levels beneath it in its payload.
- Every gene sequence possesses the following attributes:
    - Matrix *F*, which contains the starting location of each Type in the gene along the diagonal.
    - Matrix, *G*, which contains a probability of expression for the gene in a given state. The number states are given by the number of diagonal entries in *G*. *F* and *G* are square and the same size.
    - A matrix *C*, which is the product of *F* and *G*.
    - Encryption matrices E_1, E_2, ... , E_n, that operate on *C*. Inverse decryption matrices E_1^{-1}, E_2^{-1},... E_n^{-1} that return *C*. In their simplest form, they could be rotations.
    - A series of regulatory networks that describes the interactions with proteins and other nucleic acids necessary for all the processes within this protocol.
    - K_{Tn} are binary sequences representing unique symmetric encryption keys. P_T is a binary sequence representing a message authentication code that is a pre-shared secret between transmitter and receiver. For this application, it could be any user specified binary sequence satisfying the requirements of a keyed message authentication code.
- One or more Types with each Type possess the following attributes:
    - A probability mass function to derive a code to represent each Type as utilized by the ciphergene.
    - A position in a regulatory network to describe its relationship to the other Types required for transcription or translation of the ciphergene. Each Type-to-Type relationship is a joint event.
    - A joint probability matrix with its mutual information to other Types required for transcription and translation using the joint event.
    - For every joint event, a code is derived from the joint probability matrix and the coding of the Types. This code is typically much longer than either of the codes for an individual Type in a joint event.
- For sequences that are converted from a DNA message to a DNA sequence or a DNA message to an mRNA sequence (and vice versa), there exists a coding process of ring subtraction over a subset of integers producing an addend and an inverse process of a ring addition over a subset of integers.
    - In a simple example, assume the plaintext in a message has been converted to a nucleotide sequence CCTACTAGT to be coded in a *β-globin* sequence ATGGTGCAT. Table 5 provides a simple example of ring addition process. A realistic application would use longer, and more complex substitution with multiple rounds.

Table 5. Example Coding and Decoding Message onto DNA.

	β-Globin	**A T G G T G C A T**
	Message	C C T A C T A G T
nt_j		1 4 2 2 4 2 3 1 4
M_j		3 3 4 1 3 4 1 2 4
$A_j = M_j - nt_j$	AddendCode	2 3 2 3 3 2 2 1 4
$M_j = A_j + nt_j$	Message	3 3 4 1 3 4 1 2 4

- For sequences that are converted from mRNA to protein (and vice versa) there exists a substitution process for selecting the amino acid code from a triplet of mRNA codes (codon) and a reverse substitution for recovering the codon from the amino acid code. The synonymous codons are coded uniquely.

Figure 5 summarizes the steps of the process. The initiating process can utilize the floating point encryption process developed by the author [27], however any process for converting plaintext to a DNA nucleotide string can be used. The overhead for the entire process ranges from approximately 40:1 to 1144:1 in terms of number of bits required to implement all levels of the protocols, although the actual overhead depends upon the user's choices of coding in the transcription and translation processes.

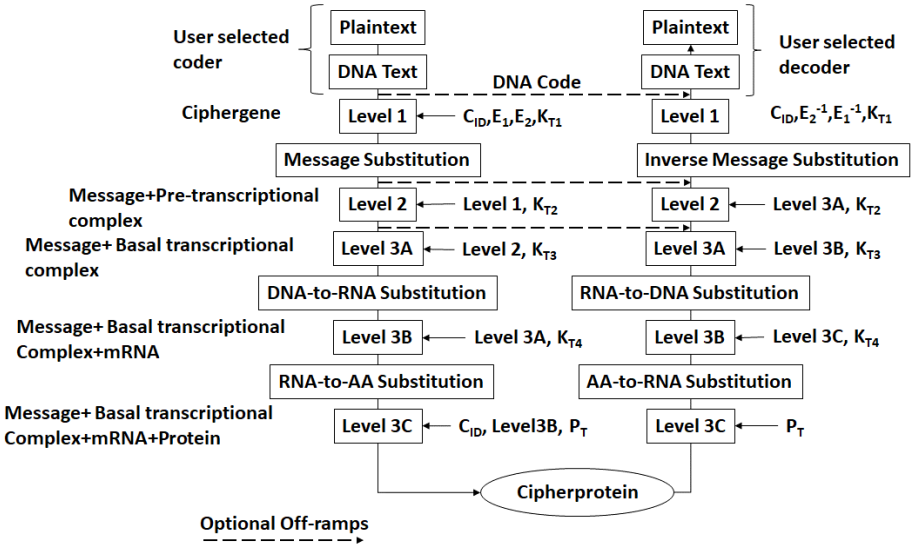

Figure 5. Genomic and Proteomic Flowchart for Encryption and Decryption through all levels of the protocol. Every message carries its entire transcriptional and translational basis.

3. Results

Applications of the Protocol That Fit within the Context of Existing Security Protocols

Assume that Alice and Bob have the necessary components of this system. One possible scenario for sending a secure message incorporating legacy protocols is shown in Figure 6.

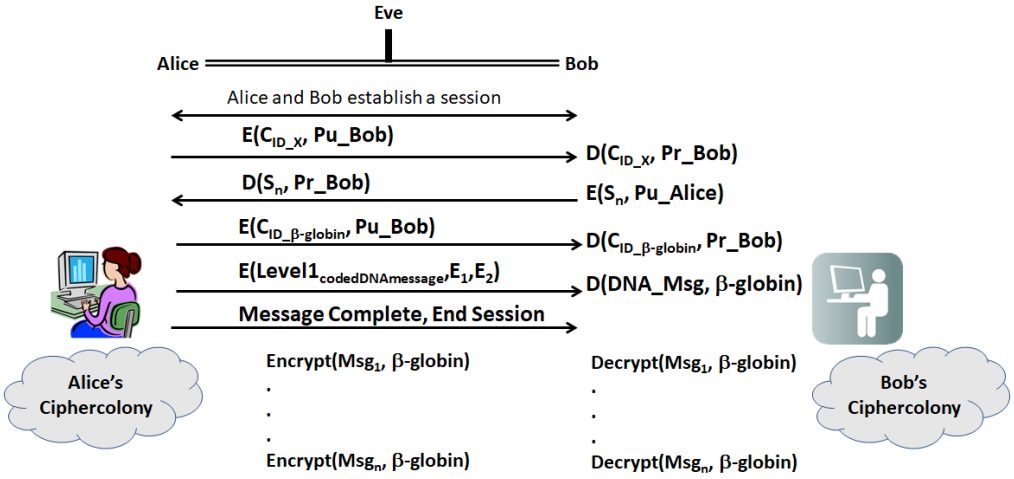

Figure 6. Alice and Bob communicating with established *β-globin* keys.

(a) First, Alice and Bob establish a secure session with their legacy protocols. Then, Alice sends Bob a ciphergene ID (CID), for a given gene, X, encrypted with Bob's public key

(b) Bob decrypts the CID with his private key and returns a sequence, S_n, which is a sequence of n bases from X. The location of the sequence is a pre-shared secret between Bob and Alice.

(c) Having established two forms of identity verification between Alice and Bob, Alice transmits the encrypted C_{ID} for *β-globin* with Bob's public key. Table 6 displays a set of Types that can be used in encrypting the message, which can be far more extensive that shown in the table. Implementers can construct the network of protein–protein and protein–nucleotide interactions from the literature on transcriptional regulation of *β-globin*. The other elements of the encryption and decryption at level 1 can be generated based upon Section 2.4. Alice transmits the Level 1 code derived from coding

(d) Bob decrypts the C_{ID} with his private key and uses C_{ID} to retrieve the *β-globin* sequence details and decryption keys, and then decrypts Level 1. Bob assembles the ciphergene and applies the addend code to retrieve the DNA text from the protein coding regions of the *β-globin* sequence.

(e) Bob can recover the plaintext using the source decoding process.

(f) Unless Eve can impersonate Bob or Alice in a man-in-the-middle attack, Eve must have access to keys $E_1, E_2, \ldots, E_n$ as well knowledge of the biogene regulatory structure to retrieve the plaintext or insert replacement ciphertext. Eve may be able to mount a mathematical attack on the keys, but knowledge of the regulatory structure of the message is required to completely retrieve the DNA text and knowledge of the pre-shared secret hash codes is required to retrieve the plain text from the DNA text.

Table 6. *β-globin* coding elements. *β-globin* gene sequence is taken from HBB-201 ENST00000335295.4 [30]. Regulatory regions were identified from [31], with influence from [32,33]. NC = Non-Coding, CP = Core Promoter, PC = Protein Coding, IN = Intron.

Coding Element	Sequence	Absolute Position	Type
Non Coding 1	GCAGGAGCCAGGGCTGGG	1–18	NC
TATA box	CATAAA	19–24	CP
Non Coding 2	AGTCAGGGCAGAGCCATCTATTGCTTA	25–51	NC
+20E to box	CAACTG	52–57	CP
Non Coding 3	CTTCTGACACAACTGT	58–73	NC
MARE box	GTTCACTAGCA	74–84	CP
Non Coding 4	ACCTCAAACAGACACC	85–100	NC
Protein Coding 1	ATGGTGCATCTGACTCCTGAGGAGAAGTCTGCCGTT ACTGCCCTGTGGGGCAAGGTGAACGTGGATGAAGT TGGTGGTGAGGCCCTGGGCAG	101–192	PC
Intron 1	GTTGGTATCAAGGTTACAAGACAGGTTTAAGGAGA CCAATAGAAACTGGGCATGTGGAGACAGAGAAGA CTCTTGGGTTTCTGATAGGCACTGACTCTCTCTGCCT ATTGGTCTATTTTCCCACCCTTAG	193–322	IN
Protein Coding 2	GCTGCTGGTGGTCTACCCTTGGACCCAGAGGTTCTT TGAGTCCTTTGGGGATCTGTCCACTCCTGATGCTGT TATGGGCAACCCTAAGGTGAAGGCTCATGGCAAGA AAGTGCTCGGTGCCTTTAGTGATGGCCTGGCTCACC TGGACAACCTCAAGGGCACCTTTGCCACACTGAGT GAGCTGCACTGTGACAAGCTGCACGTGGATCCTGA GAACTTCAGG	323–545	PC
Intron 2	GTGAGTCTATGGGACGCTTGATGTTTTCTTTCCCCTT CTTTTCTATGGTTAAGTTCATGTCATAGGAAGGGGA TAAGTAACAGGGTACAGTTTAGAATGGGAAACAG ACGAATGATTGCATCAGTGTGGAAGTCTCAGGATC GTTTTAGTTTCTTTTATTTGCTGTTCATAACAATTGT TTTCTTTTGTTTAATTCTTGCTTTCTTTTTTTTTCTTC TCCGCAATTTTTACTATTATACTTAATGCCTTAACA TTGTGTATAACAAAAGGAAATATCTCTGAGATACA TTAAGTAACTTAAAAAAAAACTTTACACAGTCTGC CTAGTACATTACTATTTGGAATATATGTGTGCTTATT TGCATATTCATAATCTCCCTACTTTATTTTCTTTTATTT TTAATTGATACATAATCATTATACATATTTATGGGTTA AAGTGTAATGTTTTAATATGTGTACACATATTGACCA AATCAGGGTAATTTTGCATTTGTAATTTTAAAAAAT GCTTTCTTCTTTTAATATACTTTTTTGTTTATCTTATTT CTAATACTTTCCCTAATCTCTTTCTTTCAGGGCAATA ATGATACAATGTATCATGCCTCTTTGCACCATTCTAA AGAATAACAGTGATAATTTCTGGGTTAAGGCAATAG CAATATCTCTGCATATAAATATTTCTGCATATAAATTG TAACTGATGTAAGAGGTTTCATATTGCTAATAGCAG CTACAATCCAGCTACCATTCTGCTTTTATTTTATGGT TGGGATAAGGCTGGATTATTCTGAGTCCAAGCTAGG CCCTTTTGCTAATCATGTTCATACCTCTTATCTTCCTC CCACAG	546–1395	IN
Protein Coding 3	CTCCTGGGCAACGTGCTGGTCTGTGTGCTGGCCCAT CACTTTGGCAAAGAATTCACCCCACCAGTGCAGGC TGCCTATCAGAAAGTGGTGGCTGGTGTGGCTAATGC CCTGGCCCACAAGTATCACTAA	1396–1524	PC
Non Coding 7	GCTCGCTTTCTTGCTGTCCAATTTCTATTAAAGGTTC CTTTGTTCCCTAAGTCCAACTACTAAACTGGGGGAT ATTATGAAGGGCCTTGAGCATCTGGATTCTGCCTAA TAAAAAACATTTATTTTCATTGCAA	1525–1628	NC

Another authentication scheme is shown in Figure 7. The IT security official receives a remote request for access to network assets from a remote user. The security official sends the user a message coded as a protein sequence, by a regulatory network using a message-specific set of protein–DNA joint distribution codes and a source coding scheme based upon a keyed hash function tied to a specific genome. The user successfully decrypts the message and returns the plaintext (which could be encrypted if desired) to the IT security official. The IT security official then sends a set of access credentials encrypted with a different protein and a different genome for the keyed hash code. The user

successfully decrypts the message to gain access to the network. In this scheme, an attacker needs multiple levels of information at the genomic and proteomic levels to be able to decode the message by cryptanalysis means alone.

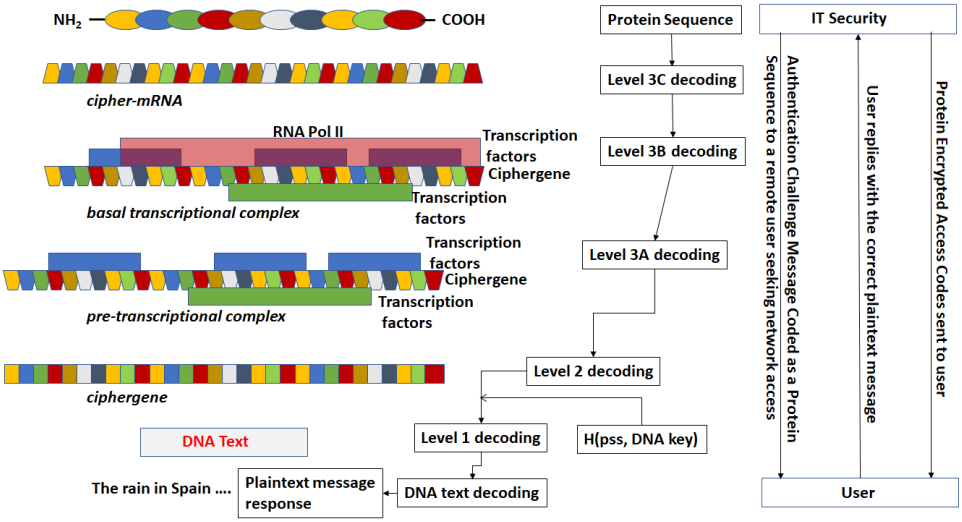

Figure 7. An authentication challenge using protein codes.

4. Discussion

The protocols have a wide range of implementation possibilities. The use of the Method of Types combined with the use of joint probabilities between binding elements permits error-tolerant authentication. This may be useful over free-space communication links in cases where the codes are very long and low E_b/N_0 leads to errors preventing authentication of acceptable users. This in turn could lead to a Quality of Protection (QoP) metric that includes a threshold ciphercode error rate.

The complexity of protocol can be shifted between the molecular biological domain and the information theory domain, i.e., more complex coding of protein–protein and protein–nucleic acid interactions can be traded against simpler implementations of the encryption scheme, and vice versa.

At a systems level, the protocols can be integrated into legacy public key infrastructure systems. Figure 8 shows the implementation the first layer of a three-layer hierarchal security framework using a Certificate Authority named the Bio-CA. It uses a traditional public key infrastructure approach to maintain compatibility with legacy security protocols.

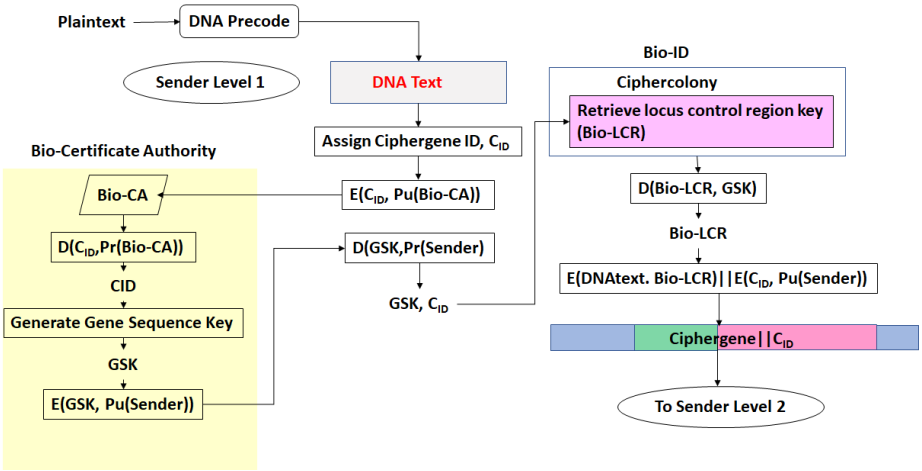

Figure 8. Level 1 Sender Encryption.

In Figure 8, the concept of a Network BioID is introduced. The Network BioID interfaces to a computer network and performs the full suite of authentication and confidentiality functions required by the protocol. It exchanges data with other Network BioIDs. It is a genomic and proteomic firewall. The heart of the Network BioID is the ciphercolony. The systems concept of a ciphercolony is now expanded to contain a combination of live and virtual inhabitants which maintain a collective pattern of gene expression.

The level 1 process is as follows: The Sender encrypts the ciphergene ID (CID) with a Bio-CA public key and transmits the encrypted CID to a remote Bio-CA. The Bio-CA decrypts the CID with its private key and retrieves a Gene Sequence Key Encryption Key (GSK) for the message associated with the CID. The Bio-CA encrypts the GSK with the Sender's public key and transmits the GSK to the Sender. The Sender decrypts the GSK with its private key and retrieves the locus control region key (Bio-LCR) from the BioID ciphercolony database. The Bio-LCR contains all of the data required for transcription, translation and all other required processes. The Bio-LCR is decrypted with the GSK. The DNA text is encrypted with the Bio-LCR, converting the DNA text to a ciphergene. The CID is encrypted with the public key of the sender and concatenated with the ciphergene for Level 2 encryption. This completes Level 1 encryption.

The process of decrypting ciphergene to DNA text is the reverse of the encryption process as shown in Figure 9. The CID is decrypted with the Receiver private key and encrypted with the Bio-CA public key and then sent to the remote Bio-CA, decrypted with the Bio-CA private key, and the GSK is retrieved. The GSK is encrypted with the Receiver public key and transmitted to the Receiver. The Receiver decrypts the GSK with its private key and retrieves the Bio-LCR from the BioID ciphercolony database. The Bio-LCR is decrypted with the GSK. The ciphergene is decrypted with the Bio-LCR and converted to DNA text for Level 1 decryption. This completes Level 1 decryption. The end result is the plaintext.

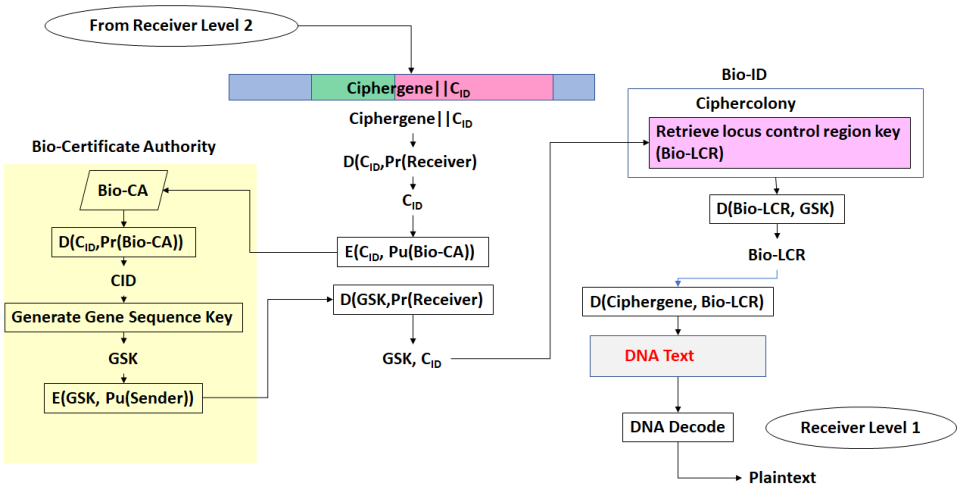

Figure 9. Receiver Level 1 Decryption.

Novel Features of the Protocol for Future Extension of the Capabilities

Genomics and proteomics involve modellable networks which can be converted into cryptographic codes at many levels. In this paper, nucleic acid–protein level (networks of nucleic acid–protein interactions and nucleic acid–nucleic acid interactions have been described. It can be expanded to include the following:

- Patterns of gene expression (networks of gene interactions)
- Intercellular systems (networks of cellular interactions, e.g., biofilms)

And so forth into higher complexity networks of complex eukaryotic and prokaryotic systems.

The protocol permits a future expansion into forms of network security in which colonies exchange patterns of gene expression and respond to those changes with alterations in their own patterns of gene expression via cellular signaling and responses to gene expression regulatory networks. It supposes that both live and virtual colonies contribute to a collective pattern of gene expression and that users can maintain colonies with a channel to exchange patterns of gene expression information for the purposes of authentication, confidentiality, data integrity, non-repudiation, and access control.

Extension of Firewall Capabilities

Attacks such as the Wannacry May 2017 attack [34] and the US Office of Personnel Management data breach from June 2014–2015 [35] demonstrate vulnerabilities in the conventional firewall. Vulnerabilities such as the "Living off the Land" [36] attack against the US Democratic National Committee demonstrate how simple spear-phishing attacks gained access to sensitive emails without the use of sophisticated malware. Use of a firewall with both biological recognition and conventional firewall capabilities may help to thwart some of these attack vectors. This type of firewall becomes more practical with advances in lab-on-chip capabilities for real-time assessment of patterns of gene expression. Figure 10 provides a high-level view of how such as firewalls could operate using the Network Bio-ID concept.

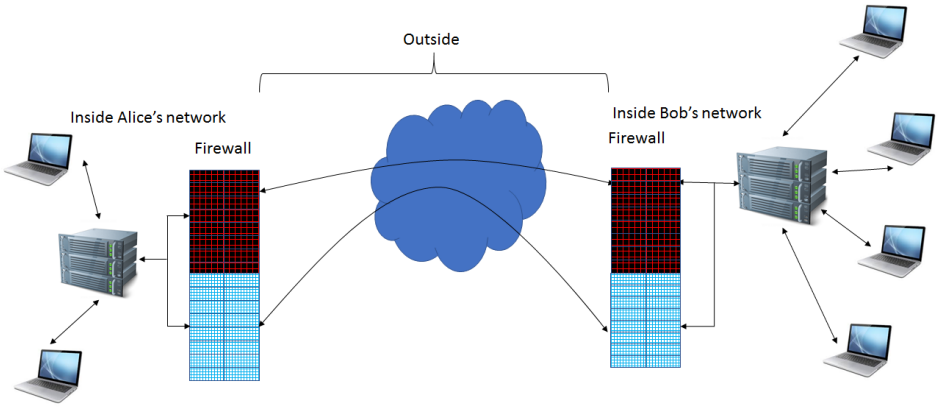

Figure 10. A depiction of cooperative conventional and bio-firewalls. Intervening routers, switches and network hardware are not shown. The two bio-firewalls consist of their respective Network Bio-ID and ciphercolonies. The Network Bio-ID will contain lab-on-a-chip capabilities as well as the entire database of information required to create, regulate and maintain algorithmic and live patterns of gene expression. This will permit the bio-firewalls to recognize and modulate patterns of gene expression with similarly equipped bio-firewalls.

Firewalls apply sets of rules to incoming and outgoing traffic. Conventional firewalls provide packet filtering. In the case of stateful firewalls, they determine the connection state of packets, and there can be application firewalls as well. The firewalls can accept, reject or drop incoming and outgoing connections. This would be augmented by a Bio-firewall which will evaluate recognition of other bio-firewalls, using a different form of rule sets. This recognition would be based upon mutual recognition of patterns of gene expression, recognition of genotypes, and frequency of contact. By exchanging information on patterns of gene expression, the gene expression patterns can be adapted as if they were in physical contact, thus providing new modes of security.

These firewalls could also be interfaced to other security applications in platforms such as mobile phones. It could be integrated with voice recognition systems [37], facial recognition [38], and fingerprint recognition [39].

5. Conclusions

A set of concepts for integrating the power of regulation of gene expression into network security has been presented. The ability to integrate regulation of gene expression into security comes with a high overhead but opens possibilities beyond the set of current legacy security solutions for information security and network security. It is also compatible with future biological instantiations of information and network security.

6. Patents

Shaw, H. C., U.S. Patent 8,898,479, 2014.

Acknowledgments: Thanks to the NASA Goddard Space Flight Center for support of this work.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Shaw, H.C. *Genomics and Proteomics Based Security Protocols for Secure Network Architectures*; The George Washington University: Washington, DC, USA, 2013.
2. Novak, R.; Naccache, D.; Paillier, P. SPA-Based Adaptive Chosen-Ciphertext Attack on RSA Implementation. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2002.
3. Schindler, W.A. Timing Attack against RSA with the Chinese Remainder Theorem. In *Cryptographic Hardware and Embedded Systems*; Springer: Berlin, Germany, 1965; pp. 109–124.
4. Wiener, M.J. Cryptanalysis of short RSA secret exponents. *IEEE Trans. Inf. Theory* **1990**, *36*, 553–558. [CrossRef]
5. Misarsky, J.F. A multiplicative attack using LLL algorithm on RSA signatures with redundancy. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 1997; pp. 221–234.
6. Bindel, N.; Buchmann, J.; Krämer, J. Lattice-Based Signature Schemes and their Sensitivity to Fault Attacks. In Proceedings of the 13th Workshop on Fault Diagnosis and Tolerance in Cryptography, Santa Barbara, CA, USA, 16 August 2016.
7. Kocher, P.C. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 1996; pp. 104–113.
8. Yarom, Y.; Genkin, D.; Heninger, N.J. CacheBleed: A timing attack on OpenSSL constant-time RSA. *J. Cryptogr. Eng.* **2017**, *2*, 99–112. [CrossRef]
9. Adham, M.; Azodi, A.; Desmedt, Y.; Karaolis, I. How to Attack Two-Factor Authentication Internet Banking. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2013; pp. 322–328.
10. Min-Seok, P.; Hüseyin, T. Security Breaches in the U.S. Federal Government. Available online: https://ssrn.com/abstract=2933577 (accessed on 20 November 2017).
11. Singh, H.; Chugh, K.; Dhaka, H.; Verma, A.K. DNA based Cryptography: An Approach to Secure Mobile Networks. *Int. J. Comput. Appl.* **2010**, *1*, 77–80. [CrossRef]
12. Karimi, M.; Haider, W. Cryptography using DNA Nucleotides. *Int. J. Comput. Appl.* **2017**, *168*, 16–18. [CrossRef]
13. Bevi, A.R.; Malarvizhi, S.; Patel, K. Information Coding and its Retrieval using DNA Cryptography. *J. Eng. Sci. Technol. Rev.* **2016**, *9*, 86–92.
14. Vijayakumar, P.; Vijayalakshmi, V.; Zayaraz, G. DNA Computing based Elliptic Curve Cryptography. *Int. J. Comput. Appl.* **2011**, *36*, 18–21.
15. Gehani, A.; LaBean, T.; Reif, J. DNA-Based Cryptography, Aspects of Molecular Computing. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2004; Volume 2950, pp. 167–188.
16. Sadeg, S.; Gougache, M.; Mansouri, N.; Drias, H. An encryption algorithm inspired from DNA. In Proceedings of the International Conference on Machine and Web Intelligence (ICMWI), Algiers, Algeria, 3–5 October 2010; pp. 344–349.
17. Raj, B.B.; Vijay, J.F.; Mahalakshmi, T. Secure Data Transfer through DNA Cryptography using Symmetric Algorithm. *Int. J. Comput. Appl.* **2016**, *133*, 19–23.

18. Bourbakis, N.G. Image Data Compression-Encryption Using G-Scan Patterns. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics and simulation, Orlando, FL, USA, 12–15 October 1997; pp. 1117–1120.
19. Leier, A.; Richter, C.; Banzhaf, W.; Rauhe, H. Cryptography with DNA binary strands. *Biosystems* **2000**, *57*, 13–22. [CrossRef]
20. Heider, D.; Barnekow, A. DNA-based watermarks using the DNA-Crypt algorithm. *BMC Bioinform.* **2007**, *8*, 176. [CrossRef]
21. Heider, D.; Barnekow, A. DNA watermarks: A proof of concept. *BMC Mol. Biol.* **2008**, *9*, 40. [CrossRef] [PubMed]
22. Hamad, S.; Elhadad, A.; Khalifa, A. DNA watermarking using Codon Postfix technique. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2017**. [CrossRef] [PubMed]
23. Shaw, H.; Hussein, S. A DNA-Inspired Encryption Methodology for Secure, Mobile Ad-Hoc Networks (MANET). In Proceedings of the First International Conference on Biomedical Electronics and Devices, BIOSIGNALS, Funchal, Portugal, 28–31 January 2008; pp. 472–477.
24. Shaw, H.; Hussein, S.; Helgert, H. Prototype Genomics-Based Keyed-Hash Message Authentication Code Protocol. In Proceedings of the Second International Conference on Evolving Internet, Valcencia, Spain, 20–25 September 2010; pp. 131–136.
25. Shaw, H.; Hussein, S.; Helgert, H. Genomics-Based Security Protocols: From Plaintext to Cipherprotein. *Int. J. Adv. Secur.* **2011**, *4*, 106–117.
26. Shaw, H.; Hussein, S.; Helgert, H. Adaptive Self-Correcting Floating Point Source Coding Methodology for a Genomic Encryption Protocol. *Int. J. Comput. Appl.* **2012**, *56*, 1–5.
27. Basehoar, A.D.; Zanton, S.J.; Pugh, B.F. Identification and Distinct Regulation of Yeast TATA Box-Containing Genes. *Cell* **2004**, *116*, 699–709. [CrossRef]
28. Meister, G. *RNA Biology: An Introduction*; WileyVCH: Weinheim, Germany, 2011; pp. 20–70.
29. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; Wiley: New York, NY, USA, 1991; pp. 347–665.
30. Ensembl Genome Browser. Available online: https://www.ensembl.org/Homo_sapiens/Transcript/Exons?db=core;g=ENSG00000244734;r=11:5225464-5227071;t=ENST00000335295 (accessed on 16 October 2017).
31. Leach, K.M.; Vieira, K.F.; Kang, S.H.L.; Aslanian, A.; Teichmann, M.; Roeder, R.G.; Bungert, J. Characterization of the human *β-globin* downstream promoter region. *Nucleic Acids Res.* **2003**, *31*, 1292–1301. [CrossRef] [PubMed]
32. Juven-Gershon, T.; Hsu, J.Y.; Theisen, J.W.; Kadonaga, J.T. The RNA Polymerase II Core Promoter—The Gateway to Transcription. *Curr. Opin. Cell Biol.* **2008**, *20*, 253–259. [CrossRef] [PubMed]
33. Sengupta, T.; Cohet, N.; Morlé, F.; Biekera, J.J. Distinct modes of gene regulation by a cell-specific transcriptional activator. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 4213–4218. [CrossRef] [PubMed]
34. Hoeksma, J. NHS cyberattack may prove to be a valuable wake up call. *Br. Med. J.* **2017**. [CrossRef] [PubMed]
35. Finklea, K.; Christensen, M.D.; Fischer, E.A.; Lawrence, S.V.; Theohary, C.A. *Cyber Intrusion into US Office of Personnel Management: In Brief*; Library of Congress: Washington, DC, USA, 2015.
36. Symantec Security Center Threat Report. Available online: https://www.symantec.com/security-center/threat-report (accessed on 3 November 2017).
37. Mittal, P.; Singh, N. Speech Based Command and Control System for Mobile Phones: Issues and Challenges. In Proceedings of the Second International Conference on Computational Intelligence & Communication Technology (CICT), Changsha, China, 13–15 October 2017; pp. 729–732.
38. Alshamsi, H.; Meng, H.; Li, M. Real time facial expression recognition app development on mobile phones. In Proceedings of the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 13–15 October 2017; pp. 1750–1755.
39. Prasad, M.V.; Anugu, J.R.; Rao, C.R. Fingerprint template protection using multiple spiral curves. In *Smart Innovation, Systems and Technologies*; Springer: New Delhi, India, 2015; pp. 593–601.